



**REPUBLIC OF TÜRKİYE
ADANA ALPARSLAN TÜRKES SCIENCE AND TECHNOLOGY
UNIVERSITY**

**INSTITUTE OF GRADUATE SCHOOL
COMPUTER ENGINEERING DEPARTMENT**

**OPTIMIZATION ALGORITHMS INSPIRED BY SEA CREATURES IN
FEATURE SELECTION APPLICATIONS**

DENİZ FURKAN KANBAK

MASTER OF SCIENCE



**REPUBLIC OF TÜRKİYE
ADANA ALPARSLAN TÜRKES SCIENCE AND TECHNOLOGY
UNIVERSITY**

**INSTITUTE OF GRADUATE SCHOOL
COMPUTER ENGINEERING DEPARTMENT**

**OPTIMIZATION ALGORITHMS INSPIRED BY SEA CREATURES IN
FEATURE SELECTION APPLICATIONS**

DENİZ FURKAN KANBAK

MASTER OF SCIENCE

**THESIS ADVISOR
Assoc. Prof. Dr. Mümine KAYA KELEŞ**

ADANA, 2023

DECLARATION OF CONFORMITY

In this thesis study, which was prepared following the thesis writing rules of Adana Alparslan Türkeş Science and Technology University Institute of Graduate School, I declare that I provide all the information, documents, evaluations and results in accordance with scientific ethics and moral codes without resorting to any means or assistance that would be contrary to scientific ethics and traditions. I also declare that I refer to all of the articles I used in this study with appropriate references and accept all moral and legal consequences if a situation is found contrary to my statement regarding my work.

[Signature]

Deniz Furkan KANBAK

ÖZET

DENİZ CANLILARINDAN İLHAM ALINARAK OLUŞTURULAN OPTİMİZASYON ALGORİTMALARININ ÖZNİTELİK SEÇİMİ UYGULAMALARINDA KULLANIMI

Deniz Furkan KANBAK

Yüksek Lisans, Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Mümine KAYA KELEŞ

Temmuz 2023, 115 sayfa

Veri madenciliği, makine öğrenimi ve örüntü tanıma gibi uygulamalarda, öznitelik seçimi önemli bir ön işleme adımıdır. Gereksiz ve alakasız özellikleri kaldırarak, öğrenme modelinin hesaplama maliyetini azaltırken sınıflandırma doğruluğunu artırır. Bu tezde, doğadan ilham alınarak geliştirilen dokuz metasezgisel algoritma ve bir geleneksel yöntem, UCI veritabanından alınan yirmi iki veri seti üzerinde test edildi. Test sonuçları; doğruluk, kesinlik, duyarlılık, F-Score ve seçim boyutu dahil olmak üzere beş farklı değerlendirme metriği kullanılarak değerlendirildi. Algoritmalar ve sınıflandırıcılar arasındaki ilişkiyi daha iyi anlamak için dört farklı sınıflandırıcı kullanıldı. Kullanılan algoritmalar üç kategoriye ayrılır: Emperor Penguin Colony (EPO), Marine Predators Algorithm (MPA), Salp Swarm Algorithm (SSA), Jellyfish Search Optimizer (JS), Sailfish Optimizer (SFO) ve Whale Optimizer (WOA) gibi deniz canlılarından esinlenen algoritmalar; Ant Colony Optimizer (ACO), Genetic Algorithm (GA) ve Particle Swarm Algorithm (PSA) gibi literatürde iyi bilinen algoritmalar; Ki-Kare (CHI2) olarak bilinen geleneksel yöntem. Bu tezde, JS algoritmasını literatürde ilk kez öznitelik seçim sürecinde kullanılmak üzere ikili JS algoritması olarak öneriyoruz. Test sonuçları; EPO'nun naïve bayes sınıflandırıcısını kullanarak en iyi algoritma olduğunu, MPA'nın destek vektör makinesi (SVM) sınıflandırıcısını kullanarak en iyi algoritma olduğunu, SFO'nun K-En Yakın Komşu (KNN) sınıflandırıcısını kullanarak en iyi algoritma olduğunu ve JS'nin özellik seçme işlemi için rassal orman (RF) sınıflandırıcısını kullanarak en iyi algoritma olduğunu gösterdi.

Anahtar Kelimeler: Denizanası Optimizasyonu, Öznitelik Seçimi, Doğadan Esinlenen Metasezgisel Algoritmalar, Deniz Canlılarından Esinlenen Algoritmalar, Makine Öğrenmesi

ABSTRACT

OPTIMIZATION ALGORITHMS INSPIRED BY SEA CREATURES IN FEATURE SELECTION APPLICATIONS

Deniz Furkan KANBAK

M.Sc., Department of Computer Engineering

Supervisor: Assoc. Prof. Dr. Mümine KAYA KELEŞ

July 2023, 115 pages

In applications such as data mining, machine learning, and pattern recognition, feature selection is an essential preprocessing step. By removing redundant and irrelevant features, it increases its classification accuracy while reducing the computational expense of the learning model. In this thesis, nine nature inspired metaheuristic algorithms and one traditional method are tested on twenty-two benchmark datasets taken from the UCI database. Test results are evaluated using five different evaluation metrics including accuracy, precision, recall, F-Score and selection size. Four different classifiers are used to better understand the correlation between the algorithms and the classifiers. Algorithms are divided into three categories: algorithms inspired from sea creatures including Emperor Penguin Colony (EPO), Marine Predators Algorithm (MPA), Salp Swarm Algorithm (SSA), Jellyfish Search Optimizer (JS), Sailfish Optimizer (SFO) and Whale Optimizer (WOA); well-known algorithms in the literature including Ant Colony Optimizer (ACO), Genetic Algorithm (GA) and Particle Swarm Algorithm (PSA); a traditional method known as Chi-Square (CHI2). In this thesis, we propose JS algorithm as binary JS algorithm to be used for the feature selection process for the first time in the literature. Test results showed that EPO is the best algorithm using naïve bayes classifier, MPA is the best algorithm using support vector machine (SVM) classifier, SFO is the best algorithm using K-Nearest Neighbor (KNN) classifier and JS is the best algorithm using random forest (RF) classifier for the feature selection process.

Keywords: Jellyfish Search Optimizer, Feature Selection, Nature Inspired Metaheuristic Algorithms, Algorithms Inspired by Sea Creatures, Machine Learning

TABLE OF CONTENTS

DECLARATION OF CONFORMITY	i
ÖZET	ii
ABSTRACT	iii
TABLE OF CONTENTS	iv
LIST OF FIGURES	vi
LIST OF TABLES	vii
NOMENCLATURE	ix
1. INTRODUCTION	0
2. LITERATURE REVIEW	5
2.1. Nature Inspired Algorithms	5
2.1.1. Whale Optimization Algorithm (WOA)	5
2.1.2. Emperor Penguin Optimization (EPO)	7
2.1.3. Marine Predators Algorithm (MPA)	9
2.1.4. Salp Swarm Algorithm (SSA)	10
2.1.5. Ant Colony Optimization (ACO)	12
2.1.6. Sailfish Optimizer (SFO)	13
2.1.7. Genetic Algorithm (GA)	14
2.1.8. Partical Swarm Optimization (PSO)	15
2.1.9. Jellyfish Search Optimization (JS)	16
3. MATERIAL AND METHODS	17
3.1. Dataset	17
3.2. Classifiers	18
3.2.1. Naïve Bayes	18
3.2.2. Support Vector Machine (SVM)	19
3.2.3. Random Forest (RF)	19
3.2.4. K-Nearest Neighbors (KNN)	20
3.3. Nature Inspired Algorithms	20
3.3.1. Emperor Penguin Colony Optimizer	20
3.3.2. Marine Predators Algorithm	23
3.3.2.1. Levy Flight	24
3.3.2.2. Brownian Motion	24

3.3.2.3. FAD's Effect	27
3.3.2.4 Marine Memory	27
3.3.3. Salp Swarm Algorithm	28
3.3.4. Jellyfish Optimizer	30
3.3.4.1. Ocean Current	31
3.3.4.2. Jellyfish Swarm	32
3.3.4.3. Time Control Mechanism	33
3.3.4.4. Binary Jellyfish Optimizer	33
3.3.5. Sailfish Optimizer	34
3.3.6. Whale Optimizer	36
3.3.6.1. Encircling the prey	37
3.3.6.2. Bubble-net attacking method	38
3.3.6.3. Search for prey	39
3.3.7. Ant Colony Optimization	40
3.3.8. Genetic Algorithm	41
3.3.9. Partical Swarm Optimization	42
3.4. Evaluation Metrics	43
4. RESULTS AND DISCUSSION	43
4.1. Parameter Setting	43
4.2. Experiment Results	45
4.2.1. Based on Accuracy	45
4.2.2. Based on Precision	49
4.2.3. Based on Recall	53
4.2.4. Based on F-Score	57
4.2.5. Based on Selection Size	61
4.2.6. Overall Comparison	73
5. CONCLUSION	78
REFERENCES	81

LIST OF FIGURES

Figure 1.1. Subcategories of metaheuristic algorithms.....	
Figure 3.1. Spirial-like movement.....	
Figure 3.2. Example of salps chain.....	
Figure 3.3. Sample of solution in binary seach space.....	
Figure 3.4. Encircling motion of sailfish.....	
Figure 3.5. Bubble-net mechanism.....	



LIST OF TABLES

Table 1.1. Optimization Algorithms Inspired by Sea Creatures.....	5
Table 3.1 Details of the datasets.....	19
Table 3.2 The evaluation criteria's formulas.....	45
Table 4.1. Accuracy results using naïve bayes.....	47
Table 4.2. Accuracy results using support vector machine.....	48
Table 4.3. Accuracy results using k-nearest neighbor.....	49
Table 4.4. Accuracy results using random forest.....	50
Table 4.5. Precision results using naïve bayes.....	51
Table 4.6. Precision results using support vector machine.....	52
Table 4.7. Precision results using k-nearest neighbor.....	53
Table 4.8. Precision results using random forest.....	54
Table 4.9. Recall results using naïve bayes.....	55
Table 4.10. Recall results using support vector machine.....	56
Table 4.11. Recall results using k-nearest neighbor.....	57
Table 4.12. Recall results using random forest.....	58
Table 4.13. F-score results using naïve bayes.....	59
Table 4.14. F-score results using support vector machine.....	60
Table 4.15. F-score results using k-nearest neighbor.....	61
Table 4.16. F-score results using random forest.....	62
Table 4.17. Comparison of the algorithms inspired by sea creatures based on selection size using KNN.....	63
Table 4.18. Comparison of the popular optimizers and CHI2 based on selection size using..	64
Table 4.19. Comparison of the best algorithms based on selection size using KNN.....	65
Table 4.20. Comparison of the algorithms inspired by sea creatures based on selection size using naïve bayes.....	66
Table 4.21. Comparison of the popular optimizers and CHI2 based on selection size using naïve bayes.....	67
Table 4.22. Comparison of the best algorithms based on selection size using naïve bayes....	68
Table 4.23. Comparison of the algorithms inspired by sea creatures based on selection size using support vector machine.....	69

Table 4.24. Comparison of the popular optimizers and CHI2 based on selection size using support vector machine.....	70
Table 4.25. Comparison of the best algorithms based on selection size using support vector machine.....	71
Table 4.26. Comparison of the algorithms inspired by sea creatures based on selection size using random forest.....	72
Table 4.27. Comparison of the popular optimizers and CHI2 based on selection size using random forest.....	73
Table 4.28. Comparison of the best algorithms based on selection size using random forest.....	74
Table 4.29. Comparison of the best algorithms based on classification accuracy.....	75
Table 4.30. Comparison of the best algorithms based on F-Score.....	76
Table 4.31. Comparison of the best algorithms based on recall.....	77
Table 4.32. Comparison of the best algorithms based on precision.....	78

NOMENCLATURE

GA	: Genetic Algorithm
BBO	: Biogeography-Based Optimizer
GP	: Genetic Programming
PBIL	: Probability-Based Incremental Learning
AAA	: Artificial Algae Algorithm
CRO	: Coral Reef Optimization
DE	: Dolphin Ecolocation
EPC	: Emperor Penguins Colony
FSS	: Fish School Search
HA	: Hydrozon Algorithm
JS	: Jellyfish Search Optimizer
KH	: Krill Herd Optimization Algorithm
MPA	: Marine Predators Algorithm
MBF	: Mouth Brooding Fish Algorithm
PeSOA	: Penguins Search Optimization Algorithm
PMSO	: Pontogammarus Maeoticus Swarm Optimization
ROA	: Remora Optimization Algorithm
SFO	: Sailfish Optimizer
SSA	: Salp Swarm Algorithm
SLnO	: Sea Lion Optimization
SOA	: Seagull Optimization Algorithm
WOA	: Whale Optimization Algorithm
ANOVA	: Analysis of Variance
PSO	: Particle Swarm Optimization
ACO	: Ant Colony Optimization
GWO	: Grey Wolf Optimization
ALO	: Ant Lion Optimizer
SA	: Simulated Annealing
EPO	: Emperor Penguin Optimizer

BPSO	: Binary Particle Swarm Optimization
BBA	: Binary Bat Algorithm
BGWO	: Binary Grey Wolf Optimizer
BDA	: Binary Dragonfly Algorithm
BSHO	: Binary Spotted Hyena Optimizer
BGSA	: Binary Gravitational Search Algorithm
BWOA	: Binary Whale Optimization Algorithm
MFO	: MothFlame Optimization Algorithm
SCA	: Sine Cosine Algorithm
BFO	: Butterfly Optimization Algorithm
HHO	: Harris Hawks Optimization
HGSO	: Henry Gas Solubility Optimization
GOA	: Grass-hopper Optimization Algorithm
DT	: Decision Tree
AGAFL	: Adaptive Genetic Algorithm with Fuzzy Logic
SVM	: Support Vector Machine
ML	: Machine Learning
RF	: Random Forest

1. INTRODUCTION

The importance of data has increased in a rapidly changing world. In order to predict and shape the future, several analyses are carried out on past data. As soon as humans realized the importance of data, every possible piece of information from every device was collected and put in a process. It is difficult to determine which data are really important when analyzing these data in real-world applications. By removing irrelevant features from a dataset, it is possible to reduce the amount of time it takes to apply a mining model. Thus, it is difficult to identify an ideal collection of features and to eliminate those that are unnecessary. In applications such as data mining, machine learning, and pattern recognition, feature selection is an essential preprocessing step. It is essential to master the art of feature selection in order to be successful in the field of machine learning. Besides reducing the computational expense of the learning model, it also increases its classification accuracy by removing redundant and irrelevant columns. By selecting only a small subset of features from the vast range of initial features, feature selection seeks to improve classification efficiency.

Methods have been developed for evaluating candidate subsets and determining the performance of the selected features. The methods are divided into three categories, which include filter methods, wrapper methods, and embedded methods in the literature. The wrapper approach is used to evaluate the generated feature subsets using a classifier. The filter approach is the process of evaluating feature subsets based on their statistical or information content. The hybrid methods combine the advantages of filtering and wrapping approaches. Filter methods evaluate and rank the features based on specific criterias such as information gain, chi-square analysis, Pearson's correlation, linear discriminant analysis, Analysis of Variance (ANOVA), V-score, Fisher Score (Kalayci, 2019; Jović et al., 2015). As a result of their lower computational cost, filter methods are usually faster than wrapper methods. Nevertheless, wrapper methods generally perform better than filter methods because they select better features from the original set of features.

As a matter of theory, a feature selection method should search through a set of subsets of features and find the best one among the possible subsets of features according to a certain evaluation criteria that is set out. It is considered to be an NP-hard problem. When a dataset

contains n features, an ideal feature selection process will have to evaluate 2^n feature subsets to determine which is the best subset out of n features. This process might be looking applicable for small datasets but the computational and time costs are not negligible. The use of heuristics and metaheuristics methods has proven to be effective in generating well-quality solutions to NP-Hard problems in acceptable run times.

Nature inspired metaheuristic algorithms solve optimization problems imitating biological or physical development. These algorithms can be divided into 4 categories: Evolutionary Algorithms, Physics-based Algorithms, Swarm-based Algorithms and Human-based Algorithms. Natural laws of evolution play a crucial role in guiding the development of evolution-based methods. At the beginning of the search process, a random population is generated that is progressively evolved over the course of successive generations. In terms of the strength of these methods, they have the advantage of always combining the best individuals into the next generation of individuals to form the next generation of individuals. In this way, the population can be optimized over a period of time over the course of generations. Popular evolutionary algorithms are Genetic Algorithm (GA), Biogeography-Based Optimizer (Simon, 2008), Genetic Programming (Koza, 1998) and Probability-Based Incremental Learning (Dasgupta and Michalewicz, 2013). Physics-based methods are based on physical laws that govern the universe as a whole. The most popular algorithms are Curved Space Optimization (Moghaddam et al., 2012), Small-World Optimization Algorithm (Du et al., 2006), Black Hole (Hatamlou, 2016), Gravitational Search Algorithm (Rashedi et al., 2009) and Simulated Annealing (Kirkpatrick et al., 1983). The third group of the metaheuristic algorithms is swarm-based algorithms. In the third group of nature-inspired techniques, algorithms mimic the social behavior of groups of animals by mimicking the behavior of individual animals. The most popular algorithms are Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Cuckoo Search (Yang and Deb, 2009), Bat Algorithm (Mirjalili et al., 2014) and Grey Wolf Optimization (GWO) (Mirjalili et al., 2014). The last category of the metaheuristic algorithms are human-based algorithms. Some of the most popular algorithms are Harmony Search (Mahdavi et al., 2007), Tabu Search (Glover et al., 1993), and League Championship Algorithm (Kashan, 2009). Regardless of their nature, population-based meta-heuristic optimization

algorithms share one common characteristic. There are two phases to the search process: exploration and exploitation.

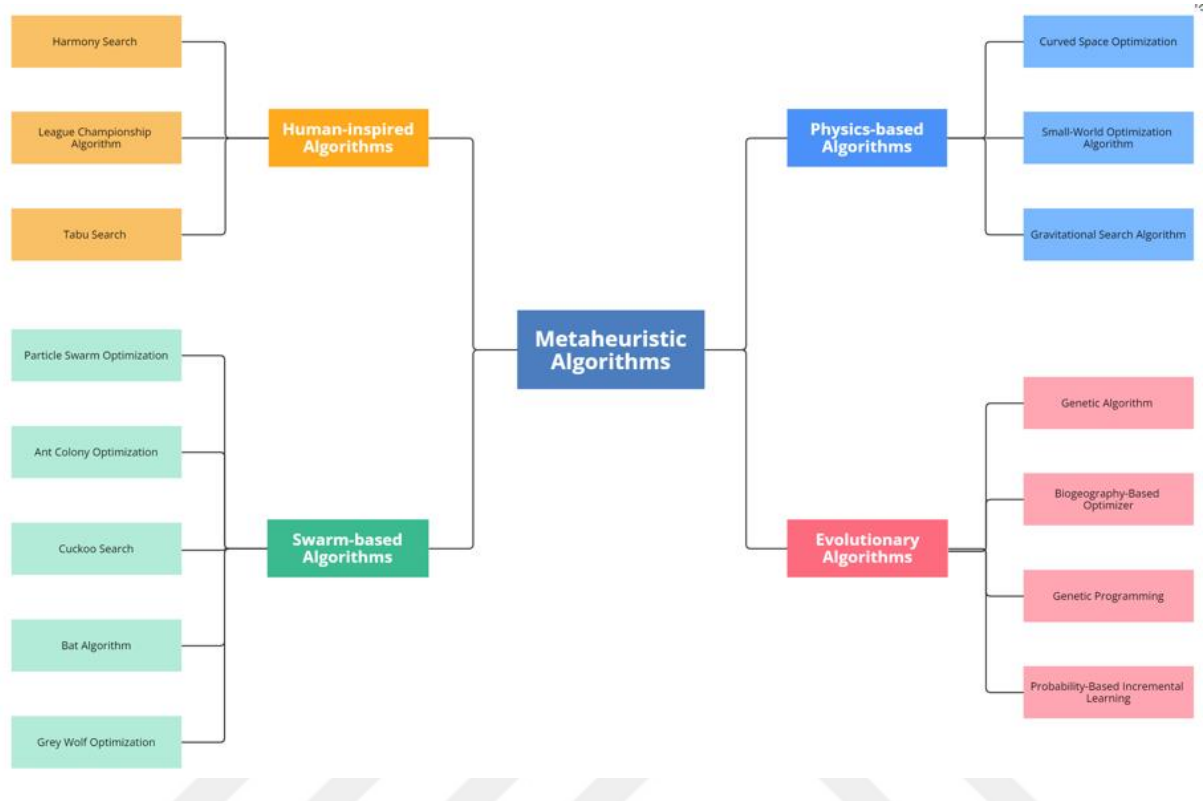


Figure 1.1. Subcategories of metaheuristic algorithms

A heuristic, metaheuristic or a random search method can be used to identify optimum subsets. In order to search the solution space, several metaheuristic search methods have been used in the literature such as GA, tabu search, PSO, GWO, Gravitational Search Algorithm, Teaching–Learning-Based Optimization (Akinola et al., 2022). It is important to note that these heuristic models employ a variety of strategies in order to maintain a balance between exploration and exploitation phases. During the exploration phase, the algorithms attempt to determine which area offers the most suitable solutions and exploitation is the process of finding the most appropriate solution in the area that has been determined (Seyyedabbasi et al., 2022). Metaheuristic approaches does not always come with good exploration and good exploitation phases. Hence, hybrid metaheuristic algorithms are proposed in the literature where authors combine the exploration phase of one algorithm with exploitation phase of another algorithm (Zorarpacı and Özel, 2016).

Metaheuristic algorithms mentioned above are developed to solve an optimization problem. The reason why we use those algorithms in feature selection applications is the feature selection problem is an optimization problem. As we mentioned before, aim of the feature selection process is reducing the feature size while enhancing the classification accuracy which makes the feature selection process an optimization problem. Metaheuristic algorithms search through hyper dimensional space to find solution. Since feature selection is a binary optimization problem, we must manipulate the algorithms to solve a binary optimization problem. As a binary optimization problem, feature selection has binary solutions that are bounded to 0 or 1. Binary vectors are used to represent each solution. The length of the vector represents the number of features within the dataset. A value of one indicates that the feature has been selected; otherwise, it has not been selected.

In this thesis, we focused on the metaheuristic algorithms that are inspired by the sea creatures. We wanted to focus on one category since there are tens of metaheuristic algorithms in the literature. Observing the behaviours of sea creatures are harder than other creatures that live on land. That is why finding a hunting pattern or gathering pattern of sea creatures requires a great effort and cost. Metaheuristic algorithms that are inspired by sea creatures are Artificial Algae Algorithm (AAA) (Uymaz et al., 2015), Coral Reef Optimization (CRO) (Salcedo-Sanz et al., 2014), Dolphin Ecolocation (DE) (Kaveh and Farhoudi, 2013), Emperor Penguins Colony (EPC) (Harifi et al., 2019), Fish School Search (FSS) (Filho et al., 2009), Hydrozon Algorithm (HA) (Tansui and Thammano, 2017), Jellyfish Search Optimizer (JS) (Chou and Truong, 2021), Krill Herd Optimization Algorithm (KH) (Saremi et al., 2014), Marine Predators Algorithm (MPA) (Faramarzi et al., 2020), Mouth Brooding Fish Algorithm (MBF) (Jahani and Chizari, 2018), Penguins Search Optimization Algorithm (PeSOA) (Gheraibia and Moussaoui, 2013), Pontogammarus Maeoticus Swarm Optimization (PMSO) (Ghojogh and Sharifian, 2018), Remora Optimization Algorithm (ROA) (Jia et al., 2021), Sailfish Optimizer (SFO) (Shadravan et al., 2019), Salp Swarm Algorithm (SSA) (Mirjalili et al., 2017), Sea Lion Optimization (SLnO) (Masadeh et al., 2019), Seagull Optimization Algorithm (SOA) (Dhiman and Kumar, 2019), Shark Smell Optimization (Abedinia et al., 2016) and Whale Optimization Algorithm (WOA) (Mirjalili and Lewis, 2016). Beforementioned algorithms are designed to solve optimization problems. Some of them are used in the feature selection applications (Turkoglu et al., 2022; Abd Elminaam et al., 2021; Ghosh et al., 2020; Dasgupta et al., 2019; Kumar et

al., 2021; Hegazy et al., 2020; Mafarja and Mirjalili, 2018; Rani and Ramyachitra, 2017; Abouelmagd et al., 2021; Dhiman et al., 2021). Feature selection applications of optimization algorithms of sea creatures are popular and new in this area as it is seen from references and Table 1.1.

Table 1.1. Optimization Algorithms Inspired by Sea Creatures

Optimizer Name	Proposed Year	Used For Feature Selection	Proposed Year for Feature Selection Application
AAA	2015	yes	2022
CRO	2014	yes	2021
DE	2013	no	-
EPO	2019	yes	2020
FSS	2009	no	-
HA	2017	no	-
JS	2021	This Thesis	2023
KH	2014	yes	2019
MPA	2020	yes	2021
MBF	2018	no	-
PeSOA	2013	yes	2020
PMSO	2018	no	-
ROA	2021	no	-
SFO	2019	yes	2018
SSA	2017	yes	2017
SLnO	2017	no	-
SOA	2019	yes	2021
Shark Smell Optimization	2016	no	-
WOA	2016	yes	2021

This thesis aims to bring another feature selection application of an optimization algorithm to the literature in this rising trend. JS is used as feature selector for the first time in the literature in this thesis. The performance of JS is compared with metaheuristic algorithms that are inspired

by sea creatures. The algorithm is also compared with most popular metaheuristic algorithms at the end of this thesis.

The rest of this thesis is organized as follows: A literature review of existing studies in the area of nature inspired algorithms that are used in this thesis is presented in Chapter 2. In Section 3, we represent the materials and methods used in this thesis. Chapter 4 show our experiment scenarios and discusses the results, and finally we conclude in Chapter 5.

2. LITERATURE REVIEW

In the literature, variety of studies with nature inspired algorithms have been conducted for feature selection purposes. This section provides a detailed overview about feature selection applications, nature inspired algorithms, how they effect the feature selection process and its results. This section is divided into two, feature selection and nature inspired algorithms.

2.1. Nature Inspired Algorithms

2.1.1. Whale Optimization Algorithm (WOA)

WOA is first introduced by Mirjalili and Lewis (2016) to the literature. Since 2016, WOA is used in variety of engineering problems and proved its value in the literature (Mafarja and Mirjalili, 2018; Sharawi et al., 2017; Zheng et al., 2018; Nasiri and Khiyabani, 2018; Bozorgi and Yazdani, 2019; Chakraborty et al., 2021).

Mafarja and Mirjalili (2018) made some improvements to the WOA in order to enhance the performance of the algorithm and its competences. They used Tournament and Roulette Wheel selection mechanisms for the searching process. Addition to that, to improve the exploitation process of WOA, they used crossover and mutation operators. They tested their work on 18 benchmark datasets and compared their improved algorithm with nature inspired algorithms like PSO, GA and Ant Lion Optimizer (ALO) and filter methods like Information Gain, Spectrum, Fisher Score, Fast Correlation Based Filter and Correlation-based Feature Selection. The study concluded that using a selection mechanism (Tournament or Roulette) instead of random selection gives better results in average selection ratio and classification accuracy. Although they produce better results, the execution time for the algorithm with selection

mechanisms is nearly twice as much as the original WOA. They showed that WOA with Crossover and Mutation operators gives better results in nearly whole datasets in terms of average selection and classification accuracy. In terms of execution time, the algorithm is tied with its improved version. Authors proved that improved version of WOA selects less features in almost all datasets compared to ALO, GA and PSO. The classification accuracy of improved WOA is better in 15 datasets compared to other nature inspired algorithms. On 80% of the datasets, the improved WOA provided the best classification accuracy and the smallest number of features selected. Because improved WOA utilizes simple crossovers and mutations, the most informative features are selected while the irrelevant and redundant features are eliminated. Authors also showed that out of 18 datasets, the improved WOA outperforms filter-based feature selection methods.

Mafarja and Mirjalili (2017) created hybrid models combining WOA with Simulated Annealing (SA). They tested their work with 18 benchmark datasets. They compared the performance results with wrapper feature selection methods. Authors created two hybridization models where SA used to enhance the exploitation ability of WOA in the first model (WOASA-1) and SA used to enhance the final solution (WOASA-2). These two approaches tested the version of WOA where WOA uses Tournament Selection mechanism. These models are called WOASAT-1 and WOASAT-2. Results showed that hybrid methods are much better than the plain WOA. Evaluation metrics were classification accuracy and average selected attributes. In terms of accuracy, hybrid models gave better accuracy results except 2 datasets. Between two hybrid models, WOASA-2 gives better results in more than half of the datasets. In Exactly dataset, WOASA-1 provides 100% accuracy by using only 6 features which is nearly the perfect condition for feature selection objective. Hybrid methods outperform the original WOA about average selected attributes. Except one dataset, hybrid methods select less features than WOA. Using the Tournament Selection mechanism gives the algorithm great advantage about accuracy and average selected attributes. WOASAT-1 and WOASAT-2 outperforms WOA in each dataset in both accuracy and average selected attributes. Modified approaches provide 100% results in two datasets. Modified approaches are almost tied about average selected attributes within each other. The study concluded that the performance of the approaches can be sorted as $WOA < WOAT < WOASA-1 < WOASAT-1 < WOASA-2 < WOASAT-2$.

Nadimi-Shahraki et al. (2022) conducted a study where they enhance the WOA in different ways. They used pooling mechanism and three other search strategies to improve the performance of WOA and named it E-WOA. Using the pooling mechanism, the worst solutions of each iteration are crossed with promising solutions in order to maintain population diversity. Three other strategies are migrating search strategy, preferential selecting search strategy and enriched encircling prey search strategy. They tested their work with well-known optimization algorithms. The authors concluded that the E-WOA is characterized by good local and global search capabilities and is able to achieve superior results when dealing with unimodal and multimodal problems. The E-WOA manages to escape from local optima by striking a balance between two tendencies: exploration and exploitation. They also tested the improved algorithm with medical datasets and also their novel coronavirus dataset. Results showed that the algorithm can detect the optimal subset. In terms of accuracy, binary E-WOA obtains the best score with 96.409%

Sharawi et al. (2017) used WOA as wrapper-method for feature selection. They tested the algorithm with 16 different benchmark datasets and compared the results with PSO and GA. The authors used variety of evaluation criteria to examine the algorithm from every aspect. They used average classification accuracy, worst fitness, mean fitness, standard deviation, average selection size and average F-score. Study shows that with each evaluation criteria, results change but WOA gives better results on 4 out of 6 criteria. WOA got ahead of other algorithms in terms of average classification accuracy, average worst fitness, average best fitness and F-Score. Even though WOA is proven to be good at explorations and exploitation, PSO selects less features in average than WOA.

2.1.2. Emperor Penguin Optimization (EPO)

EPO is first introduced by Harifi et al. (2019) to the literature. Since 2019, WOA is used in variety of engineering problems and proved its value in the literature (Jovanovic et al., 2022; Mansour et al., 2023; Tang et al., 2022; Dhiman et al., 2021; Kalra et al., 2022; Singh et al., 2023).

Dhiman et al. (2021) proposed a binary version of EPO to use it for feature selection purposes. They used V-shaped and S-shaped transfer functions to transform EPO's continuous solutions into binary one. They used 25 test functions to test the efficiency of the binary EPO. They also compared the algorithm with seven binary metaheuristic algorithms which are Binary Particle

Swarm Optimization (BPSO), Binary Bat Algorithm (BBA), Binary Grey Wolf Optimizer (BGWO), Binary Dragonfly Algorithm (BDA), Binary Spotted Hyena Optimizer (BSHO), Binary Gravitational Search Algorithm (BGSA) and Binary Whale Optimization Algorithm (BWOA). Binary EPO produces better performance for all unimodal test functions than other algorithms. In terms of mean and standard deviation, binary EPO outperforms other metaheuristics for most of the multimodal functions. In comparison to other algorithms, BGWO provides better outcomes for MF6. The authors concluded that binary EPO is more stable than the other algorithms. Binary EPO has the best fitness values on CF9 test function. According to the authors, for 88 percent of the benchmark functions, binary EPO outperforms other metaheuristic algorithms. Study is proven that intensification and diversification are balanced within binary EPO. The study also examines the feature selection properties of binary EPO. They used 12 benchmark datasets from UCI repository. Out of 12 datasets, binary EPO outperforms other metaheuristic algorithms on 10 datasets based on fitness value. The other datasets gave best fitness scores with BBA but binary ECO was the second successful algorithm in this situation. In 8 datasets, binary EPO selects less features than others. Hence, the results show that in terms of both fitness and selected feature size, binary EPO dominates other algorithms.

Kalra et al. (2022) proposed the binary version of EPO (BEPO) to find solutions for binary-nature problems. They used transform functions to map continuous search space to binary search space. The authors used 8 benchmark datasets with mostly low number of features. They compared their algorithm with BGWO, BGSA, BPSO and BBA. According to the average best fitness value for all datasets, BEPO produces higher performances than other algorithms for feature selection tasks. For the learning speed (convergence), authors proved that BEPO learns faster than other algorithms. They also mentioned that one disadvantage of BEPO can be slow convergence when dealing with large dimensional datasets because of the use of transfer functions.

Alweshah et al. (2023) conducted a study where they used a IoT dataset and used EPO as feature selector to identify the network attacks or anomalous behaviors on IoT networks. Since IoT devices receive thousands of data, the dataset is large in term of records. There are 9 datasets with 115 features. Two of the datasets have more than one million records. They used evaluation metrics like accuracy, fitness value, selection size, precision, recall, F-measure and

convergence speed. The study concluded that F-score values are higher than accuracy values for each dataset. The authors indicated that the reason why F-score values are higher than accuracy values is real negatives. Their selection size table shows that EPO reduced the selection size to the mean of 38 features with high accuracy. They also showed that EPO converges under 20 iterations on all of the datasets which can make the execution time 5 times faster.

2.1.3. Marine Predators Algorithm (MPA)

MPA is first introduced by Faramarzi et al. (2020) to the literature. Since 2020, MPA is used in variety of engineering problems and proved its value in the literature (Abdel-Basset et al., 2021; Soliman et al., 2020; Al-Qaness et al., 2020; Ramezani et al., 2021; Rai et al., 2023; Abd Elminaam et al., 2021).

Jia et al. (2022) proposed an improved version of MPA called co-evolutionary cultural mechanism-based marine predators algorithm (CECMPA). As a result of this mechanism, search agents in different spaces are able to exchange knowledge. Study tested CECMPA with 12 benchmark datasets from UCI repository and one COVID-19 dataset. CECMPA is compared with other 5 feature selection methods including PSO, GOA (Aljarah et al., 2018), SSA, WOASAT-2 (Mafarja and Mirjalili, 2017) and SOA-TEO3 (Jia et al., 2019). They used 5 evaluation metrics which are classification accuracy, feature size, fitness value, computational time and p-value. CECMPA outperforms other methods 70% of the datasets in terms of classification accuracy. The improved algorithm 100% accuracy on Divorce Predictors dataset. CECMPA keeps its domination on feature size and fitness value. Even though other methods get better results on 4 datasets, feature size of CECMPA is quite close to the best ones. On the exterior of benchmark datasets, the authors used a COVID-19 dataset to test the accuracy and feature size of the improved model. The results showed that CECMPA outperform all other algorithms in terms of classification accuracy with 92.58% and feature size with 4 on COVID-19 dataset.

Abd Elminaam et al. (2021) proposed a hybrid algorithm between MPA and KNN. Newly proposed MPA-KNN algorithm evaluated using 18 UCI medical benchmark datasets. The algorithm is compared with 8 known metaheuristic wrapper-based approaches. The algorithms they used for comparison are GWO, MothFlame Optimization Algorithm (MFO), Sine Cosine Algorithm (SCA), WOA, SSA, Butterfly Optimization Algorithm (BFO), and Harris Hawks

Optimization (HHO). The remarkable part of this study that it's the first study that uses MPA as feature selector. They used average fitness, standard deviation, accuracy, average accuracy, sensitivity and true negative rate (TNR) as evaluation metrics. Results show that MPA-KNN had the best accuracy rate in 77.7% of the datasets. Except one algorithm, other algorithms had 100% accuracy rate on some datasets but MPA-KNN had the greatest accuracy rates. MPA-KNN ranked first algorithm in standard deviation too by having the less scores on 15 datasets. The study showed that MPA-KNN provides a higher specificity rate than other algorithms in about 83.3% of the datasets. For the feature size evaluation, results must examine with detail because balance must be ensured between feature size and accuracy. MPA-KNN did not show the best performance about selecting the least features but the improved algorithm is not far away from the best performing method.

Abd Elaziz et al. (2022) presented a hybrid model of MPA named MPASCA. Sine-cosine algorithm is used with MPA to improve the search ability of the algorithm. The authors evaluated the performance of the algorithm with 18 datasets from UCI and a metabolomics dataset. The metabolomics dataset is used for evaluation to use the hybrid algorithm in a real-world problem. MPASCA is compared with state-of-the-art metaheuristics to prove its efficiency. Out of 18 datasets, 4 datasets had more than 2000 features. 6 evaluation metrics are used: accuracy, fitness value, maximum of fitness value, minimum of fitness value, selected feature and standard deviation. MPASCA is compared with Henry gas solubility optimization (HGSO), HHO, WOA, GWO, GA, SSA and SCA. According to the study, MPASCA obtained the lowest fitness values in 33% of all datasets. It was found that MPASCA was the most stable algorithm for measuring standard deviation; it recorded the lowest standard deviation values in four databases and the lowest standard deviation average across all databases. The study indicated that MPASCA yielded the smallest feature numbers in 7 out of 22 datasets, followed by SCA with the smallest feature numbers in 8 out of 22 datasets. According to the study, MPASCA, 78% of all datasets were able to achieve the highest level of accuracy.

2.1.4. Salp Swarm Algorithm (SSA)

SSA is first introduced by Mirjalili et al. (2017) to the literature. Since 2017, SSA is used in variety of engineering problems and proved its value in the literature (Singh et al. 2022; Rizk-Allah et al. 2019; Sayed et al. 2018; Salgotra et al. 2021; Qais et al. 2019; Zivkovic et al. 2022).

Hegazy et al. (2020) proposed an improved version of SSA and named it improved salp swarm algorithm (ISSA). Authors aimed to improve the structure of SSA by adding a new control parameter to enhance accuracy, reliability and convergence speed. They used 23 datasets from UCI to test the performance of ISSA. Study compared the improved algorithm with basic SSA and other 4 swarm methods like PSO, GA, ALO, GWO. Study divided datasets into three based on their feature sizes. They had 6 small datasets (feature size is less than 20), 10 medium datasets (feature size is between 20 and 100) and 7 large datasets (feature size is more than 100). They used evaluation metrics like classification accuracy, fitness values, average selection size and average execution time. Study showed that based on a comparison of 13 datasets out of 23, the proposed ISSA algorithm outperforms the standard SSA algorithm and other optimization methods about fitness score. The authors indicated that ISSA selects minimum number of features in 17 out of 23 datasets. Especially when dealing with large datasets, ISSA produces superior results compared to other optimizers. It can be observed from the study that the performance of ISSA is better than the other algorithms in 15 out of 23 datasets.

Tubishat et al. (2021) proposed a new version of SSA where they change the equation of salps' position update and add a new local search algorithm. They called the proposed algorithm Dynamic Salp Swarm Algorithm (DSSA). Authors changed the location by using Singer's chaotic map. The reason of this upgrade is to improve SSA's solution diversity. They added the local search algorithm (LSA) to improve SSA exploitation. DSSA is tested with 20 benchmark datasets from UCI and 3 Hadith datasets. The authors compared DSSA with 4 well-known optimization algorithms like PSO, GA, ALO and Grass-hopper Optimization Algorithm (GOA). They evaluated the algorithm based on classification accuracy, fitness value, number of selected features and convergence speed. Study proved that on all 23 datasets, DSSA outperformed all other optimization techniques in terms of classification accuracy. Authors indicated that DSSA was surpassed due to its ability to escape local optima using the LSA and its improvement in solution diversity through the use of the Singer's chaotic map and the new update equation. It is also showed that DSSA outperformed the standard SSA in all 23 datasets based on the number of selected features. DSSA outperformed other optimization algorithms in 21 out of 23 datasets based on the number of selected features.

Ibrahim et al. (2019) proposed a new hybrid optimization algorithm combining SSA and PSO. They called the new algorithm SSAPSO. They aimed to improve the exploration and the exploitation steps by combining these two algorithms. They compared the SSAPSO with other similar approaches. The authors also evaluated the performance of the SSAPSO using 15 benchmark optimization functions and 10 benchmark UCI datasets. They used several evaluation metrics including mean fitness, maximum fitness, minimum fitness, standard deviation, success rate, acceleration rate, time average, RMSE and selection ratio. Study concluded that SSAPSO has better performance than the other algorithms based on mean fitness, best fitness, worst fitness and standard deviation. Only metric where SSAPSO was worse than SSA was time average. Since PSO consumes more time in the exploration phase, the improved algorithm worked slower. In the feature selection evaluation of the study, it concluded that SSAPSO has the lowest computation time on half of the datasets. It also concluded that SSAPSO has the lowest RMSE value on 60% of the datasets. In addition, SSAPSO achieved the highest reduction of the features in all datasets except for the Breastcancer dataset, where it ranked second after SSA.

2.1.5. Ant Colony Optimization (ACO)

ACO is first was brought to the literature in 1990's by Marco Dorigo and his colleagues (Colormi et al., 1991; Dorigo, 1992; Dorigo et al., 1996). Since 90s, ACO is used in variety of engineering problems and proved its value in the literature (Rokbani et al., 2021; Zhou et al., 2022; Deng et al., 2019; Wang and Han, 2021; Karimi et al., 2023; Paniri et al., 2020).

Aghdam et al. (2009) studied ACO with text dataset. Text datasets are known for their high dimensionality of the feature space. They compared the algorithm with one metaheuristic approach and two statistical approaches. They used GA, IG and CHI for the comparisons. The authors used Reuters-21578 dataset which has 5785 training documents and 2299 test documents. The study uses precision and recall as its evaluation metrics. Depending on the table presented on the study, ACO outperformed the other methods on average. The graphics of the study also showed that best categorization performance is obtained by GA and ACO. It is proven in the study that ACO is faster in locating the optimal solution for the mentioned dataset.

Peng et al. (2018) proposed an improved version of the ACO to prevent the algorithm from falling into a local optimum. The authors designed a new pheromone updating rule to improve the algorithm. They used KDD CUP99 dataset to test their method. The dataset has 41 features

which makes it high dimensional dataset. Proposed algorithm is compared with MOACO (Lutvica and Konjicija et al., 2017) and BPPA-ACO (Meier et al., 2017) algorithms. Figures in the study showed that proposed version of ACO converged at the 12th iteration with the 98% accuracy. MOACO and BPPA-ACO converge at 22nd and 26th iterations with a accuracy value lower than improved ACO, respectively. In comparison to MOACO and BPPA-ACO algorithms, the improved ACO algorithm provided more accurate feature selection. In addition, the improved algorithm can also reduce the rate of false alarms generated by the classifier. Compared to the other two algorithms, the improved algorithm exhibits lower false alarm rates, and its rate of increase is slower than the other two.

Ghosh et al. (2020) proposed an APO algorithm which they used a wrapper and filter approaches together for feature selection. They introduced subset evaluation using a filter method instead of using a wrapper method. They developed a memory where they keep the best ants. Authors used real-life datasets, benchmark datasets from UCI and NIPS2003 FS dataset. Study divided the datasets into three: small, medium and large datasets. Out of 10 datasets, 3 datasets were small (feature size is less than 10), 4 datasets were medium (feature size is between 10 and 100), 3 datasets were large (feature size is more than 100). They compared the algorithm with Bee Colony Optimizer Feature Selection (BCOFS) (Forsati et al., 2012), histogram-based multi-objective GA (HMOGA) (Ghosh et al., 2018), mutation-enhanced PSO (ME-PSO) (Wei et al., 2017) and simulated annealing (SA). From the table provided in the study, it resulted that for 8 out of 10 datasets, WFACOFS achieved the best accuracy and for the rest of the datasets, it achieved second best accuracy. It can be concluded that WFACOFS made good impression for medium and small size datasets. These results are all better than other ACO versions in terms of accuracy. The algorithm also outperforms feature selection algorithms like BCOFS, ME-PSO, HMOGA and SA. For larger datasets, it performs poorly but it is still better than other ACO algorithms. Even the classification accuracy might not be perfect, WFACOFS reduced the feature size by a significant margin.

2.1.6. Sailfish Optimizer (SFO)

SFO is first introduced by Shadravan et al. (2019) to the literature. Since 2019, SFO is used in variety of engineering problems and proved its value in the literature (Zhang and Mo, 2021; Rajamoorthy et al., 2022; Ikram et al., 2023; Naji et al., 2022; Rai and Das, 2022).

Ghosh et al. (2020) proposed an improved version of SFO with β -hill climbing. They first converted the algorithm to Binary Sailfish Optimizer (BSFO) to map the continuous search space to binary one using Sigmoid transfer function. They combined adaptive β -hill climbing (A β HC) algorithm with BSFO to enhance the exploitation ability of BSFO. Proposed BSF and A β BSF algorithms are tested on 18 benchmark datasets and compared with other meta heuristic algorithms. Study showed that BSF achieved more than %90 accuracy in 61.1% of the datasets and A β BSF achieved more than %90 accuracy in 83.33% of the datasets. Additionally, A β BSF achieved 100% accuracy for 10 datasets. It is proven in the study that A β HC algorithm helps BSF algorithm to investigate the search space. The authors compared the algorithm with 10 state-of-the-art methods such as GA, PSO, ALO, GSA, serial grey-whale optimizer (HSGW), random switching grey-whale optimizer (RSGW), adaptive switching grey-whale optimizer (ASGW), WOASAT-2, WOA-CM and BGWOPSO (Al-Tashi et al., 2019). It is observed in the study that A β BSF outperforms other algorithms by giving the best accuracy score in 16 datasets (88.9%). The authors emphasized that A β BSF outperforms BSF on all cases. About feature size, A β BSF selected the lowest number of features in 6 datasets which is quite low for this particular evaluation metric.

Even though SFO is used in variety of engineering problem, there are not many feature selection application but one (Ghosh et al., 2020).

2.1.7. Genetic Algorithm (GA)

GA is first introduced by Holland (1975) to the literature. Since 1975, GA is used in variety of engineering problems and proved its value in the literature (Weile and Michielssen, 1997; Sivanandam et al., 2008; Johnson and Rahmat-Samii 1994; Wetter and Wright, 2003; Gen and Cheng, 1999; Liang et al., 2020).

Sayed et al. (2019) proposed a new ensemble method where they combined t-test and GA. They used the nested version of GA called Nested-GA to find the smallest number of features with the highest accuracy. Nested-GA has two different GAs which are Outer Genetic Algorithm (OGA-SVM) and Inner Genetic Algorithm (IGA-NNW). The authors used DNA Methylation dataset and an independent dataset to evaluate the performance of the algorithms. Results in the study showed that Nested-GA enhances the classification accuracy using fewer features. They also made an experiment where they tested the unselected features and found that classification

accuracy is fell down to 0.553. The experiment had come to a conclusion that replacing the selected (important) features by less important ones leaded to crash on classification accuracy.

Maleki et al. (2021) proposed an GA with KNN where they determined the k value using an experimental procedure. They used a lung cancer dataset which has 23 features and 1000 samples. The classes on the dataset were divided into 3: low level, medium level and high level. They compared the proposed method with Decision Tree (DT), KNN with k being 10 and KNN with k being 6. The results showed that accuracies of DT, KNN with k being 10, KNN with k being 6 and proposed method are 95.4%, 96.4%, 99.8% and 100%, respectively.

Reddy et al. (2020) proposed an adaptive genetic algorithm with fuzzy logic (AGAFL) model for the diagnostic of heart disease at early stages. The model is tested on UCI heart diseases dataset. They used 3 different heart disease datasets named Cleveland, Hungarian and Switzerland. The authors used sensitivity, specificity, execution time and accuracy as evaluation metrics. The proposed method is compared with LPP+RBFL (Reddy et al., 2016) and RS+FL (Rough Set and Fuzzy Logic Classifier) algorithms. Results of the study showed that accuracy of proposed method outperforms other methods for each dataset. Table mentioned on the study showed that accuracy values for the proposed method are %90, %91 and %89 for Cleveland, Hungarian and Switzerland datasets, respectively. Even though classification accuracy showed its dominance, it is not the case for sensivity metric. RS+FL method outperforms proposed method on Cleveland dataset but the proposed method outperforms the other two methods on the other datasets. The study concluded that AGAFL outperforms other two methods in specificity metric. For the Hungarian dataset, specificity scores are %88, %38 and %35 for the proposed, LPP+RBFL and RS+FL algorithms, respectively. There is a big gap between the specificity scores.

2.1.8. Partical Swarm Optimization (PSO)

PSO is first introduced by Kennedy and Eberhart (1995) to the literature. Since 1995, GA is used in variety of engineering problems and proved its value in the literature (Wang et al., 2021; Zhang et al., 2020; Zeng et al., 2020; Pozna et al., 2022; Liu et al., 2019).

Liu et al. (2011) proposed an improved version of the PSO which they called IPS. Their approach is based on a modified Multi-Swarm PSO (MSPSO) for solving discrete problems, which consists of a number of subswarms and a multi-swarm scheduler that is capable of

monitoring and controlling each subswarm each based on the rules. Their proposed approach is combination of MSPSO and SVM. Study compares IPS with GA based, PSO based and grid search based methods. The performance of the method is evaluated using 10 benchmark datasets taken from UCI and StatLog databases. Study showed that all datasets give better results with feature selection process done by IPS. Based on the results given, the accuracy rates of classification using the IFS approach with and without feature selection were consistently higher than those of grid search. For all datasets, IFS with feature selection was more accurate than GA + SVM, while GA + SVM was more accurate than IFS without feature selection. The authors concluded that eliminating the irrelevant features is important to increase the accuracy rates. PSO+SVM and GA+SVM achieved faster convergence rates than IFS, but their classification accuracy was lower than IFS as a result.

Lin et al. (2008) used PSO for parameter determination. The study compares the classification accuracy performances of several algorithms such as PSO+SVM, NSVM (Newton SVM), SVM, LSVM (Lagrangina SVM), SVM with Guassian kernel, SVM with Polynominal kernel and SVM with Sigmoid kernel. They tested the algorithms with 17 benchmark UCI dataset. Study concluded that given abovementioned algorithms, PSO+SVM reaches the highest accuracy scores for each dataset. Another comparison made in the study that PSO+SVM algorithm compared with GA+SVM. These two algorithms compared with using feature selection and not using feature selection. Results about these two algorithms showed that 6 out of 10 datasets, PSO+SVM algorithm outperforms GA+SVM algorithm with feature selection process. Without feature selection, the results were tied.

2.1.9. Jellyfish Search Optimization (JS)

JS is first introduced by Chou and Truong (2020) to the literature. Since 2020, JS is used in variety of engineering problems and proved its value in the literature (Farhat et al., 2021; Choue and Molla, 2022; Abdel-Basset et al., 2021; Chou and Truong et al., 2020; Kaveh et al., 2021; Alam et al., 2021).

There are not any study of JS as feature selector in the literature. The purpose of this thesis is to use JS as feature selector for the first time in the literature.

3. MATERIAL AND METHODS

We aimed to compare the performances of the optimization algorithms inspired by the sea creatures in this thesis. For this purpose, we selected 6 different optimization algorithms including EPO, MPA, SFO, SSA, WOA and JS. In order to evaluate the performance of these algorithms, we also compared them with well-known optimization algorithms including ACO, GA and PSO. There are studies in the literature where EPO, MPA, SFO, SSA and WOA are used as feature selectors (Dhiman et al., 2021; Elminaam et al., 2021; Ghosh et al., 2020; Hegazy et al., 2020; Mafarja and Mirjalili, 2018). Another aim of this thesis is to use JS as feature selector for the first time in the literature. JS is not used as feature selector in feature selection applications in any study in the literature. Datasets are splitted using cross validation, 70% for train and %30 for test.

3.1. Dataset

In order to validate the performance of the algorithms given below, twenty-two benchmark data sets are used from the UCI repository. Table 3.1 shows the number of instances, feature size and classes of the datasets. Based on the Table 3.1, data sets with different instances and features are chosen in order to evaluate the algorithm's performance in search spaces of low, medium, and high sizes.

Table 3.1 Details of the datasets

Dataset Name	No. Of Features	No. Of Instances	No. Of Classes
breast	30	569	2
breastcancer	9	699	2
car	6	1728	4
clean1	166	476	2
clean2	166	6598	2
colon	2000	62	2
credit-g	20	1000	2
dermatology	34	366	6
ecoli	7	336	8
glass	9	214	6
heart-c	13	303	2

ionosphere	34	351	2
kr-vs-kp	36	3196	2
leukemia	7129	72	2
lymphography	18	148	4
sonar	60	208	2
spect	22	267	2
tic-tac-toe	9	958	2
waveform	40	5000	3
wine	13	178	3
yeast	8	1484	10
zoo	16	101	7

3.2. Classifiers

3.2.1. Naïve Bayes

The Naive Bayes algorithm is one of the probability-based ML algorithms that has been applied in the literature for many issues in fields like health, engineering and business and has given efficient results (Ren et al., 2009). Based on Bayes' theorem, it calculates the probability for each attribute in the dataset in order to show which class each belongs to. A naive Bayes classifier greatly simplifies the learning process by assuming that features are independent of class variables given the class variables. A naive Bayes classifier, in its simplest form, assumes that the presence (or absence) of a particular feature within a class has no relation to the presence (or absence) of any other feature within the class (Novakovic, 2010). Although many naive Bayes classifiers are characterized by their naive design and oversimplified assumptions, however, they have worked quite well in a wide range of complex real-world situations. Naïve Bayes classifiers only needs a small portion of training data for classifications. Based on the assumption that independent variables exist, it is only necessary to determine the variance of the variables for each class, and not the entire matrix of covariance. Classifier can be formulized as follows:

$$f_i(x) = \prod_{j=1}^N P(x_j|c_i)P(c_i) \quad (1)$$

where X denotes a feature vector and c represents class labels. Prior probabilities $P(c_i)$ and conditional probabilities $P(x_j|c_i)$ are estimated during the training phase of a classifier. By counting the training samples that fall into class c_i , $P(c_i)$ is calculated by dividing the count by the number of training examples (Novakovic, 2010). Similar to conditional probabilities,

conditional probabilities are estimated by observing the frequency distributions of features x_j within the training subset.

3.2.2. Support Vector Machine (SVM)

SVM is developed to find the hyperplane that produces the best separation boundary between two class points in the vector plane and has been successful on many problems (Widodo and Yang, 2007; Huang et al., 2018; Trafalis and Ince, 2000). In order to determine the maximum distance between points of the two classes, we need to find a hyperplane.

In linearly separable planes, the hyperplane formula is presented for two classes x_1, x_2 that consist of M vector v_i ($i= 1,2,3,\dots,M$) where search space $S \in R^n$ and target $t_i \in \{-1,+1\}$ as follows:

$$f(x) = wv_i + b \quad (2)$$

where v_i vector to be classified, w is the weight vector learned from the train dataset and $b \in R$ is the bias. In complex structures that are not linearly separated, the kernel function is used to transform data into a higher dimensional search space and make it more separable (Liu et al., 2022).

3.2.3. Random Forest (RF)

RF, also known as random decision forest, which is an ensemble learning method that is used to bring a solution to classification and regression problems. The decision forest ensemble is made up of variously constructed classification and regression decision trees. RF applies combination of random subspace methods (RSM) and bagging by performing random feature subspace at each node of the classification and regression tree (CART) (Qiu et al., 2017). The RSM method involves training learning machines on randomly selected sub-spaces within the original input space. By using a median or majority vote, the outputs are merged. At each node of the decision tree, m features are randomly selected from a total of n input features. RF is highly accurate and robust against outliers and noise, and one of its most valuable features is its ability to calculate a large number of internal diagnostic indicators. During the creation of decision forests, data are gathered to estimate the model (Karasu and Altan, 2019). RF provides a unique evaluation of validity and model interpretation that distinguishes it from existing machine learning algorithms. As RF incorporates random sampling and improves ensemble procedures, it delivers greater generalizations and trustworthy estimations.

3.2.4. K-Nearest Neighbors (KNN)

KNN is among the non-parameter and supervised ML algorithm that uses only the train dataset for the classification process and does not need any extra parameters (Taunk et al., 2019). As a first step in performing the classification process, the KNN algorithm uses distance metrics such as cosine and Euclidean distance to measure the distance between train and test samples. Then, the classification process is performed as a mode of classes of k train samples that are closest to the test sample. KNN predicts the class label of a new instance by finding the k closest neighbors from the training set and then assigning the dominant label among the k neighbors.

KNN algorithm is run between the train and test dataset. The Euclidean distance formula between the datapoints are presented in Eq (3):

$$\text{distance}(a, b) = \sqrt{(a - b)^2} \quad (3)$$

where a and b being datapoints from train and test datasets.

3.3. Nature Inspired Algorithms

3.3.1. Emperor Penguin Colony Optimizer

Among the penguin species, the emperor penguin is the largest, growing up to 120 centimeters tall and weighing up to 40 kilograms (Le Maho, 1977). The species is found only in Antarctica, where they breed and raise their young on the sea ice. In addition to being excellent swimmers and divers, Emperor penguins are also capable of staying underwater for a period of up to 20 minutes at a time. The majority of their diet consists of fish, krill, and squid. The Emperor penguins are social animals that live in large colonies, which can number in the thousands. There is a monogamous relationship between them and they mate for life.

Emperor Penguins Colony Optimizer (EPO) is a metaheuristic algorithm that is based on the behaviour of emperor penguins during cold winters. A heterogeneous group of penguins huddling together ensures that everyone is sufficiently warmed. By huddling together, the emperor penguins are able to make the coolest environments into the warm environments. There is much more heat in the center of the huddle. Individuals use heat by spiraling towards the center. As a result of this spiral-like movement, the same amount of heat is generated by each individual.

This algorithm simulates the huddling behaviour of emperor penguins in order to solve optimization problems (Harifi et al., 2019). The EPO algorithm represents each penguin as a possible solution to the optimization problem (Dhiman et al., 2021). Initially, the penguins are randomly distributed throughout the search space. In order to minimize their individual cost functions, the penguins move in a spiralling pattern towards each other. In a cold environment, penguins who are closer to each other receive more heat, allowing them to survive. It is more likely for penguins that are farther apart to die, since they receive less heat.

EPC algorithm can be verbalized as (Harifi et al., 2019):

- 1 - In the initial population of penguins, all penguins emit heat radiation, which causes them to be attracted to each other due to absorption coefficients.
- 2- In terms of body surface area, all penguins are considered equal.
- 3 - Penguins absorb the full heat radiation without considering the effects of the earth's surface or the atmosphere.
- 4 - Penguins radiate heat in a linear manner.
- 5 - Penguins are attracted to each other depending on the amount of heat in the distance between them.
- 6 - A longer distance will result in less heat being received and a shorter distance will result in more heat being received.
- 7 – Penguins move in a spiral way and they deviate with uniform distribution.



Figure 3.1. Sprial-like movement (Harifi et al., 2019).

In the mathematical model, finding an efficient mover from the swarm would be the primary goal. To achieve this, distances between EP (X_{ep}) are calculated, followed by their temperature profiles (θ'). In order to achieve the best value, the positions of other EPs are modified in order to determine the productive mover. The EPO algorithm can be formulated as:

$$\theta' = (\theta - \frac{Iter_{max}}{C - Iter_{max}}) \quad (4)$$

$$\theta = \begin{cases} 0 & \text{if } R > 0.5 \\ 1 & \text{if } R < 0.5 \end{cases} \quad (5)$$

where C is the current iteration and $Iter_{max}$ is the maximum number of iterations. R is a random number between $[0,1]$.

$$\vec{U} = \{M \cdot (\theta' + X_{grid}(accuracy)) \cdot Rand()\} - \theta' \quad (6)$$

$$\vec{V} = Rand() \quad (7)$$

$$X_{grid}(accuracy) = |\vec{X} - \vec{X}_{ep}| \quad (8)$$

Due to their tendency to huddle together in order to maintain temperatures, EPs must be protected from neighbouring collisions by taking precautions. For this reason, it proposes vecotrs that are shown in Eqs. (6) and (7). In the equations given above, M represents the

parameter for movements, $\vec{X}$ is the best optimum solution, $\vec{X}_{ep}$ indicates the locations of other EPs.

$$\vec{D}_{ep} = |\{S(\vec{U}) \cdot \vec{X}(x) - \vec{V} \cdot \vec{X}_{ep}(x)\}| \quad (9)$$

$$S(\vec{U}) = \sqrt{(f \cdot e^{-\frac{c}{v}} - e^{-c})^2} \quad (10)$$

Eqs. (9) and (10) are constructed to estimate the distance between the EP and the best matching search agent ($\vec{D}_{ep}$). The social forces for which the optimal search agent is preceded by EPs are shown by $S()$, with e representing the exponential function. Control parameters f and v are those whose best values lie within the ranges of $[2,12]$ and $[1.5,2]$, respectively. The positions of the EPs are updated according to the optimal agent obtained using Eq (8).

$$\vec{X}_{ep}(x+1) = \vec{X}(x) - \vec{U} \cdot \vec{D}_{ep} \quad (11)$$

As it is well known in the literature, feature selection is a discrete problem. EPO and other optimizers are built to solve continuous optimization problems. These problems have different search spaces, so it is not possible to apply an optimizer for feature selection problem directly. For this purpose, authors use transform functions to turn the continuous problem into a discrete one.

3.3.2. Marine Predators Algorithm

MPA is an algorithm designed to optimize predator-prey encounter rates in marine ecosystems by following the principles that govern optimal foraging strategy and encounter rates of predators and prey. Essentially, the algorithm is designed by Faramarzi et. al. to simulate the behavior of marine predators as they search for prey (Faramarzi et al., 2020). In order to increase their chances of survival, predators must select an optimal strategy for encountering prey in natural environments. Foraging patterns in nature are often dominated by random walks. It is referred to as a stochastic or random process in the literature. Based on the animal's current position and the probability that it will transition to the next position (Tawhid and Dsouza, 2018), the animal's next position can be calculated mathematically. As predators have evolved to survive, these strategies have developed spontaneously in nature.

In this algorithm, the predator is represented by a population of particles, and the prey is represented by a fitness function. With a combination of Lévy flights and Brownian motions,

the particles traverse the search space in a manner that mimics predatory movement (Abd Elminaam et al., 2021). With Levy flights, the particles can move over a larger search space by making long-range jumps (Balakrishnan et al., 2021). In Brownian motion, particles apply fine-tuning to their search for the optimal solution due to short-range movement (Balakrishnan et al., 2022). Particles move at a certain speed based on their velocity ratio. When the velocity ratio is high at the beginning of the optimization, the particles are able to explore the search space more quickly. By lowering the velocity ratio, the particles are able to reach the optimal solution faster at the later stages of the optimization.

3.3.2.1. Levy Flight

A Lévy flight movement is a stochastic process with independent, stationary increments named after French mathematician Paul Lévy. It establishes that a point's motion is stochastic, that displacements in pairwise separated time intervals are distinct, and that displacements in various time intervals of the same length have the same probability distributions (Hussein et al., 2014). The continuous-time equivalent of a random walk is a Lévy process. A random walk of the type Lévy flight is one in which the step sizes are shown by a probability function denoted by the Lévy distribution (power-law tail) as shown in Eq. (12).

$$L(x_j) = |x_j|^{1-\alpha} \quad (12)$$

where x_j is flight length and α is power-law exponent which has values lie within the ranges of (1,2]. As shown in Eq. (13), the integral form of the probability density of the Lévy stable process can be calculated as follows:

$$f_L(x; \alpha, \gamma) = \frac{1}{\pi} \int_0^\infty \exp(-\gamma q^\alpha) \cdot \cos(qx) \cdot dq \quad (13)$$

where α indicates the distribution index and controls the scale properties, while γ determines the scale unit.

3.3.2.2. Brownian Motion

The Brownian motion is the process whereby the probability function of Gaussian distribution determines the length of each step when the mean is equal to zero ($\mu = 1$) and the variance ($\sigma^2 = 1$). Brownian motion formula is showed in Eq. (14).

$$f_B(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \quad (14)$$

By themselves, neither Lévy flight nor Brownian motion are capable of searching a domain globally or locally. A systematic explorer-exploiter framework can be developed by combining each of these strategies and using them properly, which can be more efficient than each strategy alone. Levy flight is usually associated with tiny steps and, on occasion, long jumps. Compared to uniform random search, this characteristic can be used as improved searchability in optimization literature (Yang and Deb, 2013). As a result of its short steps, Levy flight is capable of efficiently and deeply exploring a nearby neighborhood, while its long steps allow it to explore other areas of a domain. Alternatively, walks from Brownian motion can explore distant areas of the neighborhood, but cannot search as accurately and deeply as Lévy strategies. It can be observed that a domain can be explored and exploited more globally and locally if one utilizes the unique characteristics of each random walk. In academic literature, MPA has been extensively demonstrated to be more efficient when exploring and exploiting domains when combining Lévy strategy and Brownian motion features. There are three main phases of MPA based on varying velocity ratios and mimicking natural predator-prey interactions: (1) When prey is moving faster than predator or when there is a high velocity ratio. (2) When predator and prey move at almost the same speed or in a unit velocity ratio. (3) When the predator moves faster than the prey in a low velocity ratio (Faramarzi et al., 2020).

Phase (1): The extracted rules suggest that predators' best strategy when there is a high-velocity ratio is not to move ($v \geq 10$). It is important to explore in this initial iteration of optimization. As in Eq (15), this rule is modeled mathematically.

$$\text{while } Iter < \frac{1}{3} \max Iter$$

$$stepsize_i = \vec{R}_B \otimes (\vec{Elite}_i - \vec{R}_B \otimes \vec{Prey}_i) \quad i = 1, \dots, n \quad (15)$$

$$\vec{Prey}_i = \vec{Prey}_i + P \cdot \vec{R} \otimes stepsize_i$$

Brownian motion is characterized by random numbers in the vector of R_B based on the Normal distribution of Brownian motion. The movement of prey can be imitated by multiplying R_B by Prey. P is a constant number equal to 0.5, and R is a uniform random numbers vector in [0,1]. A high velocity of movement allows high levels of exploration during this phase, which occurs in 1/3 of the iterations.

Phase (2): When both predator and prey are moving simultaneously. When exploration is converted briefly to exploitation, this scenario occurs. Prey is responsible for exploration, while predator is responsible for exploitation. Based on the rule, in the unit velocity ratio ($v \approx 1$), Brownian is the best strategy for the predator if the prey moves in Lévy. The mathematical model for phase 2 is shown in Eq. (16).

while $\frac{1}{3} \maxIter < Iter < \frac{2}{3} \maxIter$

$$\overrightarrow{stepsize}_i = \overrightarrow{R}_L \otimes (\overrightarrow{Elite}_i - R_L \otimes \overrightarrow{Prey}_i) \quad i = 1, \dots, n/2 \quad (16)$$

$$\overrightarrow{Prey}_i = \overrightarrow{Prey}_i + P \cdot \overrightarrow{R} \otimes \overrightarrow{stepsize}_i$$

Based on Lévy distribution, $\overrightarrow{R}_L$ represents Lévy movement as a vector of random numbers. Multiplying $\overrightarrow{R}_L$ and Prey simulates prey movement in Lévy manner, while adding step size to prey position simulates prey movement. For second half of the populations:

$$\overrightarrow{stepsize}_i = \overrightarrow{R}_B \otimes (\overrightarrow{R}_B \otimes \overrightarrow{Elite}_i - \overrightarrow{Prey}_i) \quad i = n/2, \dots, n$$

$$\overrightarrow{Prey}_i = \overrightarrow{Elite}_i + P \cdot CF \otimes \overrightarrow{stepsize}_i \quad (17)$$

where $CF = (1 - \frac{Iter}{\maxIter})^{(2 \frac{Iter}{\maxIter})}$ is considered as an adaptive parameter to control the step size for predator movement. Multiplying $\overrightarrow{R}_B$ simulates Predator's Brownian movement. As a result, Elite simulates the movement of Prey by updating its position according to the movement of predators in Brownian motion.

Phase (3): If predator moves faster than prey in a low-velocity ratio. The optimization process usually ends in this scenario when the exploitation capability is high. Lévy is the best predator strategy at low velocity ratios ($v = 0.1$). This phase can be formulated as:

while $Iter > \frac{2}{3} \maxIter$

$$\overrightarrow{stepsize}_i = \overrightarrow{R}_L \otimes (\overrightarrow{R}_L \otimes \overrightarrow{Elite}_i - \overrightarrow{Prey}_i) \quad i = 1, \dots, n \quad (18)$$

$$\overrightarrow{Prey}_i = \overrightarrow{Elite}_i + P \cdot CF \otimes \overrightarrow{stepsize}_i$$

Adding the step size to Elite position simulates the movement of the predator in Lévy strategy, while multiplying $\vec{R}_L$ and Elite simulates the movement of the prey. Based on rules and points extracted from different literature, MPA simulates predators and prey moving as they do in nature. Predators take these steps to catch prey in these phases. During the lifetime of a predator, the Lévy and Brownian movements should be equal in percentage. Phase 1 shows no movement whatsoever, Phase 2 shows Brownian movement, and Phase 3 shows Lévy movement. Phase 1 involves Brownian movement, while Phase 2 involves Lévy movement for prey.

3.3.2.3. FAD's Effect

Marine predators' behavior can also be affected by environmental factors. A good example of this is the effect of Fish Aggregating Devices (FADs), also known as eddy formation. Sharks spend 80 percent of their time near FADs and 20 percent jumping in different dimensions to find prey, probably because they spend most of their time near FADs (Parouha and Das, 2016). In search space, FADs entrap these points, causing them to be considered local optima. By making longer jumps, these local optima are prevented from stagnating. The mathematical model of FAD's effect can be defined as:

$$\vec{Prey}_i = \begin{cases} \vec{Prey}_i + CF[\vec{X}_{min} + \vec{R} \otimes (\vec{X}_{max} - \vec{X}_{min})] \otimes \vec{U} & \text{if } r \leq FADs \\ \vec{Prey}_i + [FADs(1 - r) + r](\vec{Prey}_{r1} - \vec{Prey}_{r2}) & \text{if } r > FADs \end{cases} \quad (19)$$

Where FADs=0.2 is the probability of FADs effect. $\vec{U}$ is the binary vector with arrays including zero and one. $\vec{X}_{min}$ and $\vec{X}_{max}$ are the vectors that represents the lower and upper bounds. r1 and r2 represents random indexes of prey matrix.

3.3.2.4 Marine Memory

Predators in the ocean have excellent memory for recalling the places where they have foraged successfully. MPA simulates the memory of marine predators. As soon as the Prey and FADs effects have been updated and implemented, this matrix must be evaluated for its suitability to update the Elite. For each solution, the current iteration is compared to the previous one, and if it is more suitable, it replaces the older one. With each iteration, the solution is enhanced, simulating predators returning to prey-rich areas to forage efficiently.

3.3.3. Salp Swarm Algorithm

SSA is a metaheuristic algorithm based on salp swarming behavior introduced by Mirjalili et al. in 2017 (Mirjalili et al., 2017). Salps are marine organisms that are known for their ability to form chains and swim together. In order to simulate the behavior of this algorithm, SSA creates a population of salps, each of which represents a possible solution to an optimization problem. Following the leader salp and one another, they move throughout the search space (Abualigah et al., 2020). Salps are ranked based on their fitness, and the salp with the best fitness is elected as the leader, and will be updated as new solutions are found.

A number of advantages are associated with the SSA algorithm as compared to other metaheuristic algorithms. The first advantage of this algorithm is that it is relatively simple to implement and understand. In addition, it is efficient, and that means that it can usually find an optimal solution to optimization problems in a relatively short amount of time. A third advantage of this algorithm is that it is robust and does not easily get trapped in local optima.

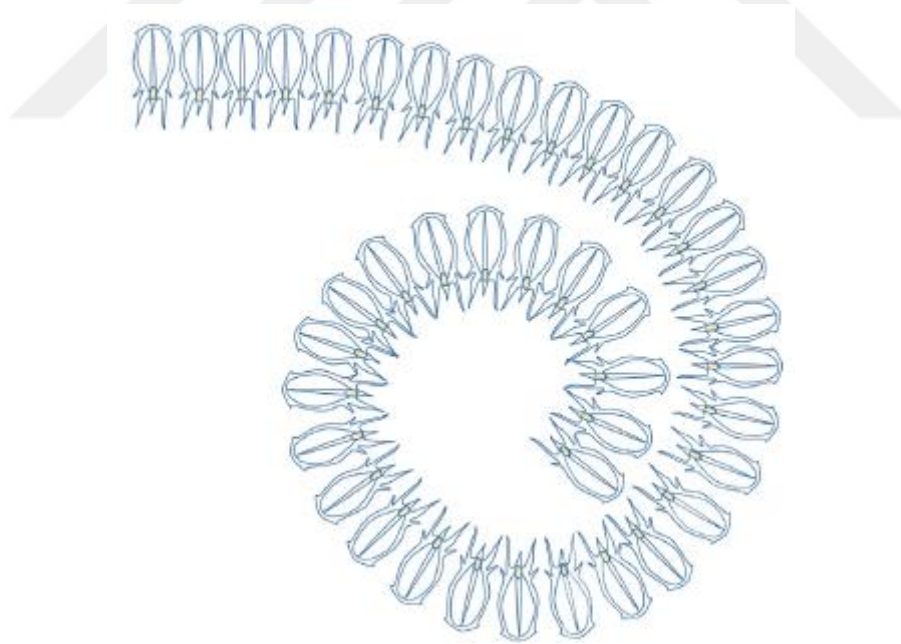


Figure 3.2. Example of Salps Chain (Mirjalili et al., 2017).

Due to its ability to balance exploration and exploitation, SSA has proven effective for feature selection. The process of exploration involves looking for new, promising solutions, while the process of exploitation involves refining existing ones. In order to solve feature selection problems, SSA balances these two processes effectively. Similarly to other swarm-based

techniques, the position of salps is defined in an n -dimensional search space where n is the number of variables of a given problem (Hegazy et al., 2020). As a result, the salps' positions are stored in a matrix called x . The swarm also targets a food source called F in the search space. The leader's position can be formulized as:

$$x_j^1 = \begin{cases} F_j + c_1 \left((ub_j - lb_j)c_2 + lb_j \right) & c_3 \geq 0 \\ F_j - c_1 \left((ub_j - lb_j)c_2 + lb_j \right) & c_3 < 0 \end{cases} \quad (20)$$

where x_j^1 shows the position of the leader in the j th dimension, F_j is the position of the food, ub_j and lb_j indicates upper and lower bound of j th dimension, c_1, c_2 and c_3 are random numbers. From all these coefficients, c_1 is the most important one because it creates a balance between exploration and exploitation phases.

$$c_1 = 2e^{-\left(\frac{4l}{L}\right)^2} \quad (21)$$

where l is the current iteration and L is the maximum number of iterations. Coefficients c_2 and c_3 changes the direction of the position and the step size based on their value which are between $[0,1]$.

Followers of the leader changes their position based on Newton's law of motion:

$$x_j^i = \frac{1}{2}at^2 + \vartheta_0 t \quad (22)$$

where $i \geq 2$ and x_j^i shows the position of the i th follower salp. As it is known from Newton's law of motion, t is time, ϑ_0 is initial speed, a is acceleration and ϑ is velocity. Acceleration and velocity can be calculated as $a = \vartheta_{final}/\vartheta_0$ and $\vartheta = X - X_0/t$. Since iterations in optimization problems represents time in these equations and considering $\vartheta_0 = 0$, Eq. (23) can be expressed as follows:

$$x_j^i = \frac{1}{2}(x_j^i + x_j^{i-1}) \quad (23)$$

where $i \geq 2$ and x_j^i shows the position of i th follower salp.

3.3.4. Jellyfish Optimizer

Throughout the world, jellyfish can be found in surface waters as well as deep waters. Their bell and tentacles make up most of their body, which is composed of about 95% water. Bells and tentacles are used for locomotion and capturing prey, respectively. Jellyfish have stinging cells in their tentacles that they use to paralyze their prey.

There are over 2,000 species of jellyfish that are known to exist, and they come in a wide variety of colors, shapes, and sizes. Jellyfish come in all sizes, from very small ones to those with a diameter of several meter (Bastian et al., 2014). There is no doubt that jellyfish play an important role in the marine ecosystem. There are many other marine animals that are dependent upon them for food, including fish, turtles, and seabirds, and thus they play an important role in controlling the population of those animals in the ocean.

There are several jellyfish that use their tentacles to sting their prey, and then release a venom that paralyzes them. Unlike some creatures, they do not attack others. However, those who get too close to them or touch them may end up being killed by their sting. The sting of a jellyfish can be painful, but it is not usually fatal. Despite this, some jellyfish, such as the box jellyfish, have very potent venom that can be fatal to humans (Chou and Truong, 2021). As a result of a typical sting, one may experience pain, red marks, itchiness, numbness, or tingling.

Jellyfish have features that control their movements. In order to move forward, their undersides close like an umbrella, pushing water out of their body. Even though they are capable of drifting in the water, they are mostly swept along by the tide and currents. In favorable conditions, jellyfish will form a swarm of jellyfish, and the grouping of jellyfish together is known as a jellyfish bloom. It is important to note that jellyfish are weak swimmers, and so their orientation with respect to currents is crucial to the maintenance of blooms as well as in ensuring that they do not become stranded in the water (Youssef et al., 2021).

In addition to their ability to appear almost everywhere in the ocean, jellyfish also move within the swarm and follow ocean currents to form jellyfish blooms. Jellyfish visit varying sites for food; therefore, it is possible to identify the best location by comparing the proportions of food at each location. As a result, Chou and Truong (2021) developed an algorithm that is inspired

by the search behavior and movement of jellyfish in the ocean. Given three rules can make the algorithm work (Chou and Truong, 2021):

1- Jellyfish follow ocean currents or move within swarms, and a "time control mechanism" controls their movements.

2- Jellyfish move in the ocean in search of food. The greater the amount of food available in a location, the more likely they will be to flock there in order to feed.

3- The amount of food that is found depends on the location as well as the objective function associated with the location.

Mathematical model for the algorithm can be formulized as follows:

3.3.4.1. Ocean Current

Jellyfish are attracted to the sea current because it contains large amounts of nutrients, which is why it attracts so many of them. The direction of the ocean current ($\overrightarrow{trend}$) is determined by averaging all the vectors from each jellyfish in the ocean to jellyfish that is currently in the best location. Ocean current can be calculated as:

$$\overrightarrow{trend} = \frac{1}{n_{pop}} \sum \overrightarrow{trend}_i = \frac{1}{n_{pop}} \sum (X^* - e_c X_i) = X^* - e_c \frac{\sum X_i}{n_{pop}} = X^* - e_c \mu \quad (24)$$

where n_{pop} is the number of jellyfish, X^* is the best jellyfish, e_c is the attraction factor, μ is the average location of all jellyfish. If we determine the difference between the current best location of the jellyfish and the mean location of all jellyfish ($e_c \mu$) as df , $\overrightarrow{trend}$ can be formulated as:

$$\overrightarrow{trend} = X^* - df \quad (25)$$

Assuming that jellyfish are disposed in all dimensions of the ocean according to a normal spatial distribution, df can be written as:

$$df = \beta \cdot \sigma \cdot rand^f(0,1) \quad (26)$$

where β is a distribution coefficient that is set to 3 (Chou and Truong, 2021). If we set σ as $rand^\alpha(0,1) \cdot \mu$, df can be rewritten as:

$$df = \beta \cdot rand(0,1) \cdot \mu \quad (27)$$

With given information above, the new location of each jellyfish is given by:

$$X_i(t+1) = X_i(t) + rand(0,1) + (X^* \beta \cdot rand(0,1) \cdot \mu) \quad (28)$$

3.3.4.2. Jellyfish Swarm

The motions of jellyfish in swarms can be either passive (type A) or active (type B); depending on the situation. As the jellyfish swarm has just formed, most of the jellyfish exhibit type A motion at the beginning of the process. There is an increasing tendency of them to exhibit type B motion with time. An example of type A motion is jellyfish moving around their own locations and the updated locations of each jellyfish is given by:

$$X_i(t+1) = X_i(t) + \gamma \cdot rand(0,1) \cdot (U_b - L_b) \quad (29)$$

where L_b and U_b are the lower and upper bound of the search spaces, respectively. $\gamma > 0$ is a motion coefficient that is set to 0.1. Simulating type B motion involves selecting a jellyfish other than the one of interest at random and determining the direction of movement using a vector between the interest jellyfish and the selected jellyfish. Selected jellyfish move toward the jellyfish of interest when their food supply exceeds their food supply at the jellyfish of interest's location; jellyfish of interest move away from selected jellyfish if the selected jellyfish's food supply is less than that of the interested jellyfish. To find food in a swarm, each jellyfish moves toward the better direction. Local search space is effectively exploited through this movement:

$$\overrightarrow{Step} = X_i(t+1) - X_i(t) \text{ where } \overrightarrow{Step} = rand(0,1) \cdot \overrightarrow{Direction} \quad (30)$$

$$\overrightarrow{Direction} = \begin{cases} X_j(t) - X_i(t) & \text{if } f(X_i) \geq f(X_j) \\ X_i(t) - X_j(t) & \text{if } f(X_i) < f(X_j) \end{cases} \quad (31)$$

where f is an objective function of location X . Hence,

$$X_i(t + 1) = X_i(t) + \overrightarrow{Step} \quad (32)$$

3.3.4.3. Time Control Mechanism

Ocean currents contain a large amount of nutritious food, which attracts jellyfish. Eventually, a swarm of jellyfish is formed as more jellyfish gather together. Jellyfish in the swarm move toward another ocean current as the temperature or wind changes, creating another jellyfish swarm. Within a jellyfish swarm, jellyfish switch between types A (passive movements) and B (active movements). In the beginning, type A is more popular than type B, but as time goes on, type B becomes more and more popular (Alam et al., 2021). Time control mechanism includes a time control function $c(t)$ and a constant C_0 that are used to control the movement of jellyfish between following the ocean current and moving inside the jellyfish swarm.

$$c(t) = \left| \left(1 - \frac{t}{Max_{iter}} \right) \cdot (2 \cdot rand(0,1) - 1) \right| \quad (33)$$

where t is the iteration number and Max_{iter} is the maximum number of iterations. Jellyfish follow the ocean current when time control function value exceeds C_0 . They move inside the swarm when its value is less than C_0 . There is no exact value for C_0 , and the time control varies randomly between zero and one. In this case, C is set to 0.5, which is the mean of zero and one.

3.3.4.4. Binary Jellyfish Optimizer

Based on the type of variables (search space), there are two types of real problems: continuous and discrete. There is a continuous search space for a continuous problem, as its name implies. Heuristic algorithms can move easily over the search space because the variables in the problem have continuous ranges. Contrary to this, when dealing with discrete problems, the continuity of parameters is no longer relevant. It is possible to think of a discrete search space as a hypercube. Metaheuristic algorithms require the agents to jump from one corner of the hypercube to another. In order to achieve these jumps, bits are usually flipped in a certain order. Binary problems cannot be solved directly by a continuous version of a heuristic algorithm. Since metaheuristic algorithms are designed to solve optimizations in the continuous search space, we need to convert the algorithm into binary one. To achieve this goal, the search space of the solution and the moving mechanism of the algorithm needs to change. A sample of a solution is given in Figure X.

1	0	0	1	1	0
---	---	---	---	---	---

Figure 3.3. Sample of solution in binary search space.

Figure 3.3 represents a sample of feature subset solution for a dataset of 6 attributes. Features that are 1 is used in classification process while features that are 0 are not. The sample shows that 3 features are selected (1^{st} , 4^{th} , 5^{th}) out of 6 features. The change of the position is depend on the transfer function. The big change in the position indicates that the search agent is not close to the best solution and it represents the exploration phase of the algorithm. On the contrary, small changes in the position indicate that search agent is close to the best solution and it represents the exploitation phase of the algorithm (Saremi et al., 2015). The change of position specify a different process in the binary search space. Since the agent cannot move as it does in the continuous search space, feature subset solution flips its binary values to reach the best solution.

3.3.5. Sailfish Optimizer

Sailfish are large, elongated fish that can grow up to 3 meters long and weigh up to 90 kilograms. World-wide, they are most commonly found in tropical and subtropical waters. Speed and agility are two of the most notable characteristics of sailfish. Their bill is long and pointed, and they spear their prey with it using their long, pointed bill. As a part of the marine ecosystem, sailfish play an important role to control the population of smaller fish, they are also an important food resource for larger predators, such as sharks and dolphins, which are responsible for eating smaller fish. These animals are capable of traveling at 100 kilometers per hour and making sharp turns and jumps (Shadravan et al., 2019).

The sailfish is a predator, and it feeds primarily on small fish, squid, shrimp and sardines. They hunt in groups so that they can save up energy while hunting. It is very challenging for sardines to maneuver and accelerate during the attack of sailfish as they are very maneuverable and fast. Since sailfish are one of the fastest fish on water, sardines cannot swim fast enough to avoid sailfish attack. With its sharp bill, the sailfish either makes a slashing motion, causing several sardines to be injured, or it taps one sardine, causing it to become unstable. While some members of the sailfish group attack the sardines, others rest and save up energy (Issam et al., 2019). Injured sardines will be left out from the prey school and caught by sailfish group.

This optimizer considers sailfish candidates as solutions, and their position in the search space is a critical factor. In the search space, sailfish are dispersed, whereas sardine placements help find the optimal solution. It is possible for Sailfish to search in one, two, three, or hyperdimensional space by changing its location vectors (Kumar et al., 2022). In both sailfish and sardine searches, the algorithm makes every effort to randomize agents' movements. Sailfish can update their position in relation to prey when they attack. In addition, sailfish can adjust their position to occupy unoccupied space around prey schools and imitate surrounding them. In order to avoid sailfish attacks when a member of the prey group (sardine) is damaged, the prey group (sardine) adjusts its position. An elitist process may be necessary since the upgraded locations may have marginal strength. By exploring and exploiting the dynamic attack parameter balancing algorithm, the population elite approach maintains the best individuals for each search.



Figure 3.4. Encircling motion of sailfish (Shadravan et al., 2019).

Notation $P_{SlfElite}^i$ is used for the elite sailfish (the one with the best fitness value) at i^{th} iteration. For the injured sardine which has best fitness value at i^{th} iteration, $P_{SrdInjured}^i$ is used. Positions of sardines and sailfish are updated at each iteration. New position of sailfish at each iteration is updated using positions of elite sailfish and injured sardine as it is shown in Eq. (34):

$$P_{Slf}^{i+1} = P_{SlfElite}^i - \mu_i \cdot (rnd \cdot \frac{P_{SlfElite}^i + P_{SrdInjured}^i}{2} - P_{Slf}^i) \quad (34)$$

where rnd is a random value between 0 and 1 and μ_i is a coefficient which is generated as per Eq. (35):

$$\mu_i = 2 \cdot rnd \cdot PrD - PrD \quad (35)$$

where PrD is the number of preys at each iteration. PrD decreases at each iteration during group hunting.

$$PrD = 1 - \frac{Num_{Slf}}{Num_{Slf} + Num_{Srd}} \quad (36)$$

As Num_{Slf} and Num_{Srd} represents number of sailfish and sardine, respectively, number of sailfish can be calculated as follows:

$$Num_{Slf} = Num_{Srd} \cdot Prcnt \quad (37)$$

where $Prcnt$ represents the percentage of sardines. Initially, the number of sardines are considered to be more than number of sailfish. Position of sardines are updated at each iteration as follows:

$$P_{Srd}^{i+1} = rnd(0,1) \cdot (P_{SlfElite}^i - P_{Srd}^i + ATK) \quad (38)$$

$$ATK = A \cdot (1 - (2 \cdot itr \cdot \kappa)) \quad (39)$$

where ATK represents the attack power of sailfish at iteration i . Number of sardines (γ) and the number of variables of them (δ) can be calculated using ATK parameters as follows:

$$\gamma = Num_{Srd} \cdot ATK \quad (40)$$

$$\delta = v \cdot ATK \quad (41)$$

where v is the number of variables. Sailfish update their position following any sardine that becomes better than any sailfish, and the sardine is eliminated from the population.

3.3.6. Whale Optimizer

Whales are known as the largest mammals on earth. Whales can reach 180 tons of weight and 30 meters in length as adults. One of the rare mammals with this feature is the large size of the females when compared to the males. In spite of the fact that they are generally called whales,

there are different species of whales within them. Whales are often known as predators. Whales cannot fully sleep because they breathe at the surface of the ocean. When needed, only half of their brains sleep. There is a high level of intelligence and emotion in whales. Whales have some cells called spindle cells that are similar to those of humans in certain parts of their brains (Hof and van der Gucht, 2006). People use these cells to make decisions, express emotions, and interact socially. We differ from other living organisms because of our spindle cells. Whales have twice as many spindle cells as humans, and this is the main source of their intelligence. Whales (especially killer whales) are also known to develop dialects through which they communicate. Whales have an interesting social behaviors. It is common for them to live in packs, but they can also live alone. One of the largest whale species is humpback whales (*Megaptera novaeangliae*). The majority of humpback whales' diet consists of krill and small fish, which they feed in groups. Hunting behavior is one of the most interesting characteristics of humpback whales. Humpback whales hunt herds at the surface of the ocean. They create bubbles by exhaling near the surface. In this way, they are able to trap flocks of fish near the surface in the bubble net they expel by moving in a circular fashion and exhaling at regular intervals. During their movement toward the surface, they slowly reduce the diameter of their circular motion as they slowly move towards it (Mafarja and Mirjalili, 2018). When they are compressing their prey, they use a movement called a bubble-net in order to compress their prey. A mathematical model for bubble network motion has been developed by Mirjalili and Lewis (2016) and has been used as an optimization algorithm in a number of engineering problems. Whale Optimization Algorithm is divided into 3 parts while modeling; encircling the prey, bubble-net attacking method and search for prey.

3.3.6.1. Encircling the prey

The point to be reached in general optimization algorithms is considered as the optimum solution. Although the required point is not always reached, a point near this point can be considered as the optimum solution. After determining the solution in the best position in each iteration, the next iteration is formed around this solution. In the Whale Optimization Algorithm, the point to be reached is the hunt. In each iteration, the positions of the other

solutions are updated for the solution that comes closest to the hunt. In Eqs. (42) and (43), the mathematical model of prey encircling behavior is shown:

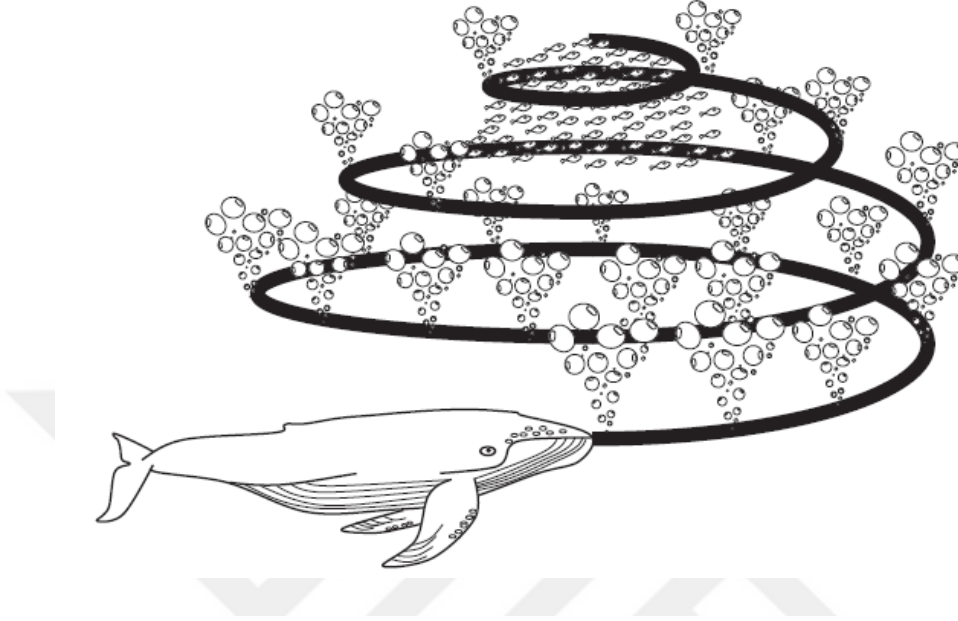


Figure 3.5. Bubble-net mechanism (Mafarja and Mirjalili, 2018)

$$\vec{D} = |\vec{C} \cdot \vec{X}^*(t) - \vec{X}(t)| \quad (42)$$

$$\vec{X}(t+1) = |\vec{X}^*(t) - \vec{A} \cdot \vec{D}| \quad (43)$$

where t is the iteration, A and C is the coefficient vectors and $\vec{X}^*$ is the best solution vector.

3.3.6.2. Bubble-net attacking method

The behavior of moving towards the hunt includes the part after the bubble popping action. In this movement, it expresses the narrowing of the circle formed by the whales and moving in a spiral. The circle narrowing behavior is modeled in Eqs. (44) and (45). Spiral motion is modeled as stated in Eq. (46). The circle narrowing behavior occurs by decreasing the value of a in Eq. (44).

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \quad (44)$$

$$\vec{C} = 2 \cdot \vec{r} \quad (45)$$

where $\vec{a}$ is linearly decreased from 2 to 0 over the iterations and $\vec{r}$ is a random vector which take values within the intervals [0,1]. Eq. (46) is formed by calculating the distance between the target position and the solution candidate for the spiral motion behavior.

$$\vec{X}(t+1) = \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) \quad (46)$$

$$\vec{D}' = \vec{X}^*(t) - \vec{X}(t)$$

where b represents logarithmic spiral constant, l represents a random number within intervals [-1,1]. WOA determines with 50% probability whether the movement is spiral or linear. This selection is modeled in Eq. (47):

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t) - \vec{A} \cdot \vec{D}, & p < 0,5 \\ \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t), & p \geq 0,5 \end{cases} \quad (47)$$

where p is a random number between 0 and 1.

3.3.6.3. Search for prey

By looking at other whales in their herd, humpback whales may be on a random search. For the global solution, the best-known point is sometimes replaced by a randomly determined candidate when determining the new positions of the solution candidates. Other solution candidates form around this candidate. As a result, values greater than 1 and less than -1 are used for vector A . Solution candidates diverge from reference candidates in such cases. Model of this behavior is as follows:

$$\vec{D}' = \vec{C} \cdot \overrightarrow{X_{rand}} - \vec{X} \quad (48)$$

$$\vec{X}(t+1) = \overrightarrow{X_{rand}} - \vec{A} \cdot \vec{D} \quad (49)$$

where $\overrightarrow{X_{rand}}$ is a random solution vector. Eq. (49) uses vector A as a determinant of the scale at which the search should be performed. Whenever A is less than -1 or greater than 1, it is considered a global search and the equation mentioned above are applied. In other cases, a local call is made.

3.3.7. Ant Colony Optimization

Ant Colony Optimization (ACO) is a metaheuristic optimization algorithm that is based upon the behavior of ants in nature. ACO has become popular in the field of combinatorial optimization since it was introduced by Marco Dorigo in the early 1990s (Dorigo, 1992). Based on ants' behavior, ACO seeks the shortest path between a colony and a food source (Aghdam et al., 2009). An ant communicates with another ant by releasing a pheromone trail that is followed by other ants to the food source, and this trail is used for communication between ants. As more ants follow the trail, the concentration of pheromones on it increases, making it more attractive to other ants as more of them follow it. The shortest path from the colony to the food source becomes more narrow and defined as the concentrations of pheromones increase along the shortest path (Tabakhi et al., 2014). Based on the behavior of ACO, an optimization algorithm can be derived from it. As a method of solving the optimization problem, virtual ants are created, each representing a potential solution to the problem. An ant constructs a solution by selecting components in the solution space based on probabilistic rules that are influenced by pheromone trails in their environment. A stronger pheromone trail is created when the quality of the solutions constructed by the ants is improved, and a better solution leads to a stronger pheromone trail as a result. One of the key advantages of ACO is that it is capable of handling combinatorial optimization problems with a large solution space, which makes it a powerful optimizer. Even when the problem is highly complex, ACO can efficiently explore the solution space and arrive at a satisfactory solution (Ghosh et al., 2020).

At the beginning of the ACO process, ants construct their solutions based on the action choice rule:

$$s = \begin{cases} \arg \max\{\tau(u) \cdot [\varphi(u)]^\beta\} & \text{if } q \leq q_0 \\ s & \text{otherwise} \end{cases} \quad (50)$$

In this model, $\tau(u)$ represents the pheromone level of feature u , which indicates the level of information provided by the feature; $\varphi(u)$ represents the heuristic position of the feature u as a desirable feature. For an ant t , $F_t(r)$ represents the set of features that have not yet been selected, that is, they are not included in the current solution set. The significance of the heuristic values is determined by a parameter called β . q is a uniformly distributed value in

$[0,1]$ and q_0 is an exploitation probability factor parameter. The feature S was selected by maximising the following probability function using the pheromones and the heuristics values.

$$P_t(s) = \begin{cases} \frac{[\tau(s)] \cdot [\varphi(s)]^\beta}{\sum_{u \in F_t(r)} [\tau(u)] \cdot [\varphi(u)]^\beta} & \text{if } s \in F_t(r) \\ 0 & \text{otherwise} \end{cases} \quad (51)$$

If $q < q_0$, the ants will select the features that maximize both the pheromone level and the heuristic value. In this case, the information learned from the past experience will be exploited. A second step involves iterating both the global and local pheromone trails. As a result of global updating, features that belong to a subset with the lowest classification error are rewarded. Following is the global updating rule that will modify the pheromone level $\tau(u)$ of only these features:

$$\tau(u) \leftarrow (1 - \alpha) \cdot \tau(u) + \alpha \cdot \Delta\tau \quad (52)$$

where $\alpha \in (0,1]$ is a pheromone trail decay coefficient and $\Delta\tau$ is the inverse of the least classification error of the global best ants. By using the local updating rule, it is ensured that ants are not allowed to select features with higher pheromone levels, so that they may investigate features that have not yet been selected by all ants, allowing them to explore features that have not yet been explored by all of them. In each iteration, the pheromone level $\tau(r)$ of those features which are not selected in the best subset are updated by the rule:

$$\tau(r) \leftarrow (1 - \alpha) \cdot \tau(r) + \alpha \cdot \tau_0 \quad (53)$$

where τ_0 is the initial pheromone level.

3.3.8. Genetic Algorithm

GA is an optimization algorithm that is based on the principles of natural selection in biology. It is a metaheuristic algorithm available to solve a variety of optimization problems that can be classified into a wide range of categories. In general, GA mimics the process of natural selection by generating a population of candidate solutions iteratively and selecting the healthiest individuals from that population to create new offspring. Based on a set of criteria that assesses whether the solution to a problem is of a high quality or not, fitness functions are used to find the fittest individuals (Babatunde et al., 2014). Genetic operators like crossover and mutation

are used to create offspring from the genetic information of the selected individuals as well as the genetic information from the crossed individuals (Maleki et al., 2021). In this process, iterations are repeated until a certain number of iterations has been completed or a specific termination criterion has been met, such as convergence or a maximum number of iterations has been reached.

GA offers the advantage of handling complex optimization problems with a large solution space. GA is capable of exploring the solution space efficiently and converge on a good solution even when the problem is highly nonlinear and multimodal. The main advantage of GA is that it is relatively easy to implement and can be used for the solution of a wide range of optimization problems. The application of GA has been successful in feature selection, parameter optimizations, neural network architectures searching and many other fields in machine learning studies.

3.3.9. Partical Swarm Optimization

PSO is introduced by Eberhart and Kennedy (1995), by simulating the evolution of knowledge of a social behavior and simulating the behavior of group communication, it mimics the behavior of sharing private information regarding migrations, flockings, or hunting with others. It is said that the swarm and its members represent the solution and are called a swarm and particle (Niknam and Amiri, 2010). Particles constantly update their positions based on their experience and the presence of their neighbors. Particles change their search direction based on two factors: their own best previous experience and the best experiences of the other particles (Lin et al., 2008).

Initially, the swarm establishes a set of random particles along with their positions (x_i) and velocities (v_i) in a dimension j^{th} and generates a set of random particles in order to perform the swarm. As soon as PSO starts its main loop, it will evaluate each particle based on its fitness function. The results will be compared with its global best value as well as its best value for each particle (Ali Ibrahim et al., 2019). Update of particles can be formulated as follows:

$$x_{ij}^{(t+1)} = x_{ij}^{(t)} + v_{ij}^{(t+1)} \quad (54)$$

$$v_{ij}^{(t+1)} = wv_{ij}^{(t)} + c_1r_1(x_{ij}^{p(t)} - x_{ij}^{(t)}) + c_2r_2(x_j^{g(t)} - x_{ij}^{(t)}) \quad (55)$$

where x_{ij} is the i^{th} particle position in the j^{th} dimension, v_{ij} is the i^{th} velocity in the j^{th} dimension, t represents the current iteration, w indicates an inertia weight and applied to improve the speed of the population convergence. c_1 and c_2 represents acceleration coefficients. $x_{ij}^{p(t)}$ defines the best previous position of particle i in j^{th} ; $x_j^{g(t)}$ defines the global best position in j^{th} dimension. r_1 and r_2 are random numbers within the interval $[0,1]$.

3.4. Evaluation Metrics

A classification problem must be analyzed using various evaluation criteria in order to determine whether it is successful. Although accuracy is commonly used to determine the success of a model, it is not considered adequate on its own. Similarly, precision measures how many of the values predicted as positive are actually positive, whereas recall measures how many of the transactions users must correctly predict as positive in order to produce a positive outcome. The F-Score plays a significant role in balancing precision and recall. As a measure of precision and recall, the F-Score is the harmonic mean. The formulas of these evaluation criteria are shown in Table 3.

Table 3.2 The evaluation criteria's formulas

Evaluation Criteria	Formula
Accuracy	$(\text{True Positive} + \text{True Negative}) / (\text{Positive} + \text{Negative})$
Precision	$\text{True Positive} / (\text{True Positive} + \text{False Positive})$
Recall	$\text{True Positive} / \text{Positive}$
F-Score	$2 * \text{Recall} * \text{Precision} / (\text{Precision} + \text{Recall})$

4. RESULTS AND DISCUSSION

4.1. Parameter Setting

Parameters of the algorithms given in Table 4.1.

Table 4.1. Parameters of the algorithms

Optimizer Name	Name of the Parameter	Value of the Parameter
Sailfish Optimizer	population	50
	iteration	50
	A	4
	epxilon	0.001
Marine Predators Algorithm	p	0.5
	FADs	0.2
Ant Colony Optimizer	population	50
	iteration	50
	initial pheromone	1
	α	1
	β	0.1
	ρ	0.2
Whale Optimizer	population	50
	iteration	50
Emperor Penguins Colony Optimizer	population	50
	iteration	50
Salp Swarm Optimizer	population	50
	iteration	50
Genetic Algorithm	population	50
	iteration	50
	crossover rate	0.8
	mutation rate	0.01
Partical Swarm Optimization	population	50
	iteration	50
	cognitive factor	2
	social factor	2
	inertia weight	0.9
Jellyfish Optimizer	population	50
	iteration	50
KNN	k	5
Random Forest	number of trees	100

As it is seen from Table 4.1, population and iteration values for all of the algorithms are 50 and 50.

4.2. Experiment Results

Experiments are run on 22 datasets. Feature selection process is applied by 9 different metaheuristic algorithms and one traditional method. For each evaluation, 4 different classifiers are used. Results are evaluated by 5 different evaluation metrics. In this section of the thesis, results are presented based on the evaluation metrics.

4.2.1. Based on Accuracy

Table 4.1. Accuracy results using naïve bayes

Classifier		Naive Bayes									
Dataset Name	Accuracy without FS	Accuracy									
		SEA CREATURES						POPULAR OPTIMIZERS			TRADITIONAL
		EPO	MPA	SSA	JS	SFO	WOA	ACO	GA	PSO	CHI2
breast	0,900	0,929	0,941	0,929	0,894	0,831	0,924	0,918	0,941	0,924	0,894
breastcancer	0,962	0,986	0,976	0,962	0,981	0,949	0,976	0,976	0,971	0,981	0,971
car	0,678	0,714	0,691	0,687	0,726	0,932	0,708	0,687	0,703	0,687	0,701
clean1	0,704	0,852	0,915	0,845	0,775	0,881	0,852	0,845	0,901	0,894	0,711
clean2	0,843	0,897	0,914	0,845	0,852	0,881	0,886	0,874	0,888	0,881	0,678
colon	0,833	0,944	1,000	0,778	0,667	0,881	0,944	0,944	0,667	0,944	0,778
credit-g	0,750	0,767	0,787	0,750	0,743	0,797	0,763	0,757	0,817	0,777	0,693
dermatology	0,945	0,991	0,991	0,991	0,982	0,881	1,000	0,991	0,991	0,991	0,817
ecoli	0,510	0,640	0,650	0,590	0,600	0,780	0,670	0,680	0,670	0,690	0,630
glass	0,406	0,391	0,406	0,516	0,484	0,763	0,375	0,406	0,500	0,531	0,250
heart-c	0,822	0,867	0,900	0,911	0,922	0,831	0,889	0,856	0,867	0,856	0,833
ionosphere	0,724	0,886	0,867	0,771	0,762	0,864	0,886	0,838	0,867	0,867	0,829
kr-vs-kp	0,847	0,943	0,946	0,932	0,953	0,847	0,956	0,939	0,953	0,956	0,904
leukemia	1,000	1,000	1,000	1,000	0,952	0,898	0,952	1,000	1,000	1,000	0,905
lymphography	0,818	0,818	0,909	0,818	0,795	0,898	0,841	0,864	0,886	0,818	0,636
sonar	0,710	0,839	0,935	0,823	0,661	0,881	0,887	0,855	0,952	0,887	0,774
spect	0,712	0,775	0,788	0,775	0,862	0,864	0,825	0,825	0,763	0,813	-
tic-tac-toe	0,683	0,718	0,749	0,780	0,749	0,898	0,711	0,732	0,728	0,739	0,707
waveform	0,793	0,827	0,827	0,817	0,815	0,814	0,827	0,835	0,843	0,839	-
wine	0,962	1,000	0,981	1,000	0,981	0,915	1,000	1,000	1,000	1,000	0,943
yeast	0,510	0,490	0,474	0,485	0,461	0,915	0,470	0,479	0,452	0,488	0,418
zoo	0,767	0,900	0,933	0,933	0,833	0,881	0,933	0,900	0,933	0,833	-

We can conclude from Table 4.1 that not any particular algorithm outperforms all other algorithms. GA outperforms other algorithms on 8 out of 22 datasets, having the most highest scores, which shows that there is a real competition between algorithms and none of the algorithms stand out among others. The table shows that MPA is the second best and SFO is the third best algorithm among other algorithms. JS outperforms other algorithms on 4 out of 22 datasets. JS enhances the accuracy score of 15 datasets after the feature selection process. The results on Table 4.1 also shows that traditional method CHI2 does not have any highest score which makes it the worst method among others.

Table 4.2. Accuracy results using support vector machine

Classifier		Support Vector Machine									
Dataset Name	Accuracy without FS	Accuracy									
		SEA CREATURES						POPULAR OPTIMIZERS			TRADITIONAL
		EPO	MPA	SSA	JS	SFO	WOA	ACO	GA	PSO	CHI2
breast	0,476	0,647	0,647	0,612	0,929	0,831	0,865	0,629	0,747	0,612	0,565
breastcancer	0,962	0,971	0,957	0,986	0,971	0,847	0,971	0,967	0,971	0,962	0,957
car	0,892	0,842	0,855	0,861	0,838	0,797	0,875	0,863	0,838	0,869	0,766
clean1	0,880	0,951	0,972	0,958	0,717	0,780	0,880	0,887	0,972	0,965	0,725
clean2	0,644	0,930	0,923	0,847	0,867	0,797	0,840	0,928	0,862	0,845	0,767
colon	0,889	0,944	1,000	0,500	0,905	0,780	1,000	0,833	0,667	0,722	0,833
credit-g	0,717	0,753	0,783	0,730	0,942	0,814	0,720	0,777	0,813	0,770	0,723
dermatology	0,954	0,991	0,982	0,982	0,774	0,814	0,982	0,982	1,000	0,972	0,853
ecoli	0,790	0,810	0,810	0,790	0,850	0,780	0,870	0,870	0,730	0,760	0,740
glass	0,641	0,609	0,719	0,625	0,983	0,780	0,719	0,672	0,703	0,609	0,578
heart-c	0,789	0,833	0,867	0,833	0,774	0,814	0,878	0,822	0,844	0,833	0,767
ionosphere	0,914	0,990	0,981	0,952	0,850	0,814	0,924	0,981	0,981	0,990	0,886
kr-vs-kp	0,953	0,967	0,993	0,965	0,983	0,763	0,958	0,991	0,995	0,989	0,900
leukemia	0,857	0,857	0,524	0,762	0,842	0,797	1,000	1,000	0,571	0,810	0,571
lymphography	0,750	1,000	1,000	1,000	0,972	0,763	0,909	0,619	1,000	1,000	0,591
sonar	0,774	0,887	0,952	0,903	0,850	0,831	0,855	0,903	0,952	0,968	0,726
spect	0,800	0,900	0,888	0,925	0,594	0,797	0,888	0,813	0,825	0,900	-
tic-tac-toe	0,979	0,878	0,986	0,990	0,909	0,797	0,983	0,979	0,965	0,965	0,610
waveform	0,863	0,849	0,888	0,870	0,857	0,864	0,875	0,863	0,881	0,891	-
wine	1,000	1,000	1,000	0,981	0,981	0,814	1,000	1,000	0,981	1,000	0,981
yeast	0,485	0,535	0,551	0,521	0,578	0,814	0,526	0,510	0,512	0,528	0,481
zoo	1,000	1,000	1,000	0,967	0,900	0,780	0,967	1,000	1,000	1,000	0,867

We can conclude from Table 4.2 that not any particular algorithm outperforms all other algorithms. MPA, WOA and PSO outperform other algorithms on 6 out of 22 datasets, having the most highest scores. The table shows that EPO, ACO and GA are the second best algorithms amongs other algorithms. JS outperforms other algorithms on 3 out of 22 datasets which is quite low compared to performances of other algorithms. JS enhances the accuracy score of 10 datasets after the feature selection process. It might be considered low but given the fact that 12 datasets already have higher than 85% accuracy score without any feature selection process makes it comprehensible. The results on Table 4.2 also shows that traditional method CHI2 does not have any highest score which makes it the worst method amongs others.

Table 4.3. Accuracy results using k-nearest neighbor

Classifier		K-Nearest Neighbor									
Dataset Name	Accuracy without FS	Accuracy									
		SEA CREATURES						POPULAR OPTIMIZERS			TRADITIONAL
		EPO	MPA	SSA	JS	SFO	WOA	ACO	GA	PSO	CHI2
breast	0,924	0,941	0,982	0,953	0,947	0,947	0,959	0,959	0,953	0,959	0,941
breastcancer	0,986	0,990	0,981	0,981	0,981	0,978	0,976	0,981	0,981	0,957	0,957
car	0,900	0,942	0,917	0,950	0,950	0,954	0,927	0,925	0,923	0,944	0,689
clean1	0,824	0,894	0,915	0,852	0,944	1,000	0,908	0,873	0,986	0,887	0,761
clean2	0,965	0,958	0,976	0,966	0,966	1,000	0,969	0,969	0,982	0,977	0,934
colon	0,667	0,944	0,944	0,722	0,556	0,615	0,944	0,944	1,000	0,667	0,778
credit-g	0,643	0,740	0,763	0,763	0,703	0,745	0,763	0,737	0,783	0,760	0,667
dermatology	0,835	0,982	0,991	0,982	0,972	0,903	0,991	0,991	0,991	1,000	0,817
ecoli	0,860	0,870	0,900	0,900	0,890	0,985	0,890	0,890	0,900	0,900	0,780
glass	0,688	0,688	0,672	0,734	0,719	0,744	0,641	0,734	0,641	0,672	0,734
heart-c	0,633	0,822	0,878	0,856	0,789	0,814	0,867	0,867	0,822	0,844	0,689
ionosphere	0,848	0,943	0,981	0,914	0,867	0,914	0,943	0,962	0,952	0,952	0,876
kr-vs-kp	0,927	0,929	0,969	0,932	0,921	0,998	0,949	0,947	0,981	0,978	0,763
leukemia	0,905	1,000	1,000	1,000	0,905	1,000	1,000	1,000	1,000	1,000	1,000
lymphography	0,727	0,909	0,909	0,841	0,932	0,900	0,886	0,864	0,909	0,932	0,795
sonar	0,710	0,919	0,887	0,871	0,839	1,000	0,887	0,919	0,952	0,919	0,742
spect	0,713	0,813	0,863	0,875	0,850	1,000	0,863	0,875	0,875	0,875	-
tic-tac-toe	0,997	0,937	0,997	0,885	0,993	0,995	0,993	0,993	0,990	0,997	0,655
waveform	0,806	0,808	0,851	0,816	0,811	0,826	0,830	0,833	0,844	0,854	-
wine	0,755	0,943	0,981	0,981	0,981	1,000	0,962	1,000	0,981	0,981	0,642
yeast	0,560	0,553	0,591	0,571	0,593	0,562	0,602	0,591	0,569	0,566	0,476
zoo	0,867	0,967	1,000	1,000	0,967	0,950	0,900	1,000	1,000	0,967	0,833

We can conclude from Table 4.3 that SFO outperforms other algorithms on half of the datasets with quite high accuracy scores. The table shows that MPA is the second best and PSO is the third best algorithm amongs other algorithms. JS outperforms other algorithms on 3 out of 22 datasets. JS enhances the accuracy score of 17 datasets after the feature selection process even they are not the best amongs other algorithms. The results on Table 4.3 also shows that traditional method CHI2 does not have any highest score which makes it the worst method amongs others.

Table 4.4. Accuracy results using random forest

Classifier		Random Forest									
Dataset Name	Accuracy without FS	Accuracy									
		SEA CREATURES						POPULAR OPTIMIZERS			TRADITIONAL
		EPO	MPA	SSA	JS	SFO	WOA	ACO	GA	PSO	CHI2
breast	0,929	0,929	0,965	0,941	0,935	0,847	0,918	0,941	0,971	0,971	0,924
breastcancer	0,919	0,981	0,962	0,957	0,976	0,763	0,967	0,967	0,986	0,976	0,938
car	0,944	0,973	0,936	0,950	0,967	0,746	0,954	0,948	0,971	0,936	0,768
clean1	0,803	0,880	0,923	0,880	0,866	0,797	0,887	0,845	0,937	0,915	0,754
clean2	0,955	0,975	0,983	0,976	0,971	0,847	0,982	0,967	0,981	0,979	0,942
colon	0,833	0,944	1,000	0,944	0,889	0,797	0,889	0,944	0,944	0,944	0,778
credit-g	0,740	0,783	0,777	0,737	0,717	0,831	0,760	0,783	0,783	0,750	0,643
dermatology	0,936	0,972	0,991	0,972	0,991	0,847	0,972	0,991	0,991	0,982	0,862
ecoli	0,740	0,830	0,820	0,870	0,872	0,814	0,850	0,810	0,860	0,820	0,820
glass	0,531	0,734	0,813	0,766	0,797	0,780	0,766	0,719	0,781	0,781	0,703
heart-c	0,778	0,789	0,878	0,833	0,822	0,763	0,856	0,878	0,900	0,856	0,767
ionosphere	0,867	0,962	0,962	0,952	0,933	0,847	0,943	0,943	0,971	0,952	0,848
kr-vs-kp	0,993	0,983	0,998	0,982	0,996	0,763	0,996	0,992	0,994	0,995	0,903
leukemia	0,905	1,000	1,000	0,952	0,952	0,780	1,000	0,905	0,952	0,952	0,857
lymphography	0,750	0,864	0,864	0,886	0,864	0,797	0,841	0,886	0,818	0,864	0,727
sonar	0,742	0,887	0,839	0,839	0,758	0,831	0,871	0,806	0,855	0,919	0,629
spect	0,775	0,900	0,900	0,900	0,900	0,881	0,863	0,875	0,913	0,888	-
tic-tac-toe	0,934	0,948	0,909	0,927	0,913	0,780	0,934	0,920	0,955	0,913	0,693
waveform	0,756	0,755	0,777	0,755	0,751	0,797	0,772	0,770	0,783	0,777	-
wine	0,981	0,943	1,000	0,981	0,925	0,729	0,981	0,981	0,962	0,981	0,943
yeast	0,515	0,560	0,535	0,560	0,553	0,847	0,535	0,539	0,548	0,546	0,429
zoo	0,800	0,933	0,967	0,967	0,933	0,797	0,933	1,000	0,967	1,000	0,800

We can conclude from Table 4.4 that GA outperforms other algorithms on 10 out of 22 datasets. The table shows that MPA is the second best algorithm and EPO, JS, SFO, WOA, ACO and PSO are tied being the third algorithms. JS outperforms other algorithms on 3 out of 22 datasets. JS enhances the accuracy score of 18 datasets after the feature selection process. The results on Table 4.4 also shows that traditional method CHI2 does not have any highest score which makes it the worst method amongs others.

4.2.2. Based on Precision

Table 4.5. Precision results using naïve bayes

Classifier		Naive Bayes									
Dataset Name	Precision without FS	Precision									
		SEA CREATURES						POPULAR OPTIMIZERS			TRADITIONAL
		EPO	MPA	SSA	JS	SFO	WOA	ACO	GA	PSO	CHI2
breast	0,878	0,951	0,928	0,932	0,953	0,893	0,957	0,944	0,945	0,963	0,911
breastcancer	0,957	1,000	0,986	0,986	0,987	0,968	0,986	0,986	0,939	0,971	1,000
car	1,000	1,000	1,000	1,000	1,000	0,938	1,000	1,000	1,000	1,000	1,000
clean1	0,645	0,838	0,907	0,769	0,744	0,903	0,776	0,833	0,884	0,881	0,566
clean2	0,857	0,975	0,989	0,866	0,889	0,879	0,966	0,912	0,939	0,915	0,649
colon	0,833	1,000	1,000	0,769	0,778	0,931	0,900	1,000	0,455	1,000	0,786
credit-g	0,796	0,921	0,891	0,851	0,854	0,750	0,883	0,833	0,897	0,872	0,780
dermatology	1,000	1,000	1,000	1,000	1,000	0,857	1,000	1,000	1,000	1,000	1,000
ecoli	0,971	1,000	1,000	0,974	1,000	0,732	0,973	0,974	0,974	0,979	1,000
glass	0,045	0,889	0,286	0,400	0,286	0,821	0,944	0,375	0,368	0,455	0,824
heart-c	0,848	0,787	0,867	0,980	0,907	0,789	1,000	0,900	0,952	0,778	0,792
ionosphere	0,310	0,744	0,667	0,385	0,306	0,900	0,684	0,553	0,600	0,667	0,379
kr-vs-kp	0,844	0,920	0,923	0,925	0,916	0,795	0,928	0,915	0,947	0,951	0,822
leukemia	1,000	1,000	1,000	1,000	1,000	0,906	1,000	1,000	1,000	1,000	0,929
lymphography	0,717	0,874	0,848	0,950	-	0,964	0,879	1,000	0,836	1,000	-
sonar	0,774	0,742	0,966	0,880	0,920	0,857	0,903	0,871	1,000	0,903	0,828
spect	0,724	0,803	0,056	0,053	0,077	0,875	0,798	0,873	0,837	0,868	-
tic-tac-toe	0,429	0,440	0,485	0,478	0,458	0,882	0,434	0,427	0,472	0,505	0,592
waveform	0,699	0,843	0,824	0,852	0,810	0,839	0,821	0,813	0,815	0,819	-
wine	0,867	1,000	0,933	1,000	0,944	0,909	1,000	1,000	1,000	1,000	1,000
yeast	0,750	0,793	0,636	0,746	0,774	0,935	0,701	0,792	0,693	0,713	0,638
zoo	1,000	1,000	1,000	1,000	1,000	0,931	1,000	1,000	1,000	1,000	-

We can conclude from Table 4.5 that EPO and PSO outperform other algorithms on 9 out of 22 datasets. The table shows that MPA and ACO are the second best algorithms and WOA is the third best algorithm. JS outperforms other algorithms on 5 out of 22 datasets. JS enhances the precision score of 18 datasets after the feature selection process. The results on Table 4.5 also shows that traditional method CHI2 have 5 highest scores among others.

Table 4.6. Precision results using support vector machine

Classifier		Support Vector Machine									
Dataset Name	Precision without FS	Precision									
		SEA CREATURES						POPULAR OPTIMIZERS			TRADITIONAL
		EPO	MPA	SSA	JS	SFO	WOA	ACO	GA	PSO	CHI2
breast	0,540	0,951	0,990	1,000	0,907	0,844	0,901	0,990	1,000	1,000	0,685
breastcancer	0,922	0,985	0,957	1,000	0,951	0,829	0,954	0,942	1,000	0,956	0,932
car	0,972	0,938	0,955	0,960	0,925	0,813	0,953	0,963	0,941	0,964	0,866
clean1	0,885	0,976	1,000	0,969	1,000	0,771	0,854	0,939	0,988	1,000	0,854
clean2	0,702	0,989	0,996	1,000	0,837	0,763	1,000	0,990	1,000	1,000	0,854
colon	0,857	1,000	1,000	1,000	0,794	0,771	1,000	1,000	1,000	1,000	0,867
credit-g	0,815	0,886	0,930	0,915	0,968	0,839	0,919	0,935	0,938	0,943	1,000
dermatology	1,000	1,000	1,000	1,000	0,786	0,784	1,000	1,000	1,000	1,000	1,000
ecoli	1,000	1,000	1,000	1,000	0,529	0,852	1,000	1,000	1,000	1,000	1,000
glass	0,650	0,619	0,800	0,636	0,950	0,757	0,947	0,650	0,895	0,722	0,591
heart-c	0,816	0,894	0,891	0,926	0,786	0,889	0,889	0,827	0,865	0,896	0,843
ionosphere	0,853	0,969	0,972	0,931	0,529	0,862	0,810	0,977	0,976	0,963	0,763
kr-vs-kp	0,935	0,981	0,993	0,967	0,950	0,765	0,943	0,998	0,993	0,985	0,823
leukemia	0,750	1,000	1,000	1,000	1,000	0,833	1,000	1,000	1,000	1,000	1,000
lymphography	0,909	1,000	1,000	1,000	1,000	0,846	0,960	1,000	1,000	1,000	-
sonar	0,724	0,769	0,897	0,813	1,000	0,775	0,880	0,800	0,913	0,955	0,667
spect	0,563	0,800	0,667	0,733	0,500	0,833	0,632	0,609	0,421	0,684	-
tic-tac-toe	0,941	0,782	0,962	0,967	1,000	0,833	0,950	0,948	0,906	0,906	0,700
waveform	0,896	0,922	0,933	0,944	0,918	0,875	0,915	0,901	0,937	0,948	-
wine	1,000	1,000	1,000	1,000	1,000	0,769	1,000	1,000	1,000	1,000	0,933
yeast	0,875	0,924	0,870	0,879	0,857	0,862	0,794	0,877	0,865	0,864	0,833
zoo	1,000	1,000	1,000	1,000	1,000	0,806	1,000	1,000	1,000	1,000	1,000

We can conclude from Table 4.6 that SSA, GA and PSO outperform other algorithms on 12 out of 22 datasets. The table shows that MPA is the second best algorithms and ACO is the third best algorithm. JS outperforms other algorithms on 8 out of 22 datasets based on precision. JS enhances the precision score of 14 datasets after the feature selection process. The results on Table 4.6 also shows that traditional method CHI2 have 4 highest scores among others.

Table 4.7. Precision results using k-nearest neighbor

Classifier		K-Nearest Neighbor									
Dataset Name	Precision without FS	Precision									
		SEA CREATURES						POPULAR OPTIMIZERS			TRADITIONAL
		EPO	MPA	SSA	JS	SFO	WOA	ACO	GA	PSO	CHI2
breast	0,981	0,953	1,000	0,981	0,963	0,947	0,982	0,981	0,990	0,990	0,939
breastcancer	0,987	0,986	0,986	0,986	0,986	0,978	0,966	0,985	0,972	0,962	0,968
car	0,992	0,977	0,980	0,971	1,000	0,954	0,968	0,969	0,960	0,988	1,000
clean1	0,718	0,882	0,938	0,753	0,917	1,000	0,881	0,822	0,973	0,815	0,774
clean2	0,992	0,987	0,994	0,986	0,993	1,000	0,991	0,989	0,995	0,995	0,968
colon	0,750	1,000	1,000	0,900	0,833	0,615	1,000	1,000	1,000	0,875	0,889
credit-g	0,823	0,923	0,909	0,907	0,920	0,745	0,862	0,901	0,900	0,850	0,620
dermatology	0,939	1,000	1,000	1,000	1,000	0,903	0,972	1,000	1,000	1,000	1,000
ecoli	1,000	1,000	1,000	1,000	1,000	0,985	1,000	1,000	1,000	1,000	1,000
glass	0,875	0,870	0,643	0,923	0,917	0,744	0,900	0,941	0,944	0,880	0,913
heart-c	0,750	0,872	0,898	0,881	0,765	0,814	0,941	0,955	0,776	0,864	0,905
ionosphere	0,629	0,871	0,943	0,854	0,649	0,914	0,850	0,917	0,865	0,872	0,581
kr-vs-kp	0,859	0,863	0,948	0,880	0,847	0,998	0,906	0,895	0,966	0,961	0,501
leukemia	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
lymphography	0,765	1,000	0,952	0,923	0,929	0,900	0,914	0,918	0,926	0,969	-
sonar	0,667	0,889	0,813	0,852	0,771	1,000	0,846	0,815	0,906	0,931	0,600
spect	0,500	0,450	0,696	0,600	0,400	1,000	0,647	0,647	0,167	0,667	-
tic-tac-toe	0,990	0,838	0,991	0,687	0,979	0,995	0,981	0,981	0,973	0,991	0,000
waveform	0,901	0,863	0,898	0,864	0,897	0,826	0,887	0,902	0,900	0,912	-
wine	1,000	0,941	1,000	0,947	1,000	1,000	1,000	1,000	1,000	1,000	0,917
yeast	0,740	0,638	0,705	0,748	0,742	0,562	0,787	0,707	0,699	0,735	0,735
zoo	1,000	1,000	1,000	1,000	1,000	0,950	1,000	1,000	1,000	1,000	1,000

We can conclude from Table 4.7 that MPA outperforms other algorithms on 10 out of 22 datasets. The table shows that ACO is the second best algorithm and SFO is the third best algorithm. JS outperforms other algorithms on 7 out of 22 datasets based on precision. JS enhances the precision score of 16 datasets after the feature selection process. The results on Table 4.7 also shows that traditional method CHI2 have 4 highest scores among others.

Table 4.8. Precision results using random forest

Classifier		Random Forest									
Dataset Name	Precision without FS	Precision									
		SEA CREATURES						POPULAR OPTIMIZERS			TRADITIONAL
		EPO	MPA	SSA	JS	SFO	WOA	ACO	GA	PSO	CHI2
breast	0,935	0,919	0,979	0,980	0,951	0,780	0,952	0,971	0,972	0,990	0,909
breastcancer	0,872	1,000	0,988	0,938	0,944	0,765	1,000	0,987	0,976	0,973	0,925
car	0,972	0,975	0,949	0,948	1,000	0,758	0,975	0,975	0,981	0,954	0,786
clean1	0,821	0,865	0,939	0,920	0,842	0,763	0,962	0,893	0,965	0,940	0,728
clean2	0,976	0,986	0,993	0,990	1,000	0,897	0,992	0,986	0,990	0,993	0,966
colon	0,933	1,000	1,000	1,000	1,000	0,794	1,000	1,000	1,000	1,000	0,889
credit-g	0,804	0,906	0,870	0,842	1,000	0,867	0,877	0,854	0,912	0,885	0,765
dermatology	0,970	1,000	1,000	1,000	0,896	0,871	1,000	1,000	1,000	1,000	1,000
ecoli	1,000	0,978	0,976	1,000	0,865	0,769	1,000	1,000	0,978	1,000	1,000
glass	0,588	0,778	0,960	0,769	0,975	0,757	0,938	0,850	0,864	0,850	0,944
heart-c	0,788	0,813	0,904	0,840	1,000	0,765	0,829	0,940	0,878	0,887	0,704
ionosphere	0,717	0,886	0,892	0,912	1,000	0,829	0,943	0,925	0,971	0,921	0,763
kr-vs-kp	1,000	0,998	1,000	0,989	0,839	0,714	0,996	0,993	0,993	0,998	0,838
leukemia	0,938	1,000	1,000	0,917	0,818	0,744	1,000	0,867	1,000	0,929	0,846
lymphography	0,679	0,955	0,930	0,908	0,870	0,794	0,893	0,889	0,950	0,934	-
sonar	0,792	0,857	0,786	0,828	0,853	0,775	0,880	0,793	0,829	0,903	0,552
spect	0,636	0,615	0,720	0,687	0,933	0,903	0,500	0,647	0,667	0,579	-
tic-tac-toe	0,900	0,913	0,753	0,875	1,000	0,788	0,848	0,844	0,914	0,884	0,563
waveform	0,869	0,853	0,864	0,865	1,000	0,750	0,886	0,845	0,852	0,887	-
wine	1,000	0,824	1,000	1,000	0,932	0,700	0,947	1,000	1,000	1,000	1,000
yeast	0,706	0,659	0,658	0,654	0,988	0,871	0,627	0,571	0,667	0,695	0,534
zoo	1,000	1,000	1,000	1,000	0,857	0,833	1,000	1,000	1,000	1,000	1,000

We can conclude from Table 4.8 that JS outperforms other algorithms on 11 out of 22 datasets. The table shows that WOA and GA are the second best algorithms and EPO, MPA and PSO are the third best algorithms. JS outperforms other algorithms on half of the datasets based on precision score. JS enhances the precision score of 15 datasets after the feature selection process. The results on Table 4.8 also shows that traditional method CHI2 have 4 highest scores among others.

4.2.3. Based on Recall

Table 4.9. Recall results using naïve bayes

Classifier		Naive Bayes									
Dataset Name	Recall without FS	Recall									
		SEA CREATURES						POPULAR OPTIMIZERS			TRADITIONAL
		EPO	MPA	SSA	JS	SFO	WOA	ACO	GA	PSO	
breast	0,971	0,933	0,981	0,950	0,936	0,781	0,932	0,927	0,963	0,921	0,911
breastcancer	0,931	0,952	0,947	0,907	0,973	0,938	0,948	0,947	0,969	0,971	0,920
car	0,745	0,776	0,749	0,753	0,790	0,938	0,779	0,739	0,755	0,751	0,763
clean1	0,766	0,893	0,932	0,938	0,902	0,875	0,937	0,897	0,950	0,894	0,843
clean2	0,954	0,910	0,917	0,949	0,950	0,906	0,905	0,936	0,930	0,942	0,960
colon	0,909	0,933	1,000	0,909	0,667	0,844	1,000	0,923	1,000	0,923	0,917
credit-g	0,832	0,789	0,811	0,801	0,823	0,938	0,794	0,813	0,853	0,830	0,773
dermatology	0,971	1,000	1,000	1,000	1,000	0,938	1,000	1,000	1,000	1,000	1,000
ecoli	0,767	0,822	0,868	0,864	0,955	0,938	0,878	0,905	0,787	0,807	0,827
glass	1,000	0,471	0,800	1,000	1,000	0,719	0,447	0,563	0,700	1,000	0,368
heart-c	0,813	0,949	0,929	0,875	0,925	0,938	0,792	0,849	0,800	0,921	0,884
ionosphere	1,000	0,935	0,923	1,000	1,000	0,844	1,000	1,000	1,000	0,880	1,000
kr-vs-kp	0,842	0,956	0,964	0,927	0,924	0,969	0,982	0,954	0,957	0,959	0,972
leukemia	1,000	1,000	1,000	1,000	0,909	0,906	0,923	1,000	1,000	1,000	0,929
lymphography	0,913	0,872	0,900	0,760	-	0,844	0,906	0,769	0,800	0,714	-
sonar	0,686	0,920	0,903	0,733	0,537	0,938	0,875	0,844	0,903	0,875	0,727
spect	0,778	0,869	1,000	1,000	1,000	0,875	0,869	0,870	0,885	0,800	-
tic-tac-toe	0,592	0,761	0,725	0,729	0,661	0,938	0,667	0,710	0,694	0,618	0,569
waveform	0,989	0,918	0,939	0,883	0,987	0,813	0,939	0,956	0,946	0,958	-
wine	1,000	1,000	1,000	1,000	1,000	0,938	1,000	1,000	1,000	1,000	0,950
yeast	0,673	0,601	0,618	0,615	0,642	0,906	0,605	0,617	0,652	0,601	0,583
zoo	1,000	1,000	1,000	1,000	1,000	0,844	1,000	1,000	1,000	1,000	-

We can conclude from Table 4.9 that JS outperforms other algorithms on 9 out of 22 datasets by having the most highest scores based on recall scores. The table shows that MPA is the second best algorithm. SSA, WOA and GA are the third best algorithms. JS enhances the recall score of 15 datasets after the feature selection process. The results on Table 4.8 also shows that traditional method CHI2 have 4 highest scores among others.

Table 4.10. Recall results using support vector machine

Classifier		Support Vector Machine									
Dataset Name	Recall without FS	Recall									
		SEA CREATURES						POPULAR OPTIMIZERS			TRADITIONAL
		EPO	MPA	SSA	JS	SFO	WOA	ACO	GA	PSO	CHI2
breast	0,622	0,638	0,629	0,612	0,931	0,844	0,875	0,605	0,724	0,612	0,661
breastcancer	0,973	0,929	0,918	0,957	0,974	0,906	0,954	0,956	0,915	0,929	0,945
car	0,923	0,896	0,908	0,893	0,824	0,813	0,927	0,897	0,889	0,893	0,911
clean1	0,896	0,943	0,955	0,940	0,727	0,844	0,933	0,875	0,965	0,940	0,722
clean2	0,849	0,933	0,919	0,847	0,911	0,906	0,840	0,930	0,859	0,844	0,872
colon	1,000	0,929	1,000	0,500	1,000	0,844	1,000	0,824	0,667	0,722	0,929
credit-g	0,796	0,777	0,799	0,754	0,971	0,813	0,727	0,792	0,834	0,778	0,723
dermatology	1,000	1,000	0,969	1,000	0,750	0,906	1,000	1,000	1,000	1,000	1,000
ecoli	0,929	0,979	0,957	0,957	0,667	0,719	0,958	0,978	0,974	0,979	1,000
glass	0,650	0,813	0,842	0,667	1,000	0,875	0,692	0,867	0,739	0,867	0,619
heart-c	0,800	0,808	0,854	0,820	0,750	0,750	0,906	0,860	0,865	0,811	0,768
ionosphere	0,879	1,000	0,972	0,900	0,667	0,781	1,000	0,977	0,976	1,000	0,906
kr-vs-kp	0,973	0,953	0,991	0,959	1,000	0,813	0,965	0,982	0,995	0,991	0,964
leukemia	1,000	0,850	0,500	0,762	0,904	0,781	1,000	1,000	0,571	0,810	0,571
lymphography	0,714	1,000	1,000	1,000	1,000	0,688	0,889	0,619	1,000	1,000	-
sonar	0,778	0,952	1,000	1,000	1,000	0,969	0,786	1,000	0,955	0,955	0,786
spect	0,500	0,571	0,800	0,846	0,909	0,781	0,857	0,700	0,727	0,867	-
tic-tac-toe	1,000	0,810	1,000	1,000	0,833	0,781	1,000	1,000	1,000	1,000	0,426
waveform	0,942	0,922	0,933	0,895	0,914	0,875	0,924	0,928	0,924	0,926	-
wine	1,000	1,000	1,000	0,933	1,000	0,938	1,000	1,000	0,950	1,000	1,000
yeast	0,530	0,598	0,585	0,624	0,693	0,781	0,614	0,578	0,553	0,573	0,603
zoo	1,000	1,000	1,000	1,000	1,000	0,781	1,000	1,000	1,000	1,000	1,000

We can conclude from Table 4.10 that JS and WOA outperform other algorithms on 9 out of 22 datasets by having the most highest scores based on recall scores. The table shows that MPA and ACO are the second best algorithms. PSO and EPO are the third best algorithms. JS enhances the recall score of 13 datasets after the feature selection process. The results on Table 4.10 also shows that traditional method CHI2 have 4 highest scores among others.

Table 4.11. Recall results using k-nearest neighbor

Classifier		K-Nearest Neighbor									
Dataset Name	Recall without FS	Recall									
		SEA CREATURES						POPULAR OPTIMIZERS			TRADITIONAL
		EPO	MPA	SSA	JS	SFO	WOA	ACO	GA	PSO	CHI2
breast	0,906	0,953	0,970	0,946	0,962	0,947	0,957	0,953	0,936	0,945	0,958
breastcancer	0,975	0,986	0,959	0,959	0,958	0,978	0,950	0,957	0,972	0,926	0,896
car	0,936	0,966	0,958	0,976	0,973	0,954	0,973	0,974	0,965	0,963	0,738
clean1	0,949	0,918	0,916	0,967	0,984	1,000	0,961	0,974	1,000	0,985	0,813
clean2	0,967	0,963	0,978	0,974	0,970	1,000	0,972	0,975	0,984	0,979	0,955
colon	0,750	0,929	0,917	0,692	0,417	0,615	0,923	0,909	1,000	0,583	0,727
credit-g	0,711	0,757	0,784	0,782	0,731	0,745	0,821	0,756	0,821	0,819	0,852
dermatology	0,969	1,000	1,000	1,000	1,000	0,903	1,000	1,000	1,000	1,000	1,000
ecoli	0,979	1,000	1,000	0,974	1,000	0,985	1,000	0,978	1,000	1,000	0,979
glass	0,737	0,800	0,692	0,828	0,778	0,744	0,750	0,842	0,739	0,759	0,808
heart-c	0,632	0,804	0,880	0,822	0,745	0,814	0,842	0,808	0,884	0,826	0,613
ionosphere	0,880	0,931	1,000	0,921	1,000	0,914	1,000	0,971	1,000	1,000	1,000
kr-vs-kp	0,979	0,985	0,986	0,976	0,987	0,998	0,988	0,998	0,993	0,991	1,000
leukemia	0,857	1,000	1,000	1,000	0,786	1,000	1,000	1,000	1,000	1,000	1,000
lymphography	0,762	0,871	0,909	0,917	1,000	0,900	0,911	0,923	0,926	0,969	-
sonar	0,667	0,923	0,963	0,852	0,957	1,000	0,880	1,000	1,000	0,900	0,714
spect	0,348	0,692	0,800	0,692	0,667	1,000	0,688	0,733	1,000	0,750	-
tic-tac-toe	1,000	0,976	1,000	0,971	1,000	0,995	1,000	1,000	1,000	1,000	-
waveform	0,881	0,870	0,919	0,926	0,897	0,826	0,901	0,883	0,913	0,914	-
wine	0,952	0,941	0,955	1,000	1,000	1,000	0,938	1,000	0,941	0,933	1,000
yeast	0,641	0,659	0,731	0,577	0,702	0,562	0,706	0,674	0,651	0,654	0,581
zoo	1,000	1,000	1,000	1,000	1,000	0,950	1,000	1,000	1,000	1,000	1,000

We can conclude from Table 4.11 that GA outperforms other algorithms on half of the datasets by having the most highest scores based on recall scores. The table shows that MPA is the second best algorithm. JS and ACO are the third best algorithms. JS enhances the recall score of 18 datasets after the feature selection process. The results on Table 4.11 also shows that traditional method CHI2 have 7 highest scores among others.

Table 4.12. Recall results using random forest

Classifier		Random Forest									
Classifier		Support Vector Machine									
Dataset Name	Recall without FS	Recall									
		SEA CREATURES						POPULAR OPTIMIZERS			TRADITIONAL
		EPO	MPA	SSA	JS	SFO	WOA	ACO	GA	PSO	CHI2
breast	0,953	0,971	0,960	0,925	0,960	1,000	0,917	0,934	0,981	0,961	0,971
breastcancer	0,907	0,946	0,922	0,923	0,930	0,813	0,901	0,928	0,988	0,959	0,914
car	0,994	1,000	1,000	1,000	1,000	0,781	1,000	0,991	1,000	1,000	1,000
clean1	0,841	0,901	0,928	0,863	0,778	0,906	0,854	0,827	0,933	0,918	0,819
clean2	0,970	0,984	0,987	0,982	1,000	0,813	0,986	0,974	0,987	0,983	0,965
colon	0,875	0,917	1,000	0,909	0,976	0,844	0,846	0,917	0,929	0,917	0,727
credit-g	0,834	0,811	0,818	0,784	0,944	0,813	0,802	0,833	0,799	0,773	0,718
dermatology	1,000	1,000	1,000	1,000	0,833	0,844	1,000	1,000	1,000	1,000	1,000
ecoli	0,975	0,978	0,953	0,977	0,941	0,938	0,981	0,974	0,978	1,000	0,950
glass	0,625	1,000	0,800	0,952	0,981	0,875	0,938	0,739	0,864	0,810	0,773
heart-c	0,820	0,796	0,887	0,857	0,909	0,813	0,850	0,855	0,935	0,870	0,884
ionosphere	0,971	1,000	1,000	0,939	1,000	0,906	0,892	0,925	0,943	0,946	0,806
kr-vs-kp	0,984	0,967	0,995	0,974	0,818	0,938	0,996	0,989	0,993	0,992	0,970
leukemia	0,938	1,000	1,000	1,000	0,615	0,906	1,000	1,000	0,917	1,000	0,917
lymphography	0,905	0,808	1,328	1,193	0,879	0,844	0,893	0,960	0,731	1,258	-
sonar	0,633	0,889	0,846	0,828	0,863	0,969	0,815	0,793	0,906	0,933	0,615
spect	0,333	0,727	0,947	0,786	1,000	0,875	0,545	0,733	0,833	0,917	-
tic-tac-toe	0,909	0,940	0,973	0,903	0,701	0,813	0,967	0,910	0,944	0,892	0,540
waveform	0,860	0,857	0,862	0,847	1,000	0,938	0,856	0,837	0,873	0,881	-
wine	1,000	1,000	1,000	1,000	0,859	0,875	1,000	1,000	0,933	1,000	1,000
yeast	0,575	0,643	0,681	0,631	0,978	0,844	0,622	0,615	0,630	0,645	0,539
zoo	1,000	1,000	1,000	1,000	0,917	0,781	1,000	1,000	1,000	1,000	1,000

We can conclude from Table 4.12 that MPA outperforms other algorithms on 13 out of 22 datasets by having the most highest scores based on recall scores. The table shows that EPO, JS and GA are the second best algorithms. WOA and PSO are the third best algorithms. JS enhances the recall score of 13 datasets after the feature selection process. The results on Table 4.12 also shows that traditional method CHI2 have 7 highest scores among others.

4.2.4. Based on F-Score

Table 4.13. F-score results using naïve bayes

Classifier		Naive Bayes									
Dataset Name	F-Score without FS	F-Score									
		SEA CREATURES						POPULAR OPTIMIZERS			TRADITIONAL
		EPO	MPA	SSA	JS	SFO	WOA	ACO	GA	PSO	CHI2
breast	0,922	0,942	0,954	0,941	0,919	0,833	0,944	0,935	0,954	0,942	0,911
breastcancer	0,944	0,976	0,966	0,944	0,974	0,952	0,967	0,966	0,954	0,971	0,958
car	0,854	0,874	0,856	0,859	0,883	0,938	0,876	0,850	0,861	0,858	0,865
clean1	0,700	0,865	0,919	0,845	0,807	0,889	0,849	0,864	0,916	0,887	0,677
clean2	0,903	0,942	0,952	0,906	0,911	0,892	0,935	0,924	0,934	0,928	0,774
colon	0,870	0,966	1,000	0,833	0,700	0,885	0,947	0,960	0,625	0,960	0,846
credit-g	0,814	0,850	0,849	0,825	0,801	0,833	0,836	0,823	0,875	0,850	0,777
dermatology	0,985	1,000	1,000	1,000	1,000	0,896	1,000	1,000	1,000	1,000	1,000
ecoli	0,857	0,902	0,930	0,916	0,977	0,822	0,923	0,938	0,871	0,885	0,905
glass	0,087	0,615	0,421	0,571	0,357	0,767	0,607	0,450	0,483	0,625	0,509
heart-c	0,830	0,860	0,897	0,925	0,926	0,857	0,884	0,874	0,870	0,843	0,835
ionosphere	0,473	0,829	0,774	0,556	0,468	0,871	0,813	0,712	0,750	0,759	0,550
kr-vs-kp	0,843	0,938	0,943	0,926	0,920	0,873	0,954	0,934	0,952	0,955	0,891
leukemia	1,000	1,000	1,000	1,000	0,952	0,906	0,960	1,000	1,000	1,000	0,929
lymphography	0,725	0,879	0,834	0,844	0,880	0,900	0,892	0,870	0,852	0,833	-
sonar	0,727	0,821	0,933	0,800	0,667	0,896	0,889	0,857	0,949	0,889	0,774
spect	0,750	0,841	0,105	0,100	0,143	0,875	0,869	0,857	0,852	0,867	-
tic-tac-toe	0,497	0,557	0,581	0,577	0,522	0,909	0,526	0,533	0,562	0,556	0,580
waveform	0,820	0,879	0,878	0,867	0,876	0,825	0,876	0,879	0,876	0,873	-
wine	0,929	1,000	0,966	1,000	0,971	0,923	1,000	1,000	1,000	1,000	0,974
yeast	0,709	0,684	0,627	0,674	0,692	0,921	0,650	0,694	0,672	0,652	0,609
zoo	1,000	1,000	1,000	1,000	1,000	0,885	1,000	1,000	1,000	1,000	-

We can conclude from Table 4.13 that GA outperforms other algorithms on 10 out of 22 datasets by having the most highest scores based on F-score. The table shows that MPA is the second best algorithm. SFO is the third best algorithms. JS enhances the F-score of 13 datasets after the feature selection process. The results on Table 4.13 also shows that traditional method CHI2 have 1 highest score among others.

Table 4.14. F-score results using support vector machine

Classifier		Support Vector Machine									
Dataset Name	F-Score without FS	F-Score									
		SEA CREATURES						POPULAR OPTIMIZERS			TRADITIONAL
		EPO	MPA	SSA	JS	SFO	WOA	ACO	GA	PSO	CHI2
breast	0,578	0,764	0,769	0,759	0,909	0,844	0,888	0,751	0,840	0,759	0,673
breastcancer	0,947	0,956	0,937	0,978	0,957	0,866	0,954	0,949	0,956	0,942	0,939
car	0,947	0,916	0,931	0,925	0,837	0,813	0,940	0,929	0,914	0,927	0,888
clean1	0,890	0,960	0,977	0,955	0,800	0,806	0,892	0,906	0,976	0,969	0,782
clean2	0,769	0,960	0,956	0,917	0,872	0,829	0,913	0,959	0,924	0,916	0,863
colon	0,923	0,963	1,000	0,667	0,839	0,806	1,000	0,903	0,800	0,839	0,897
credit-g	0,805	0,828	0,860	0,827	0,939	0,825	0,812	0,858	0,883	0,853	0,839
dermatology	1,000	1,000	0,984	1,000	0,750	0,841	1,000	1,000	1,000	1,000	1,000
ecoli	0,963	0,989	0,978	0,978	0,563	0,780	0,979	0,989	0,987	0,989	1,000
glass	0,650	0,703	0,821	0,651	0,975	0,812	0,800	0,743	0,810	0,788	0,605
heart-c	0,808	0,848	0,872	0,870	0,750	0,814	0,897	0,843	0,865	0,851	0,804
ionosphere	0,866	0,984	0,972	0,915	0,563	0,820	0,895	0,977	0,976	0,981	0,829
kr-vs-kp	0,953	0,967	0,992	0,963	0,975	0,788	0,954	0,990	0,994	0,988	0,888
leukemia	0,857	0,919	0,667	0,865	0,917	0,806	1,000	1,000	0,727	0,895	0,727
lymphography	0,800	1,000	1,000	1,000	1,000	0,759	0,923	0,765	1,000	1,000	-
sonar	0,750	0,851	0,945	0,897	1,000	0,861	0,830	0,889	0,933	0,955	0,721
spect	0,529	0,667	0,727	0,786	0,632	0,806	0,727	0,651	0,533	0,765	-
tic-tac-toe	0,969	0,795	0,981	0,983	0,909	0,806	0,975	0,973	0,950	0,950	0,529
waveform	0,919	0,922	0,933	0,919	0,916	0,875	0,919	0,915	0,930	0,937	-
wine	1,000	1,000	1,000	0,966	0,971	0,845	1,000	1,000	0,974	1,000	0,966
yeast	0,660	0,726	0,700	0,730	0,746	0,820	0,692	0,697	0,674	0,689	0,700
zoo	1,000	1,000	1,000	1,000	1,000	0,794	1,000	1,000	1,000	1,000	1,000

We can conclude from Table 4.14 that MPA outperforms other algorithms on 8 out of 22 datasets by having the most highest scores based on F-score. The table shows that WOA and PSO are the second best algorithms. EPO and JS are the third best algorithms. JS enhances the F-score of 12 datasets after the feature selection process. The results on Table 4.14 also shows that traditional method CHI2 have 3 highest score among others.

Table 4.15. F-score results using k-nearest neighbor

Classifier		K-Nearest Neighbor									
Dataset Name	F-Score without FS	F-Score									
		SEA CREATURES						POPULAR OPTIMIZERS			TRADITIONAL
		EPO	MPA	SSA	JS	SFO	WOA	ACO	GA	PSO	CHI2
breast	0,942	0,953	0,985	0,964	0,955	0,947	0,969	0,967	0,962	0,967	0,948
breastcancer	0,981	0,986	0,973	0,973	0,972	0,978	0,958	0,971	0,972	0,943	0,930
car	0,963	0,972	0,969	0,973	0,973	0,954	0,970	0,972	0,962	0,976	0,849
clean1	0,818	0,899	0,927	0,847	0,943	1,000	0,919	0,892	0,986	0,892	0,793
clean2	0,979	0,975	0,986	0,980	0,979	1,000	0,981	0,982	0,989	0,987	0,961
colon	0,750	0,963	0,957	0,783	0,556	0,615	0,960	0,952	1,000	0,700	0,800
credit-g	0,763	0,832	0,842	0,840	0,803	0,745	0,841	0,822	0,858	0,834	0,718
dermatology	0,954	1,000	1,000	1,000	1,000	0,903	0,986	1,000	1,000	1,000	1,000
ecoli	0,989	1,000	1,000	0,987	1,000	0,985	1,000	0,989	1,000	1,000	0,989
glass	0,800	0,833	0,667	0,873	0,840	0,744	0,818	0,889	0,829	0,815	0,857
heart-c	0,686	0,837	0,889	0,851	0,745	0,814	0,889	0,875	0,826	0,844	0,731
ionosphere	0,733	0,900	0,971	0,886	0,774	0,914	0,919	0,943	0,928	0,932	0,735
kr-vs-kp	0,915	0,920	0,967	0,925	0,912	0,998	0,945	0,943	0,979	0,976	0,668
leukemia	0,923	1,000	1,000	1,000	0,880	1,000	1,000	1,000	1,000	1,000	1,000
lymphography	0,757	0,931	0,930	1,180	0,945	0,900	0,913	1,200	0,926	0,969	-
sonar	0,667	0,906	0,881	0,852	0,844	1,000	0,863	0,898	0,951	0,915	0,652
spect	0,410	0,545	0,744	0,643	0,500	1,000	0,667	0,688	0,286	0,706	-
tic-tac-toe	0,995	0,902	0,995	0,805	0,990	0,995	0,990	0,990	0,986	0,995	-
waveform	0,891	0,867	0,908	0,894	0,885	0,826	0,894	0,892	0,906	0,913	-
wine	0,976	0,941	0,977	0,973	1,000	1,000	0,968	1,000	0,970	0,966	0,957
yeast	0,687	0,648	0,718	0,652	0,683	0,562	0,744	0,690	0,674	0,692	0,649
zoo	1,000	1,000	1,000	1,000	1,000	0,950	1,000	1,000	1,000	1,000	1,000

We can conclude from Table 4.15 that MPA, SFO and PSO outperform other algorithms on 8 out of 22 datasets by having the most highest scores based on F-score. The table shows that WOA and ACO are the second best algorithms. GA and JS are the third best algorithms. JS enhances the F-score of 17 datasets after the feature selection process. The results on Table 4.15 also shows that traditional method CHI2 have 3 highest score among others.

Table 4.16. F-score results using random forest

Classifier		Random Forest									
Dataset Name	F-Score without FS	F-Score									
		SEA CREATURES						POPULAR OPTIMIZERS			TRADITIONAL
		EPO	MPA	SSA	JS	SFO	WOA	ACO	GA	PSO	CHI2
breast	0,944	0,944	0,969	0,952	0,946	0,877	0,935	0,952	0,976	0,975	0,939
breastcancer	0,889	0,972	0,954	0,930	0,930	0,788	0,948	0,957	0,982	0,966	0,919
car	0,983	0,987	0,974	0,973	0,992	0,769	0,987	0,983	0,990	0,977	0,880
clean1	0,831	0,883	0,933	0,890	0,800	0,829	0,905	0,859	0,949	0,929	0,771
clean2	0,973	0,985	0,990	0,986	1,000	0,852	0,989	0,980	0,989	0,988	0,965
colon	0,903	0,957	1,000	0,952	0,988	0,818	0,917	0,957	0,963	0,957	0,800
credit-g	0,819	0,856	0,843	0,812	0,919	0,839	0,838	0,843	0,852	0,825	0,741
dermatology	0,985	1,000	1,000	1,000	0,840	0,857	1,000	1,000	1,000	1,000	1,000
ecoli	0,987	0,978	0,965	0,989	0,901	0,845	0,990	0,987	0,978	1,000	0,974
glass	0,606	0,875	0,873	0,851	0,973	0,812	0,938	0,791	0,864	0,829	0,850
heart-c	0,804	0,804	0,895	0,848	0,917	0,788	0,840	0,895	0,905	0,879	0,784
ionosphere	0,825	0,939	0,943	0,925	1,000	0,866	0,917	0,925	0,957	0,933	0,784
kr-vs-kp	0,992	0,982	0,998	0,981	0,765	0,811	0,996	0,991	0,993	0,995	0,899
leukemia	0,938	1,000	1,000	0,957	0,667	0,817	1,000	0,929	0,957	0,963	0,880
lymphography	0,776	0,875	1,396	1,190	0,874	0,818	0,893	0,923	0,826	1,326	-
sonar	0,704	0,873	0,815	0,828	0,850	0,861	0,846	0,793	0,866	0,918	0,582
spect	0,438	0,667	0,818	0,733	0,966	0,889	0,522	0,688	0,741	0,710	-
tic-tac-toe	0,905	0,926	0,849	0,889	0,704	0,800	0,904	0,876	0,929	0,888	0,551
waveform	0,865	0,855	0,863	0,856	1,000	0,833	0,870	0,841	0,862	0,884	-
wine	1,000	0,903	1,000	1,000	0,852	0,778	0,973	1,000	0,966	1,000	1,000
yeast	0,634	0,651	0,670	0,643	0,983	0,857	0,624	0,593	0,648	0,669	0,537
zoo	1,000	1,000	1,000	1,000	0,857	0,806	1,000	1,000	1,000	1,000	1,000

We can conclude from Table 4.16 that JS outperforms other algorithms on 9 out of 22 datasets by having the most highest scores based on F-score. The table shows that GA is the second best algorithms. MPA is the third best algorithm. JS enhances the F-score of 14 datasets after the feature selection process. The results on Table 4.15 also shows that traditional method CHI2 have 3 highest score among others.

4.2.5. Based on Selection Size

Selection size is one of the evaluation criteria in this thesis. It represents ratio of features that are selected from set of features. It is important for feature selection process to select as less features as possible. We cannot evaluate two algorithms as same if they choose 5 features from two datasets that have 10 and 100 features. Choosing 5 features from 100 features is considered more successful.

Table 4.17. Comparison of the algorithms inspired by sea creatures based on selection size using KNN

Classifier		K-Nearest Neighbor											
Dataset Name	No. Of Features	Number of Selected Features											
		SEA CREATURES											
		EPO	SS	MPA	SS	SSA	SS	JS	SS	SFO	SS	WOA	SS
breast	30	2	6,67%	10	33,33%	12	40,00%	17	56,67%	3	10,00%	22	73,33%
breastcancer	9	5	55,56%	5	55,56%	4	44,44%	60	666,67%	2	22,22%	3	33,33%
car	6	5	83,33%	5	83,33%	5	83,33%	4	66,67%	5	83,33%	5	83,33%
clean1	166	13	7,83%	62	37,35%	78	46,99%	136	81,93%	57	34,34%	22	13,25%
clean2	166	45	27,11%	93	56,02%	84	50,60%	87	52,41%	60	36,14%	91	54,82%
colon	2000	3	0,15%	45	2,25%	923	46,15%	995	49,75%	1253	62,65%	2	0,10%
credit-g	20	4	20,00%	8	40,00%	11	55,00%	10	50,00%	1	5,00%	14	70,00%
dermatology	34	12	35,29%	9	26,47%	15	44,12%	20	58,82%	9	26,47%	14	41,18%
ecoli	7	4	57,14%	4	57,14%	6	85,71%	4	57,14%	1	14,29%	6	85,71%
glass	9	5	55,56%	3	33,33%	4	44,44%	4	44,44%	2	22,22%	5	55,56%
heart-c	13	3	23,08%	6	46,15%	4	30,77%	7	53,85%	2	15,38%	3	23,08%
ionosphere	34	3	8,82%	4	11,76%	8	23,53%	16	47,06%	6	17,65%	8	23,53%
kr-vs-kp	36	10	27,78%	21	58,33%	19	52,78%	15	41,67%	4	11,11%	17	47,22%
leukemia	7129	1	0,01%	165	2,31%	3427	48,07%	3571	50,09%	3893	54,61%	118	1,66%
lymphograph	18	4	22,22%	5	27,78%	4	22,22%	9	50,00%	7	38,89%	5	27,78%
sonar	60	7	11,67%	15	25,00%	33	55,00%	42	70,00%	10	16,67%	16	26,67%
spect	22	7	31,82%	11	50,00%	11	50,00%	18	81,82%	1	4,55%	14	63,64%
tic-tac-toe	9	8	88,89%	9	100,00%	7	77,78%	4	44,44%	1	11,11%	9	100,00%
waveform	40	19	47,50%	25	62,50%	22	55,00%	25	62,50%	30	75,00%	26	65,00%
wine	13	2	15,38%	4	30,77%	5	38,46%	7	53,85%	1	7,69%	3	23,08%
yeast	8	5	62,50%	6	75,00%	6	75,00%	7	87,50%	4	50,00%	7	87,50%
zoo	16	4	25,00%	5	31,25%	6	37,50%	8	50,00%	4	25,00%	7	43,75%

It can be concluded from Table 4.17 that SFO selects less features than other algorithms on 12 out of 22 datasets. Following SFO, EPO selectes less features than other algorithms on 9 out of 22 datasets. As it is shown in the table, JS performs poorly about selection size. When compared to SFO, even though JS performs better than SFO on 4 datasets, the difference between selection size is larger than it should be for the cases that SFO outperforms JS. It can be concluded that EPO and SFO outperforms other algorithms based on selection size.

Table 4.18. Comparison of the popular optimizers and CHI2 based on selection size using

Classifier		K-Nearest Neighbor							
Dataset Name	No. Of Features	Number of Selected Features							
		POPULAR OPTIMIZERS						TRADITIONAL	
		ACO	SS	GA	SS	PSO	SS	CHI2	SS
breast	30	16	53,33%	11	36,67%	12	40,00%	10	33,33%
breastcancer	9	6	66,67%	4	44,44%	3	33,33%	8	88,89%
car	6	5	83,33%	5	83,33%	5	83,33%	2	33,33%
clean1	166	56	33,73%	66	39,76%	77	46,39%	21	12,65%
clean2	166	42	25,30%	84	50,60%	72	43,37%	14	8,43%
colon	2000	105	5,25%	820	41,00%	859	42,95%	39	1,95%
credit-g	20	7	35,00%	9	45,00%	13	65,00%	1	5,00%
dermatology	34	18	52,94%	10	29,41%	11	32,35%	19	55,88%
ecoli	7	5	71,43%	6	85,71%	4	57,14%	5	71,43%
glass	9	6	66,67%	3	33,33%	4	44,44%	6	66,67%
heart-c	13	7	53,85%	5	38,46%	3	23,08%	7	53,85%
ionosphere	34	3	8,82%	7	20,59%	11	32,35%	19	55,88%
kr-vs-kp	36	24	66,67%	17	47,22%	16	44,44%	3	8,33%
leukemia	7129	10	0,14%	3062	42,95%	3332	46,74%	52	0,73%
lymphograph	18	5	27,78%	6	33,33%	7	38,89%	3	16,67%
sonar	60	29	48,33%	16	26,67%	26	43,33%	7	11,67%
spect	22	11	50,00%	8	36,36%	10	45,45%	-	-
tic-tac-toe	9	9	100,00%	9	100,00%	9	100,00%	1	11,11%
waveform	40	32	80,00%	21	52,50%	23	57,50%	-	-
wine	13	7	53,85%	5	38,46%	4	30,77%	6	46,15%
yeast	8	6	75,00%	8	100,00%	6	75,00%	5	62,50%
zoo	16	7	43,75%	7	43,75%	7	43,75%	4	25,00%

KNN

It can be concluded from Table 4.18 that traditional method, CHI2, outperforms popular optimizers based on selection size. CHI2 outperforms other methods on 12 out of 22 datasets.

Table 4.19. Comparison of the best algorithms based on selection size using KNN

Classifier		K-Nearest Neighbor							
Dataset Name	No. Of Features	Number of Selected Features							
		Proposed Method		Sea Creature		Popular Method		Traditional	
		JS	SS	SFO	SS	PSO	SS	CHI2	SS
breast	30	17	56,67%	3	10,00%	12	40,00%	10	33,33%
breastcancer	9	6	66,67%	2	22,22%	3	33,33%	8	88,89%
car	6	4	66,67%	5	83,33%	5	83,33%	2	33,33%
clean1	166	136	81,93%	57	34,34%	77	46,39%	21	12,65%
clean2	166	87	52,41%	60	36,14%	72	43,37%	14	8,43%
colon	2000	995	49,75%	1253	62,65%	859	42,95%	39	1,95%
credit-g	20	10	50,00%	1	5,00%	13	65,00%	1	5,00%
dermatology	34	20	58,82%	9	26,47%	11	32,35%	19	55,88%
ecoli	7	4	57,14%	1	14,29%	4	57,14%	5	71,43%
glass	9	4	44,44%	2	22,22%	4	44,44%	6	66,67%
heart-c	13	7	53,85%	2	15,38%	3	23,08%	7	53,85%
ionosphere	34	16	47,06%	6	17,65%	11	32,35%	19	55,88%
kr-vs-kp	36	15	41,67%	4	11,11%	16	44,44%	3	8,33%
leukemia	7129	3571	50,09%	3893	54,61%	3332	46,74%	52	0,73%
lymphograph	18	9	50,00%	7	38,89%	7	38,89%	3	16,67%
sonar	60	42	70,00%	10	16,67%	26	43,33%	7	11,67%
spect	22	18	81,82%	1	4,55%	10	45,45%	-	-
tic-tac-toe	9	4	44,44%	1	11,11%	9	100,00%	1	11,11%
waveform	40	25	62,50%	30	75,00%	23	57,50%	-	-
wine	13	7	53,85%	1	7,69%	4	30,77%	6	46,15%
yeast	8	7	87,50%	4	50,00%	6	75,00%	5	62,50%
zoo	16	8	50,00%	4	25,00%	7	43,75%	4	25,00%

Table 4.19 showed that SFO outperforms other methods based on selection size. Between SFO and CHI2, SFO outperforms CHI2 with one more datasets by selecting less features on 12 out of 22 features. It is noticeable that JS cannot outperforms other methods on any of the datasets.

Table 4.20. Comparison of the algorithms inspired by sea creatures based on selection size using naïve bayes

Classifier		Naive Bayes											
Dataset Name	No. Of Features	Number of Selected Features											
		SEA CREATURES											
		EPO	SS	MPA	SS	SSA	SS	JS	SS	SFO	SS	WOA	SS
breast	30	8	26,67%	14	46,67%	16	53,33%	18	60,00%	23	76,67%	21	70,00%
breastcancer	9	4	44,44%	4	44,44%	6	66,67%	6	66,67%	3	33,33%	4	44,44%
car	6	1	16,67%	1	16,67%	1	16,67%	4	66,67%	6	100,00%	1	16,67%
clean1	166	24	14,46%	70	42,17%	95	57,23%	73	43,98%	36	21,69%	64	38,55%
clean2	166	5	3,01%	17	10,24%	91	54,82%	82	49,40%	58	34,94%	10	6,02%
colon	2000	1	0,05%	20	1,00%	978	48,90%	976	48,80%	981	49,05%	30	1,50%
credit-g	20	8	40,00%	9	45,00%	10	50,00%	11	55,00%	5	25,00%	8	40,00%
dermatology	34	8	23,53%	7	20,59%	14	41,18%	16	47,06%	21	61,76%	9	26,47%
ecoli	7	3	42,86%	3	42,86%	2	28,57%	5	71,43%	1	14,29%	5	71,43%
glass	9	1	11,11%	2	22,22%	4	44,44%	4	44,44%	3	33,33%	1	11,11%
heart-c	13	6	46,15%	5	38,46%	5	38,46%	8	61,54%	10	76,92%	7	53,85%
ionosphere	34	1	2,94%	1	2,94%	9	26,47%	18	52,94%	13	38,24%	1	2,94%
kr-vs-kp	36	4	11,11%	12	33,33%	16	44,44%	22	61,11%	9	25,00%	13	36,11%
leukemia	7129	5	0,07%	80	1,12%	3414	47,89%	3510	49,24%	3516	49,32%	76	1,07%
lymphography	18	8	44,44%	8	44,44%	10	55,56%	7	38,89%	12	66,67%	12	66,67%
sonar	60	4	6,67%	30	50,00%	26	43,33%	24	40,00%	10	16,67%	8	13,33%
spect	22	1	4,55%	7	31,82%	11	50,00%	11	50,00%	3	13,64%	1	4,55%
tic-tac-toe	9	3	33,33%	7	77,78%	6	66,67%	5	55,56%	1	11,11%	6	66,67%
waveform	40	10	25,00%	15	37,50%	20	50,00%	19	47,50%	20	50,00%	23	57,50%
wine	13	5	38,46%	4	30,77%	7	53,85%	10	76,92%	2	15,38%	4	30,77%
yeast	8	6	75,00%	4	50,00%	4	50,00%	4	50,00%	6	75,00%	4	50,00%
zoo	16	8	50,00%	7	43,75%	8	50,00%	10	62,50%	12	75,00%	6	37,50%

It can be concluded from Table 4.20 that EPO selects less features than other algorithms on 11 out of 22 datasets. Following EPO, WOA selectes less features than other algorithms on 6 out of 22 datasets. As it is shown in the table, JS performs poorly about selection size. JS outperforms other algorithms on 2 datasets, one of them being tied up with others. It can be concluded that EPO and WOA outperforms other algorithms based on selection size.

Table 4.21. Comparison of the popular optimizers and CHI2 based on selection size using naïve bayes

Classifier		Naive Bayes							
Dataset Name	No. Of Features	Number of Selected Features							
		POPULAR OPTIMIZERS						TRADITIONAL	
		ACO	SS	GA	SS	PSO	SS	CHI2	SS
breast	30	16	53,33%	12	40,00%	14	46,67%	10	33,33%
breastcancer	9	3	33,33%	2	22,22%	3	33,33%	8	88,89%
car	6	1	16,67%	1	16,67%	1	16,67%	2	33,33%
clean1	166	87	52,41%	63	37,95%	67	40,36%	21	12,65%
clean2	166	47	28,31%	74	44,58%	70	42,17%	14	8,43%
colon	2000	28	1,40%	781	39,05%	883	44,15%	39	1,95%
credit-g	20	8	40,00%	9	45,00%	9	45,00%	1	5,00%
dermatology	34	14	41,18%	8	23,53%	10	29,41%	19	55,88%
ecoli	7	4	57,14%	4	57,14%	4	57,14%	5	71,43%
glass	9	2	22,22%	2	22,22%	3	33,33%	6	66,67%
heart-c	13	5	38,46%	5	38,46%	7	53,85%	7	53,85%
ionosphere	34	2	5,88%	2	5,88%	2	5,88%	19	55,88%
kr-vs-kp	36	20	55,56%	14	38,89%	10	27,78%	3	8,33%
leukemia	7129	148	2,08%	3040	42,64%	3238	45,42%	52	0,73%
lymphography	18	12	66,67%	10	55,56%	6	33,33%	3	16,67%
sonar	60	15	25,00%	18	30,00%	21	35,00%	7	11,67%
spect	22	1	4,55%	1	4,55%	1	4,55%	-	-
tic-tac-toe	9	8	88,89%	3	33,33%	4	44,44%	1	11,11%
waveform	40	31	77,50%	17	42,50%	19	47,50%	-	-
wine	13	5	38,46%	5	38,46%	3	23,08%	6	46,15%
yeast	8	5	62,50%	5	62,50%	5	62,50%	5	62,50%
zoo	16	11	68,75%	6	37,50%	6	37,50%	4	25,00%

It can be concluded from Table 4.21 that traditional method, CHI2, outperforms popular optimizers based on selection size. CHI2 outperforms other methods on 11 out of 22 datasets. Results are better for popular optimizers when it is used naïve bayes as classifier. Compared to Table 4.18, popular optimizers performs better with naïve bayes classifier but they do not outperform CHI2 using those two classifiers.

Table 4.22. Comparison of the best algorithms based on selection size using naïve bayes

Classifier		Naive Bayes							
Dataset Name	No. Of Features	Number of Selected Features							
		Proposed Method		Sea Creature		Popular Method		Traditional	
		JS	SS	EPO	SS	GA	SS	CHI2	SS
breast	30	18	60,00%	8	26,67%	12	40,00%	10	33,33%
breastcancer	9	6	66,67%	4	44,44%	2	22,22%	8	88,89%
car	6	4	66,67%	1	16,67%	1	16,67%	2	33,33%
clean1	166	73	43,98%	24	14,46%	63	37,95%	21	12,65%
clean2	166	82	49,40%	5	3,01%	74	44,58%	14	8,43%
colon	2000	976	48,80%	1	0,05%	781	39,05%	39	1,95%
credit-g	20	11	55,00%	8	40,00%	9	45,00%	1	5,00%
dermatology	34	16	47,06%	8	23,53%	8	23,53%	19	55,88%
ecoli	7	5	71,43%	3	42,86%	4	57,14%	5	71,43%
glass	9	4	44,44%	1	11,11%	2	22,22%	6	66,67%
heart-c	13	8	61,54%	6	46,15%	5	38,46%	7	53,85%
ionosphere	34	18	52,94%	1	2,94%	2	5,88%	19	55,88%
kr-vs-kp	36	22	61,11%	4	11,11%	14	38,89%	3	8,33%
leukemia	7129	3510	49,24%	5	0,07%	3040	42,64%	52	0,73%
lymphography	18	7	38,89%	8	44,44%	10	55,56%	3	16,67%
sonar	60	24	40,00%	4	6,67%	18	30,00%	7	11,67%
spect	22	11	50,00%	1	4,55%	1	4,55%	-	-
tic-tac-toe	9	5	55,56%	3	33,33%	3	33,33%	1	11,11%
waveform	40	19	47,50%	10	25,00%	17	42,50%	-	-
wine	13	10	76,92%	5	38,46%	5	38,46%	6	46,15%
yeast	8	4	50,00%	6	75,00%	5	62,50%	5	62,50%
zoo	16	10	62,50%	8	50,00%	6	37,50%	4	25,00%

Table 4.22 showed that EPO outperforms other algorithms based on selection size. EPO selects less features on 13 out of 22 datasets. It is noticeable that JS outperforms other methods on 1 out of 22 datasets which shows that JS performs poorly for selection size using naïve bayes classifier.

Table 4.23. Comparison of the algorithms inspired by sea creatures based on selection size using support vector machine

Classifier		Support Vector Machine											
Dataset Name	No. Of Features	Number of Selected Features											
		SEA CREATURES											
		EPO	SS	MPA	SS	SSA	SS	JS	SS	SFO	SS	WOA	SS
breast	30	1	3,33%	1	3,33%	4	13,33%	20	66,67%	9	30,00%	27	90,00%
breastcancer	9	3	33,33%	3	33,33%	3	33,33%	4	44,44%	1	11,11%	1	11,11%
car	6	5	83,33%	6	100,00%	5	83,33%	5	83,33%	5	83,33%	6	100,00%
clean1	166	23	13,86%	46	27,71%	70	42,17%	79	47,59%	28	16,87%	72	43,37%
clean2	166	2	1,20%	3	1,81%	54	32,53%	3	1,81%	48	28,92%	1	0,60%
colon	2000	3	0,15%	4	0,20%	907	45,35%	886	44,30%	488	24,40%	96	4,80%
credit-g	20	4	20,00%	8	40,00%	7	35,00%	18	90,00%	1	5,00%	8	40,00%
dermatology	34	8	23,53%	12	35,29%	19	55,88%	12	35,29%	16	47,06%	16	47,06%
ecoli	7	3	42,86%	4	57,14%	3	42,86%	3	42,86%	1	14,29%	4	57,14%
glass	9	6	66,67%	6	66,67%	4	44,44%	4	44,44%	3	33,33%	2	22,22%
heart-c	13	4	30,77%	4	30,77%	4	30,77%	6	46,15%	2	15,38%	6	46,15%
ionosphere	34	5	14,71%	11	32,35%	11	32,35%	15	44,12%	8	23,53%	20	58,82%
kr-vs-kp	36	8	22,22%	27	75,00%	22	61,11%	16	44,44%	6	16,67%	27	75,00%
leukemia	7129	1	0,01%	1	0,01%	3404	47,75%	1325	18,59%	1236	17,34%	15	0,21%
lymphography	18	11	61,11%	12	66,67%	15	83,33%	12	66,67%	3	16,67%	11	61,11%
sonar	60	9	15,00%	40	66,67%	30	50,00%	34	56,67%	3	5,00%	14	23,33%
spect	22	6	27,27%	7	31,82%	8	36,36%	7	31,82%	1	4,55%	15	68,18%
tic-tac-toe	9	7	77,78%	9	100,00%	9	100,00%	3	33,33%	1	11,11%	9	100,00%
waveform	40	17	42,50%	26	65,00%	27	67,50%	19	47,50%	26	65,00%	27	67,50%
wine	13	8	61,54%	3	23,08%	5	38,46%	8	61,54%	1	7,69%	4	30,77%
yeast	8	6	75,00%	7	87,50%	7	87,50%	8	100,00%	5	62,50%	8	100,00%
zoo	16	4	25,00%	4	25,00%	7	43,75%	14	87,50%	5	31,25%	7	43,75%

Results from Table 4.23 showed that SFO selects less features than other algorithms on 12 out of 22 datasets. Following SFO, EPO selects less features than other algorithms on 8 out of 22 datasets. As it is shown in the table, JS performs poorly about selection size. JS outperforms other algorithms on 1 dataset being tied up with others.

Table 4.24. Comparison of the popular optimizers and CHI2 based on selection size using support vector machine

Classifier		Support Vector Machine							
Dataset Name	No. Of Features	Number of Selected Features							
		POPULAR OPTIMIZERS						TRADITIONAL	
		ACO	SS	GA	SS	PSO	SS	CHI2	SS
breast	30	1	3,33%	1	3,33%	2	6,67%	10	33,33%
breastcancer	9	2	22,22%	3	33,33%	3	33,33%	8	88,89%
car	6	6	100,00%	5	83,33%	6	100,00%	2	33,33%
clean1	166	46	27,71%	54	32,53%	74	44,58%	21	12,65%
clean2	166	2	1,20%	21	12,65%	48	28,92%	14	8,43%
colon	2000	8	0,40%	701	35,05%	872	43,60%	39	1,95%
credit-g	20	5	25,00%	7	35,00%	7	35,00%	1	5,00%
dermatology	34	14	41,18%	9	26,47%	12	35,29%	19	55,88%
ecoli	7	4	57,14%	3	42,86%	5	71,43%	5	71,43%
glass	9	4	44,44%	5	55,56%	5	55,56%	6	66,67%
heart-c	13	4	30,77%	5	38,46%	4	30,77%	7	53,85%
ionosphere	34	9	26,47%	9	26,47%	6	17,65%	19	55,88%
kr-vs-kp	36	28	77,78%	19	52,78%	18	50,00%	3	8,33%
leukemia	7129	136	1,91%	2991	41,96%	3278	45,98%	52	0,73%
lymphography	18	6	33,33%	8	44,44%	12	66,67%	3	16,67%
sonar	60	34	56,67%	13	21,67%	24	40,00%	7	11,67%
spect	22	5	22,73%	5	22,73%	6	27,27%	-	-
tic-tac-toe	9	9	100,00%	9	100,00%	9	100,00%	1	11,11%
waveform	40	29	72,50%	20	50,00%	24	60,00%	-	-
wine	13	7	53,85%	4	30,77%	5	38,46%	6	46,15%
yeast	8	7	87,50%	6	75,00%	8	100,00%	5	62,50%
zoo	16	4	25,00%	4	25,00%	6	37,50%	4	25,00%

It can be concluded from Table 4.24 that traditional method, CHI2, outperforms popular optimizers based on selection size. CHI2 outperforms other methods on 10 out of 22 datasets. Between popular methods, ACO outperforms GA and PSO based on selection size.

Table 4.25. Comparison of the best algorithms based on selection size using support vector machine

Classifier		Support Vector Machine							
Dataset Name	No. Of Features	Number of Selected Features							
		Proposed Method		Sea Creature		Popular Method		Traditional	
		JS	SS	SFO	SS	ACO	SS	CHI2	SS
breast	30	20	66,67%	9	30,00%	1	3,33%	10	33,33%
breastcancer	9	4	44,44%	1	11,11%	2	22,22%	8	88,89%
car	6	5	83,33%	5	83,33%	6	100,00%	2	33,33%
clean1	166	79	47,59%	28	16,87%	46	27,71%	21	12,65%
clean2	166	3	1,81%	48	28,92%	2	1,20%	14	8,43%
colon	2000	886	44,30%	488	24,40%	8	0,40%	39	1,95%
credit-g	20	18	90,00%	1	5,00%	5	25,00%	1	5,00%
dermatology	34	12	35,29%	16	47,06%	14	41,18%	19	55,88%
ecoli	7	3	42,86%	1	14,29%	4	57,14%	5	71,43%
glass	9	4	44,44%	3	33,33%	4	44,44%	6	66,67%
heart-c	13	6	46,15%	2	15,38%	4	30,77%	7	53,85%
ionosphere	34	15	44,12%	8	23,53%	9	26,47%	19	55,88%
kr-vs-kp	36	16	44,44%	6	16,67%	28	77,78%	3	8,33%
leukemia	7129	1325	18,59%	1236	17,34%	136	1,91%	52	0,73%
lymphography	18	12	66,67%	3	16,67%	6	33,33%	3	16,67%
sonar	60	34	56,67%	3	5,00%	34	56,67%	7	11,67%
spect	22	7	31,82%	1	4,55%	5	22,73%	-	-
tic-tac-toe	9	3	33,33%	1	11,11%	9	100,00%	1	11,11%
waveform	40	19	47,50%	26	65,00%	29	72,50%	-	-
wine	13	8	61,54%	1	7,69%	7	53,85%	6	46,15%
yeast	8	8	100,00%	5	62,50%	7	87,50%	5	62,50%
zoo	16	14	87,50%	5	31,25%	4	25,00%	4	25,00%

Table 4.25 showed that SFO outperforms other algorithms based on selection size. SFO selects less features on 12 out of 22 datasets. It is noticeable that JS outperforms other methods on 2 out of 22 datasets which shows that JS performs poorly for selection size using support vector machine classifier.

Table 4.26. Comparison of the algorithms inspired by sea creatures based on selection size using random forest

Classifier		Random Forest											
Dataset Name	No. Of Features	Number of Selected Features											
		SEA CREATURES											
		EPO	SS	MPA	SS	SSA	SS	JS	SS	SFO	SS	WOA	SS
breast	30	5	16,67%	6	20,00%	12	40,00%	10	33,33%	12	40,00%	12	40,00%
astcancer	9	4	44,44%	5	55,56%	4	44,44%	5	55,56%	2	22,22%	5	55,56%
car	6	5	83,33%	5	83,33%	5	83,33%	3	50,00%	5	83,33%	5	83,33%
clean1	166	14	8,43%	21	12,65%	87	52,41%	78	46,99%	60	36,14%	50	30,12%
clean2	166	37	22,29%	62	37,35%	91	54,82%	88	53,01%	53	31,93%	82	49,40%
colon	2000	1	0,05%	50	2,50%	969	48,45%	975	48,75%	1267	63,35%	4	0,20%
redit-g	20	4	20,00%	11	55,00%	11	55,00%	16	80,00%	1	5,00%	5	25,00%
matology	34	7	20,59%	8	23,53%	12	35,29%	14	41,18%	20	58,82%	15	44,12%
ecoli	7	3	42,86%	5	71,43%	5	71,43%	6	85,71%	2	28,57%	6	85,71%
glass	9	3	33,33%	5	55,56%	5	55,56%	6	66,67%	8	88,89%	4	44,44%
heart-c	13	3	23,08%	7	53,85%	6	46,15%	8	61,54%	13	100,00%	4	30,77%
osphere	34	7	20,59%	11	32,35%	16	47,06%	16	47,06%	9	26,47%	11	32,35%
r-vs-kp	36	23	63,89%	28	77,78%	25	69,44%	17	47,22%	1	2,78%	32	88,89%
ukemia	7129	2	0,03%	37	0,52%	3417	47,93%	3625	50,85%	3633	50,96%	26	0,36%
hography	18	3	16,67%	3	16,67%	9	50,00%	8	44,44%	13	72,22%	7	38,89%
sonar	60	6	10,00%	14	23,33%	30	50,00%	33	55,00%	2	3,33%	5	8,33%
spect	22	5	22,73%	8	36,36%	12	54,55%	12	54,55%	1	4,55%	6	27,27%
-tac-toe	9	8	88,89%	9	100,00%	8	88,89%	4	44,44%	1	11,11%	9	100,00%
aveform	40	13	32,50%	13	32,50%	18	45,00%	27	67,50%	11	27,50%	20	50,00%
wine	13	3	23,08%	3	23,08%	6	46,15%	7	53,85%	1	7,69%	3	23,08%
yeast	8	6	75,00%	5	62,50%	7	87,50%	4	50,00%	2	25,00%	6	75,00%
zoo	16	3	18,75%	5	31,25%	6	37,50%	8	50,00%	4	25,00%	4	25,00%

Results from Table 4.25 showed that EPO and SFO select less features than other algorithms on 10 out of 22 datasets. As it is shown in the table, JS and other methods perform poorly about selection size. After EPO and SFO, the best methods(MPA and JS) outperforms other on one dataset. WOA and SSA did not get any success considering this evaluation criteria.

Table 4.27. Comparison of the popular optimizers and CHI2 based on selection size using random forest

Classifier		Random Forest							
Dataset Name	No. Of Features	Number of Selected Features							
		POPULAR OPTIMIZERS						TRADITIONAL	
		ACO	SS	GA	SS	PSO	SS	CHI2	SS
breast	30	12	40,00%	7	23,33%	6	20,00%	10	33,33%
breastcancer	9	4	44,44%	3	33,33%	4	44,44%	8	88,89%
car	6	5	83,33%	5	83,33%	5	83,33%	2	33,33%
clean1	166	58	34,94%	49	29,52%	56	33,73%	21	12,65%
clean2	166	82	49,40%	60	36,14%	85	51,20%	14	8,43%
colon	2000	165	8,25%	768	38,40%	896	44,80%	39	1,95%
credit-g	20	13	65,00%	6	30,00%	8	40,00%	1	5,00%
dermatology	34	12	35,29%	6	17,65%	10	29,41%	19	55,88%
ecoli	7	4	57,14%	4	57,14%	4	57,14%	5	71,43%
glass	9	4	44,44%	5	55,56%	4	44,44%	6	66,67%
heart-c	13	3	23,08%	3	23,08%	7	53,85%	7	53,85%
ionosphere	34	25	73,53%	6	17,65%	9	26,47%	19	55,88%
kr-vs-kp	36	32	88,89%	20	55,56%	18	50,00%	3	8,33%
leukemia	7129	110	1,54%	3088	43,32%	3279	46,00%	52	0,73%
lymphography	18	6	33,33%	5	27,78%	4	22,22%	3	16,67%
sonar	60	12	20,00%	6	10,00%	12	20,00%	7	11,67%
spect	22	4	18,18%	6	27,27%	7	31,82%	-	-
tic-tac-toe	9	8	88,89%	9	100,00%	9	100,00%	1	11,11%
waveform	40	32	80,00%	21	52,50%	19	47,50%	-	-
wine	13	4	30,77%	3	23,08%	4	30,77%	6	46,15%
yeast	8	4	50,00%	6	75,00%	5	62,50%	5	62,50%
zoo	16	5	31,25%	4	25,00%	5	31,25%	4	25,00%

It can be concluded from Table 4.27 that traditional method, CHI2, outperforms popular optimizers based on selection size. CHI2 outperforms other methods on 10 out of 22 datasets. Between popular methods, GA outperforms ACO and PSO based on selection size.

Table 4.28. Comparison of the best algorithms based on selection size using random forest

Classifier		Random Forest									
Dataset Name	No. Of Features	Number of Selected Features									
		Proposed Method		Sea Creature				Popular Method		Traditional	
		JS	SS	EPO	SS	SFO	SS	GA	SS	CHI2	SS
breast	30	10	33,33%	5	16,67%	12	40,00%	7	23,33%	10	33,33%
breastcancer	9	5	55,56%	4	44,44%	2	22,22%	3	33,33%	8	88,89%
car	6	3	50,00%	5	83,33%	5	83,33%	5	83,33%	2	33,33%
clean1	166	78	46,99%	14	8,43%	60	36,14%	49	29,52%	21	12,65%
clean2	166	88	53,01%	37	22,29%	53	31,93%	60	36,14%	14	8,43%
colon	2000	975	48,75%	1	0,05%	1267	63,35%	768	38,40%	39	1,95%
credit-g	20	16	80,00%	4	20,00%	1	5,00%	6	30,00%	1	5,00%
dermatology	34	14	41,18%	7	20,59%	20	58,82%	6	17,65%	19	55,88%
ecoli	7	6	85,71%	3	42,86%	2	28,57%	4	57,14%	5	71,43%
glass	9	6	66,67%	3	33,33%	8	88,89%	5	55,56%	6	66,67%
heart-c	13	8	61,54%	3	23,08%	13	100,00%	3	23,08%	7	53,85%
ionosphere	34	16	47,06%	7	20,59%	9	26,47%	6	17,65%	19	55,88%
kr-vs-kp	36	17	47,22%	23	63,89%	1	2,78%	20	55,56%	3	8,33%
leukemia	7129	3625	50,85%	2	0,03%	3633	50,96%	3088	43,32%	52	0,73%
lymphography	18	8	44,44%	3	16,67%	13	72,22%	5	27,78%	3	16,67%
sonar	60	33	55,00%	6	10,00%	2	3,33%	6	10,00%	7	11,67%
spect	22	12	54,55%	5	22,73%	1	4,55%	6	27,27%		0,00%
tic-tac-toe	9	4	44,44%	8	88,89%	1	11,11%	9	100,00%	1	11,11%
waveform	40	27	67,50%	13	32,50%	11	27,50%	21	52,50%		0,00%
wine	13	7	53,85%	3	23,08%	1	7,69%	3	23,08%	6	46,15%
yeast	8	4	50,00%	6	75,00%	2	25,00%	6	75,00%	5	62,50%
zoo	16	8	50,00%	3	18,75%	4	25,00%	4	25,00%	4	25,00%

Table 4.28 showed that SFO outperforms other algorithms based on selection size. SFO selects less features on 10 out of 22 datasets. It is noticeable that JS cannot outperforms any other methods which shows that JS performs poorly for selection size using random forest classifier. Following SFO, EPO outperforms other methods on 8 out of 22 datasets.

4.2.6. Overall Comparison

Table 4.29. Comparison of the best algorithms based on classification accuracy

Classifier Dataset Name	Naive Bayes				Support Vector Machine				K-Nearest Neighbor				Random Forest			
	Proposed Method (JS)		Best of Accuracy (GA)		Proposed Method (JS)		Best of Accuracy (MPA)		Proposed Method (JS)		Best of Accuracy (SFO) and Best of SS (SFO)		Proposed Method (JS)		Best of Accuracy (GA)	
	Accuracy	SS	Accuracy	SS	Accuracy	SS	Accuracy	SS	Accuracy	SS	Accuracy	SS	Accuracy	SS	Accuracy	SS
breast	0.894	60.00%	0.941	40.00%	0.929	66.67%	0.647	3.33%	0.947	56.67%	0.947	10.00%	0.935	33.33%	0.971	23.33%
breastcancer	0.981	66.67%	0.971	22.22%	0.986	44.44%	0.957	33.33%	0.981	66.67%	0.978	22.22%	0.976	55.56%	0.986	33.33%
car	0.726	66.67%	0.703	16.67%	0.714	16.67%	0.855	100.00%	0.950	66.67%	0.954	83.33%	0.967	50.00%	0.971	83.33%
clean1	0.775	43.98%	0.901	37.95%	0.852	14.46%	0.717	47.59%	0.972	27.71%	1.000	34.34%	0.866	46.99%	0.937	29.52%
clean2	0.852	49.40%	0.888	44.58%	0.897	3.01%	0.867	1.81%	0.944	81.93%	1.000	36.14%	0.971	53.01%	0.981	36.14%
colon	0.667	48.80%	0.667	39.05%	0.944	0.05%	0.905	44.30%	1.000	20.00%	0.615	62.65%	0.889	48.75%	0.944	38.40%
credit-g	0.743	55.00%	0.817	45.00%	0.767	40.00%	0.942	90.00%	0.783	40.00%	0.745	5.00%	0.717	80.00%	0.783	30.00%
dermatology	0.982	47.06%	0.991	23.53%	0.991	23.53%	0.774	35.29%	0.982	35.29%	0.903	26.47%	0.991	41.18%	0.991	17.65%
ecoli	0.600	71.43%	0.670	57.14%	0.640	42.86%	0.850	42.86%	0.810	57.14%	0.985	14.29%	0.872	85.71%	0.860	57.14%
glass	0.484	44.44%	0.500	22.22%	0.391	11.11%	0.983	44.44%	0.719	66.67%	0.744	22.22%	0.797	66.67%	0.781	55.56%
heart-c	0.922	61.54%	0.867	38.46%	0.867	46.15%	0.774	46.15%	0.867	30.77%	0.814	15.38%	0.822	61.54%	0.900	23.08%
ionosphere	0.762	52.94%	0.867	5.88%	0.886	2.94%	0.850	44.12%	0.867	47.06%	0.914	17.65%	0.933	47.06%	0.971	17.65%
kr-vs-kp	0.953	61.11%	0.953	38.89%	0.943	11.11%	0.983	44.44%	0.993	75.00%	0.998	11.11%	0.996	47.22%	0.994	55.56%
leukemia	0.952	49.24%	1.000	42.64%	1.000	0.07%	0.842	18.59%	0.524	0.01%	1.000	54.61%	0.952	50.85%	0.952	43.32%
lymphography	0.795	38.89%	0.886	55.56%	0.818	44.44%	0.972	66.67%	1.000	66.67%	0.900	38.89%	0.864	44.44%	0.818	27.78%
sonar	0.661	40.00%	0.952	30.00%	0.839	6.67%	0.850	56.67%	0.952	66.67%	1.000	16.67%	0.758	55.00%	0.855	10.00%
spect	0.862	50.00%	0.763	4.55%	0.775	4.55%	0.594	31.82%	0.888	31.82%	1.000	4.55%	0.900	54.55%	0.913	27.27%
tic-tac-toe	0.749	55.56%	0.728	33.33%	0.718	33.33%	0.909	33.33%	0.986	100.00%	0.995	11.11%	0.913	44.44%	0.955	100.00%
waveform	0.815	47.50%	0.843	42.50%	0.827	25.00%	0.857	47.50%	0.888	65.00%	0.826	75.00%	0.751	67.50%	0.783	52.50%
wine	0.981	76.92%	1.000	38.46%	1.000	38.46%	0.981	61.54%	1.000	23.08%	1.000	7.69%	0.925	53.85%	0.962	23.08%
yeast	0.461	50.00%	0.452	62.50%	0.490	75.00%	0.578	100.00%	0.593	87.50%	0.562	50.00%	0.553	50.00%	0.548	75.00%
zoo	0.833	62.50%	0.933	37.50%	0.900	50.00%	0.900	87.50%	1.000	25.00%	0.950	25.00%	0.933	50.00%	0.967	25.00%
Success over accuracy and selection size	0		2		5		4		0		13		0		7	3

Table 4.30. Comparison of the best algorithms based on F-Score

Classifier Dataset Name	Naive Bayes				Support Vector Machine				K-Nearest Neighbor				Random Forest			
	Proposed Method (JS)		Best of F-score (GA)		Proposed Method (JS)		Best of F-score (MPA)		Proposed Method (JS)		Best of F-score (SFO) and Best of SS (SFO)		Proposed Method (JS) and Best of F-score (JS)		Best of SS (SFO)	
	F-score	SS	F-score	SS	F-score	SS	F-score	SS	F-score	SS	F-score	SS	F-score	SS	F-score	SS
breast	0.919	60.00%	0.954	40.00%	0.909	66.67%	0.769	3.33%	0.955	56.67%	0.947	10.00%	0.946	33.33%	0.877	40.00%
breastcancer	0.974	66.67%	0.954	22.22%	0.957	44.44%	0.937	33.33%	0.972	66.67%	0.978	22.22%	0.930	55.56%	0.788	22.22%
car	0.883	66.67%	0.861	16.67%	0.837	0.874	0.931	100.00%	0.973	66.67%	0.954	83.33%	0.992	50.00%	0.769	83.33%
clean1	0.807	43.98%	0.916	37.95%	0.800	0.865	0.977	27.71%	0.943	81.93%	1.000	34.34%	0.800	46.99%	0.829	36.14%
clean2	0.911	49.40%	0.934	44.58%	0.872	0.942	0.956	1.81%	0.979	52.41%	1.000	36.14%	1.000	53.01%	0.852	31.93%
colon	0.700	48.80%	0.625	39.05%	0.839	44.30%	1.000	0.20%	0.556	49.75%	0.615	62.65%	0.988	48.75%	0.818	63.35%
credit-g	0.801	55.00%	0.875	45.00%	0.939	90.00%	0.860	40.00%	0.803	50.00%	0.745	5.00%	0.919	80.00%	0.839	5.00%
dermatology	1.000	47.06%	1.000	23.53%	0.750	1.000	0.984	35.29%	1.000	58.82%	0.903	26.47%	0.840	41.18%	0.857	58.82%
ecoli	0.977	71.43%	0.871	57.14%	0.563	42.86%	0.978	57.14%	1.000	57.14%	0.985	14.29%	0.901	85.71%	0.845	28.57%
glass	0.357	44.44%	0.483	22.22%	0.975	44.44%	0.821	66.67%	0.840	44.44%	0.744	22.22%	0.973	66.67%	0.812	88.89%
heart-c	0.926	61.54%	0.870	38.46%	0.750	46.15%	0.872	30.77%	0.745	53.85%	0.814	15.38%	0.917	61.54%	0.788	100.00%
ionosphere	0.468	52.94%	0.750	5.88%	0.563	44.12%	0.972	32.35%	0.774	47.06%	0.914	17.65%	1.000	47.06%	0.866	26.47%
kr-vs-kp	0.920	61.11%	0.952	38.89%	0.975	44.44%	0.992	75.00%	0.912	41.67%	0.998	11.11%	0.765	47.22%	0.811	2.78%
leukemia	0.952	49.24%	1.000	42.64%	0.917	18.59%	0.667	0.01%	0.880	50.09%	1.000	54.61%	0.667	50.85%	0.817	50.96%
lymphography	0.880	38.89%	0.852	55.56%	1.000	44.44%	1.000	66.67%	0.945	50.00%	0.900	38.89%	0.874	44.44%	0.818	72.22%
sonar	0.667	40.00%	0.949	30.00%	1.000	56.67%	0.945	66.67%	0.844	70.00%	1.000	16.67%	0.850	55.00%	0.861	3.33%
spect	0.143	50.00%	0.852	4.55%	0.632	31.82%	0.727	31.82%	0.500	81.82%	1.000	4.55%	0.966	54.55%	0.889	4.55%
tic-tac-toe	0.522	55.56%	0.562	33.33%	0.909	33.33%	0.981	100.00%	0.990	44.44%	0.995	11.11%	0.704	44.44%	0.800	11.11%
waveform	0.876	47.50%	0.876	42.50%	0.916	47.50%	0.933	65.00%	0.885	62.50%	0.826	75.00%	1.000	67.50%	0.833	27.50%
wine	0.971	76.92%	1.000	38.46%	0.971	61.54%	1.000	23.08%	1.000	53.85%	1.000	7.69%	0.852	53.85%	0.778	7.69%
yeast	0.692	50.00%	0.672	62.50%	0.746	100.00%	0.700	87.50%	0.683	87.50%	0.562	50.00%	0.983	50.00%	0.857	25.00%
zoo	1.000	62.50%	1.000	37.50%	1.000	87.50%	1.000	25.00%	1.000	50.00%	0.950	25.00%	0.857	50.00%	0.806	25.00%
Success over F-Score and selection size	1		5		0		4		2		10		6		4	

Table 4.31. Comparison of the best algorithms based on recall

Classifier Dataset Name	Naive Bayes						Support Vector Machine						K-Nearest Neighbor						Random Forest											
	Proposed Method (JS)			Best of SS (EPO)			Proposed Method (JS)			Best of Recall (MPA)			Proposed Method (JS)			Best of SS (SFO)			Proposed Method (JS)			Best of Recall (SFO)			Proposed Method (JS)			Best of Recall (SFO)		
	Recall	SS		Recall	SS		Recall	SS		Recall	SS		Recall	SS		Recall	SS		Recall	SS		Recall	SS		Recall	SS		Recall	SS	
breast	0.936	60.00%		0.963	40.00%	0.933	26.67%	0.931	66.67%	0.931	3.33%	0.629	33.33%	0.844	30.00%	0.962	56.67%	0.936	36.67%	0.947	10.00%	0.960	33.33%	1.000	40.00%	0.960	20.00%	0.960	20.00%	
breastcancer	0.973	66.67%		0.969	22.22%	0.952	44.44%	0.974	44.44%	0.918	33.33%	0.918	11.11%	0.906	11.11%	0.958	66.67%	0.972	44.44%	0.978	22.22%	0.930	55.56%	0.813	22.22%	0.922	55.56%			
car	0.790	66.67%		0.755	16.67%	0.776	16.67%	0.824	83.33%	0.908	100.00%	0.813	83.33%	0.813	83.33%	0.973	66.67%	0.965	83.33%	0.954	83.33%	1.000	50.00%	0.781	83.33%	1.000	83.33%			
clean1	0.902	43.98%		0.950	37.95%	0.893	14.46%	0.727	47.59%	0.955	27.71%	0.844	16.87%	0.984	81.93%	0.973	81.93%	1.000	39.76%	1.000	34.34%	0.778	46.99%	0.906	36.14%	0.928	12.65%			
clean2	0.950	49.40%		0.930	44.58%	0.910	3.01%	0.911	1.81%	0.919	1.81%	0.906	28.92%	0.970	52.41%	0.970	52.41%	0.984	50.60%	1.000	36.14%	1.000	53.01%	0.813	31.93%	0.987	37.35%			
colon	0.667	48.80%		1.000	39.05%	0.933	0.05%	1.000	44.30%	1.000	0.20%	0.844	24.40%	0.844	24.40%	0.417	49.75%	1.000	41.00%	0.615	62.65%	0.976	48.75%	0.844	63.35%	1.000	2.50%			
credit-g	0.823	55.00%		0.853	45.00%	0.789	40.00%	0.971	90.00%	0.799	40.00%	0.813	5.00%	0.813	5.00%	0.731	50.00%	0.821	45.00%	0.745	5.00%	0.944	80.00%	0.813	5.00%	0.818	55.00%			
dermatology	1.000	47.06%		1.000	23.53%	1.000	23.53%	0.750	35.29%	0.969	35.29%	0.906	47.06%	0.906	47.06%	1.000	58.82%	1.000	29.41%	0.903	26.47%	0.833	41.18%	0.844	58.82%	1.000	23.53%			
ecoli	0.955	71.43%		0.787	57.14%	0.822	42.86%	0.667	42.86%	0.957	57.14%	0.719	14.29%	0.719	14.29%	1.000	57.14%	1.000	85.71%	0.985	14.29%	0.941	85.71%	0.938	28.57%	0.953	71.43%			
glass	1.000	44.44%		0.700	22.22%	0.471	11.11%	1.000	44.44%	0.842	66.67%	0.875	33.33%	0.875	33.33%	0.778	44.44%	0.739	33.33%	0.744	22.22%	0.981	66.67%	0.875	88.89%	0.800	55.56%			
heart-c	0.925	61.54%		0.800	38.46%	0.949	46.15%	0.750	46.15%	0.854	30.77%	0.750	15.38%	0.750	15.38%	0.745	53.85%	0.884	38.46%	0.814	17.65%	0.909	61.54%	0.813	100.00%	0.887	53.85%			
ionosphere	1.000	52.94%		1.000	5.88%	0.935	2.94%	0.667	44.12%	0.972	32.35%	0.781	23.53%	0.781	23.53%	1.000	47.06%	1.000	20.59%	0.914	17.65%	1.000	47.06%	0.906	26.47%	1.000	32.35%			
kr-vs-kp	0.924	61.11%		0.957	38.89%	0.956	11.11%	1.000	44.44%	0.991	75.00%	0.813	16.67%	0.813	16.67%	0.987	41.67%	0.993	47.22%	0.998	11.11%	0.818	47.22%	0.938	2.78%	0.995	77.78%			
leukemia	0.909	49.24%		1.000	42.64%	1.000	0.07%	0.904	18.59%	0.500	0.01%	0.781	17.34%	0.781	17.34%	0.786	50.09%	1.000	42.95%	1.000	54.61%	0.615	50.85%	0.906	50.98%	1.000	0.52%			
lymphography	-	38.89%		0.800	55.56%	0.872	44.44%	1.000	66.67%	1.000	66.67%	0.688	16.67%	0.688	16.67%	1.000	50.00%	0.926	33.33%	0.900	38.89%	0.879	44.44%	0.844	72.22%	1.000	16.67%			
sonar	0.537	40.00%		0.903	30.00%	0.920	6.67%	1.000	56.67%	1.000	66.67%	0.969	5.00%	0.969	5.00%	0.957	70.00%	1.000	26.67%	1.000	16.67%	0.863	55.00%	0.969	3.33%	0.846	23.33%			
spect	1.000	50.00%		0.885	4.55%	0.869	4.55%	0.909	31.82%	0.800	31.82%	0.781	4.55%	0.781	4.55%	0.667	81.82%	1.000	36.36%	1.000	4.55%	1.000	54.55%	0.875	4.55%	0.947	36.36%			
tic-tac-toe	0.661	55.56%		0.694	33.33%	0.761	33.33%	0.833	33.33%	1.000	100.00%	0.781	11.11%	0.781	11.11%	1.000	44.44%	1.000	100.00%	0.995	11.11%	0.701	44.44%	0.813	11.11%	0.973	100.00%			
waveform	0.987	47.50%		0.946	42.50%	0.918	25.00%	0.914	47.50%	0.933	65.00%	0.875	65.00%	0.875	65.00%	0.897	62.50%	0.913	52.50%	0.826	75.00%	1.000	67.50%	0.938	27.50%	0.862	32.50%			
wine	1.000	76.92%		1.000	38.46%	1.000	38.46%	1.000	61.54%	1.000	23.08%	0.938	7.69%	0.938	7.69%	1.000	53.85%	0.941	38.46%	1.000	7.69%	0.859	53.85%	0.875	7.69%	1.000	23.08%			
yeast	0.642	50.00%		0.652	62.50%	0.601	75.00%	0.693	100.00%	0.585	87.50%	0.781	62.50%	0.781	62.50%	0.702	87.50%	0.651	100.00%	0.562	50.00%	0.978	50.00%	0.844	25.00%	0.681	62.50%			
zoo	1.000	62.50%		1.000	37.50%	1.000	50.00%	1.000	87.50%	1.000	25.00%	0.781	31.25%	0.781	31.25%	1.000	50.00%	1.000	43.75%	0.950	25.00%	0.917	50.00%	0.781	25.00%	1.000	31.25%			
Success over recall and selection size	0		3		5			0		4		1		1		3		7		1		1		1		5				

Table 4.32. Comparison of the best algorithms based on precision

Classifier	Naive Bayes				Support Vector Machine				K-Nearest Neighbor				Random Forest			
	Proposed Method (JS)		Best of Precision (EPO) and Best of SS (EPO)		Proposed Method (JS)		Best of Precision (SSA)		Best of SS (SFO)		Proposed Method (JS)		Best of SS (SFO)		Proposed Method (JS) and Best of Precision (JS)	
	Precision	SS	Precision	SS	Precision	SS	Precision	SS	Precision	SS	Precision	SS	Precision	SS	Precision	SS
breast	0.953	60.00%	0.951	26.67%	0.907	66.67%	1.000	13.33%	0.844	30.00%	0.963	56.67%	1.000	33.33%	0.951	33.33%
breastcancer	0.987	66.67%	1.000	44.44%	0.951	44.44%	1.000	33.33%	0.829	11.11%	0.986	66.67%	0.986	55.56%	0.944	55.56%
car	1.000	66.67%	1.000	16.67%	0.925	83.33%	0.960	83.33%	0.813	83.33%	1.000	66.67%	0.980	83.33%	1.000	50.00%
clean1	0.744	43.98%	0.838	14.46%	1.000	47.59%	0.969	42.17%	0.771	16.87%	0.917	81.93%	0.938	37.35%	0.842	46.99%
clean2	0.889	49.40%	0.975	3.01%	0.837	1.81%	1.000	32.53%	0.763	28.92%	0.993	52.41%	0.994	56.02%	1.000	53.01%
colon	0.778	48.80%	1.000	0.05%	0.794	44.30%	1.000	45.35%	0.771	24.40%	0.833	49.75%	1.000	2.25%	1.000	48.75%
credit-g	0.854	55.00%	0.921	40.00%	0.968	90.00%	0.915	35.00%	0.839	5.00%	0.920	50.00%	0.909	40.00%	1.000	80.00%
dermatology	1.000	47.06%	1.000	23.53%	0.786	35.29%	1.000	55.88%	0.784	47.06%	1.000	58.82%	1.000	26.47%	0.896	41.18%
ecoli	1.000	71.43%	1.000	42.86%	0.529	42.86%	1.000	42.86%	0.852	14.29%	1.000	57.14%	1.000	57.14%	0.865	85.71%
glass	0.286	44.44%	0.889	11.11%	0.950	44.44%	0.636	44.44%	0.757	33.33%	0.917	44.44%	0.643	33.33%	0.975	66.67%
heart-c	0.907	61.54%	0.787	46.15%	0.786	46.15%	0.926	30.77%	0.889	15.38%	0.765	53.85%	0.898	46.15%	1.000	61.54%
ionosphere	0.306	52.94%	0.744	2.94%	0.529	44.12%	0.931	32.35%	0.862	23.53%	0.649	47.06%	0.943	11.76%	1.000	47.06%
kr-vs-kp	0.916	61.11%	0.920	11.11%	0.950	44.44%	0.967	61.11%	0.765	16.67%	0.847	41.67%	0.948	58.33%	0.839	47.22%
leukemia	1.000	49.24%	1.000	0.07%	1.000	18.59%	1.000	47.75%	0.833	17.34%	1.000	50.09%	1.000	2.31%	0.818	50.85%
lymphography	-	38.89%	0.874	44.44%	1.000	66.67%	1.000	83.33%	0.846	16.67%	0.929	50.00%	0.952	27.78%	0.870	44.44%
sonar	0.920	40.00%	0.742	6.67%	1.000	56.67%	0.813	50.00%	0.775	5.00%	0.771	70.00%	0.813	25.00%	0.853	55.00%
spect	0.077	50.00%	0.803	4.55%	0.500	31.82%	0.733	36.36%	0.833	4.55%	0.400	81.82%	0.696	50.00%	0.933	54.55%
tic-tac-toe	0.458	55.56%	0.440	33.33%	1.000	33.33%	0.967	100.00%	0.833	11.11%	0.979	44.44%	0.991	100.00%	1.000	44.44%
waveform	0.810	47.50%	0.843	25.00%	0.918	47.50%	0.944	67.50%	0.875	65.00%	0.897	62.50%	0.898	62.50%	1.000	67.50%
wine	0.944	76.92%	1.000	38.46%	1.000	61.54%	1.000	38.46%	0.769	7.69%	1.000	53.85%	1.000	30.77%	0.932	53.85%
yeast	0.774	50.00%	0.793	75.00%	0.857	100.00%	0.879	87.50%	0.862	62.50%	0.742	87.50%	0.705	75.00%	0.988	50.00%
zoo	1.000	62.50%	1.000	50.00%	1.000	87.50%	1.000	43.75%	0.806	31.25%	1.000	50.00%	1.000	31.25%	0.857	50.00%
Success over precision and selection size	0	0	16	0	0	0	2	1	1	1	1	1	6	7	8	0

In this section of the thesis, the proposed algorithm, the best of the evaluation results and the best of the selection size results are compared with the evaluation score and selection size.

Tables 4.29, 4.30, 4.31 and 4.32 shows the results of 4 different evaluation metrics. The tables also provides the success of the algorithms by adding up the information of how many datasets that the algorithm has the best score and how many datasets that the algorithm less features.

We can conclude from Table 4.29 that EPO selects less features and has greater score of accuracy in 5 datasets which makes them the best algorithm to use naïve bayes for feature selection applications. Even though JS outperforms the other algorithms in 6 datasets, the performance of the other algorithms are superior to JS. Table 4.29 showed that MPA has greater scores and selects less features in 4 datasets using SVM. The table showed that SFO selects less features but that is not the only requirement to be a successful feature selector. It can be concluded from Table 4.29 that SFO is the algorithm that has the best accuracy scores and less features using KNN. The algorithm succeeded on 13 datasets meeting the both criteria of feature selection. Lastly, it can be seen from Table 4.29 that GA is the best algorithm with 7 datasets using random forest classifier. We can conclude from Table 4.29 that JS algorithm cannot perform well about selection size problem. Even though it has comparable accuracy results, other algorithms outperforms JS about selection size.

Table 4.30 showed that EPO is the better algorithm for feature selection process with naïve bayes using F-Score as evaluation metric. The algorithm succeeded in 8 datasets by choosing the less features and having the highest F-Scores. We can conclude from Table 4.30 that MPA is the best algorithm for feature selection process with SVM. Even though JS outperforms other methods by F-Score in 8 out of 22 datasets, selection size of the algorithm was quite high compared to others. For the KNN classifier, SFO presented the highest F-Scores and lowest selection sizes. SFO outperformed JS by meeting the criterias of feature selection on 10 datasets. It is seen from the Table 4.30 that JS is the superior algorithm for feature selection process, outperforming other algorithms, on 6 datasets. JS outperforms other algorithms on 16 datasets based on F-Score using the RF classifier.

It can be concluded from Table 4.31 that EPO outperforms other algorithms by meeting the requirements of feature selection process on 5 datasets while selecting the least features on 17 datasets using recall scores. JS shows good performance over 11 datasets but selection size of the algorithm is high enough to not making the algorithm the best one. We can see from Table 4.31 that MPA is the superior algorithm for feature selection using SVM as classifier. MPA outperforms other algorithms on 14 out of 22 datasets on recall score while SFO outperforms

other algorithms selecting less features on 15 out of 22 datasets. However, recall score of SFO is not high enough to make the algorithm the best for this criteria. JS shows great performance by outperforming other algorithms on 12 out of 22 datasets with greater recall scores, selection size of the algorithm is worse than its competitors with SVM classifier. Table 4.30 also showed that SFO is the best algorithm for the feature selection process using KNN as the classifier. It shows great performance for selection size by selecting the less feature on 17 out of 22 datasets.

Lastly, Table 4.31 showed that MPA is the best algorithm for feature selection process using RF to get better recall scores. JS performs well for recall scores and SFO performs well for selection size but MPA performs well for both of this evaluation metrics which makes it the best fit for the process.

We can conclude from Table 4.32 that EPO is the best algorithm for both precision score and selection size. It outperforms other algorithms heavily on 16 out of 22 algorithms using naïve bayes as classifier. JS is tied up with EPO for some cases but did not have lower selection size. We can see from Table 4.32 that SSA shows a great performance for precision score and SFO shows a great performance for selection size. Both had outstanding results but when it comes the feature selection criteria, SSA becomes the best algorithm for feature selection process using SVM as classifier. Table 4.32 showed that SFO is the best algorithm using KNN by beating the MPA and JS about precision score. Lastly, JS shows an outstanding performance by giving the highest precision score on all of the datasets and outperforming the other algorithms. The algorithm also provides great results on 8 of the datasets about selection size which making the algorithm is the best one for the feature selection process using RF.

5. CONCLUSION

In this thesis, a nature inspired metaheuristic algorithm for feature selection process is proposed. JS is used as a feature selector for the first time in the literature. Its performance compared with other algorithms including EPO, MPA, SSA, JS, SFO, WOA, ACO, GA, PSO and CHI2. We used 5 different evaluation metrics such as accuracy, precision, recall, F-Score and selection size. We tested the algorithm on 22 benchmark datasets collected from UCI. At the end of the study, we collected 3960 different result. The main purpose of this thesis is testing the performance of the algorithms that are inspired by sea creatures and bring a new algorithm to

the literature for feature selection problem. We categorized the algorithms into three sections: sea creatures, popular algorithms and traditional method. First, we compared JS within sea creature algorithms. Then, we compared the algorithm with popular and traditional methods. We compared the method using each metric one by one. We selected the best performing algorithms for the final comparison. For the final comparison, we wanted to select algorithms that meet the criterias of features selection process: minimum feature size and maximum evaluation score. We bring together the results of the best performing algorithms.

We concluded from the test results that EPO is the best algorithm for the feature selection process using naïve bayes as classifier. EPO showed outstanding results amongs others with the naïve bayes classifier that for each of the evaluation metric, it became the best algorithm amongs others. Strongest aspect of EPO was its selection size. The algorithm outperformed any other algorithm about selection size when naïve bayes used as classifier.

We concluded from the test results that MPA is the best algorithm for the feature selection process using SVM as classifier. MPA became the best algorithm on 3 out of 4 cases of the test cases. SSA outperformed MPA in one case: precision score with SVM. The strongest property of MPA was the performance score when combined with SVM. Even though the algorithm did not provide the best selection size, it was able to become the best algorithm with the highest scores.

We concluded from the test results that SFO is the best algorithm for the feature selection process using KNN as classifier. SFO outperformed other algorithms by being the best algorithm for accuracy, precision, recall, F-Score and selection size metrics. SFO showed an outstanding performance about accuracy and F-Score by giving the highest scores nearly on all of the datasets. Different from other results, the strong side of SFO was not just scores or selection size. The algorithm became the best for the scores and selection sizes. Table 4.29 showed that SFO became the first algorithm for selection size on 18 out of 22 datasets.

We concluded from the test results that JS is the best algorithm for the feature selection process using RF as a classifier. Even though JS compared with proven algorithms, it became the best algorithm for two cases of this thesis. Using random forest classifier enhanced the performance of JS optimizer algorithm. JS became the best method for F-Score and Precision scores. Most particularly, JS outperformed any other algorithm on precision score by giving the highest scores on all of the datasets with RF classifier. For the other cases, GA and MPA became the

best algorithm for accuracy scores and recall scores, respectively. The results for the JS showed that JS better classified the datasets with the low number of false positives than other methods. The performance of the high precision scores reflected on the performance of F-Measure.

For the future work of this thesis, execution time comparison may be added as an evaluation metric since all of the optimizers take long to run. We were not able to add to this thesis because we tested the cases with different computers. It would not be fair or ethical to compare the execution time. Some other traditional feature selection algorithms might be added to the work to compare the performance of the proposed algorithm. Algorithms may be tested on different types of datasets other than numerical benchmarks. Population and iteration parameters might be varied to observe the performance of the feature selector better. Solution vectors can be recorded in order not to calculate (classify) the same vector for the second time. We concluded from the test cases that the reason behind the poor performance of JS is low ratio of selection size. JS algorithm can be hybridize with another algorithm to enhance the performance of exploration phase of the algorithm.

REFERENCES

- Abd Elaziz, M., Ewees, A. A., Yousri, D., Abualigah, L., & Al-qaness, M. A. A. (2022). Modified marine predators algorithm for feature selection: case study metabolomics. *Knowledge and Information Systems*, 64(1), 261–287. <https://doi.org/10.1007/s10115-021-01641-w>
- Abdel-Basset, M., El-Shahat, D., Chakraborty, R. K., & Ryan, M. (2021). Parameter estimation of photovoltaic models using an improved marine predators algorithm. *Energy Conversion and Management*, 227, 113491. <https://doi.org/10.1016/j.enconman.2020.113491>
- Abdel-Basset, M., Mohamed, R., Chakraborty, R. K., Ryan, M. J., & El-Fergany, A. (2021). An Improved Artificial Jellyfish Search Optimizer for Parameter Identification of Photovoltaic Models. *Energies*, 14(7), 1867. <https://doi.org/10.3390/en14071867>
- Abedinia, O., Amjady, N., & Ghasemi, A. (2014). A new metaheuristic algorithm based on shark smell optimization. *Complexity*, 21(5), 97–116. Portico. <https://doi.org/10.1002/cplx.21634>
- Abouelmagd, L. M., Shams, M. Y., El-Attar, N. E., & Hassanien, A. E. (2021). Feature Selection Based Coral Reefs Optimization for Breast Cancer Classification. *Medical Informatics and Bioimaging Using Artificial Intelligence*, 53–72. https://doi.org/10.1007/978-3-030-91103-4_4
- Abualigah, L., Shehab, M., Alshinwan, M., & Alabool, H. (2019). Salp swarm algorithm: a comprehensive survey. *Neural Computing and Applications*, 32(15), 11195–11215. <https://doi.org/10.1007/s00521-019-04629-4>
- Aghdam, M. H., Ghasem-Aghaee, N., & Basiri, M. E. (2009). Text feature selection using ant colony optimization. *Expert Systems with Applications*, 36(3), 6843–6853. <https://doi.org/10.1016/j.eswa.2008.08.022>
- Akinola, O. O., Ezugwu, A. E., Agushaka, J. O., Zitar, R. A., & Abualigah, L. (2022). Multiclass feature selection with metaheuristic optimization algorithms: a review. *Neural Computing and Applications*, 34(22), 19751–19790. <https://doi.org/10.1007/s00521-022-07705-4>

Alam, A., Verma, P., Tariq, M., Sarwar, A., Alamri, B., Zahra, N., & Urooj, S. (2021). Jellyfish Search Optimization Algorithm for MPP Tracking of PV System. *Sustainability*, 13(21), 11736. <https://doi.org/10.3390/su132111736>

Aljarah, I., Al-Zoubi, A. M., Faris, H., Hassonah, M. A., Mirjalili, S., & Saadeh, H. (2018). Simultaneous Feature Selection and Support Vector Machine Optimization Using the Grasshopper Optimization Algorithm. *Cognitive Computation*, 10(3), 478–495. <https://doi.org/10.1007/s12559-017-9542-9>

Al-qaness, M. A. A., Ewees, A. A., Fan, H., Abualigah, L., & Abd Elaziz, M. (2020). Marine Predators Algorithm for Forecasting Confirmed Cases of COVID-19 in Italy, USA, Iran and Korea. *International Journal of Environmental Research and Public Health*, 17(10), 3520. <https://doi.org/10.3390/ijerph17103520>

Al-Tashi, Q., Abdul Kadir, S. J., Rais, H. M., Mirjalili, S., & Alhussian, H. (2019). Binary Optimization Using Hybrid Grey Wolf Optimization for Feature Selection. *IEEE Access*, 7, 39496–39508. <https://doi.org/10.1109/access.2019.2906757>

Alweshah, M., Hammouri, A., Alkhalaileh, S., & Alzubi, O. (2022). Intrusion detection for the internet of things (IoT) based on the emperor penguin colony optimization algorithm. *Journal of Ambient Intelligence and Humanized Computing*, 14(5), 6349–6366. <https://doi.org/10.1007/s12652-022-04407-6>

Applications of Support Vector Machine (SVM) Learning in Cancer Genomics. (2018). *Cancer Genomics & Proteomics*, 15(1). <https://doi.org/10.21873/cgp.20063>

Babatunde, O. H., Armstrong, L., Leng, J., & Diepeveen, D. (2014). A genetic algorithm-based feature selection.

Balakrishnan, K., Dhanalakshmi, R., & Khaire, U. M. (2022). A novel control factor and Brownian motion-based improved Harris Hawks Optimization for feature selection. *Journal of Ambient Intelligence and Humanized Computing*, 14(7), 8631–8653. <https://doi.org/10.1007/s12652-021-03621-y>

Bastian, T., Lilley, M. K. S., Beggs, S. E., Hays, G. C., & Doyle, T. K. (2014). Ecosystem relevance of variable jellyfish biomass in the Irish Sea between years, regions and water

types. *Estuarine, Coastal and Shelf Science*, 149, 302–312.

<https://doi.org/10.1016/j.ecss.2014.08.018>

Chou, J.-S., & Molla, A. (2022). Recent advances in use of bio-inspired jellyfish search algorithm for solving optimization problems. *Scientific Reports*, 12(1).

<https://doi.org/10.1038/s41598-022-23121-z>

Chou, J.-S., & Truong, D.-N. (2020). Multiobjective optimization inspired by behavior of jellyfish for solving structural design problems. *Chaos, Solitons & Fractals*, 135, 109738.

<https://doi.org/10.1016/j.chaos.2020.109738>

Chou, J.-S., & Truong, D.-N. (2021). A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean. *Applied Mathematics and Computation*, 389, 125535.

<https://doi.org/10.1016/j.amc.2020.125535>

Coloni, A., Dorigo, M., & Maniezzo, V. (1991, December). Distributed optimization by ant colonies. In *Proceedings of the first European conference on artificial life* (Vol. 142, pp. 134–142).

Dasgupta, A., Banerjee, A., Dastidar, A. G., Barman, A., & Chakraborty, S. (2019). A Study and Analysis of a Feature Subset Selection Technique Using Penguin Search Optimization Algorithm. *Data Science*, 51–68. <https://doi.org/10.1201/9780429263798-3>

Dasgupta, D., & Michalewicz, Z. (Eds.). (2013). *Evolutionary algorithms in engineering applications*. Springer Science & Business Media.

Deng, W., Xu, J., & Zhao, H. (2019). An Improved Ant Colony Optimization Algorithm Based on Hybrid Strategies for Scheduling Problem. *IEEE Access*, 7, 20281–20292.

<https://doi.org/10.1109/access.2019.2897580>

Dhiman, G., & Kumar, V. (2019). Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems. *Knowledge-Based Systems*, 165, 169–196.

<https://doi.org/10.1016/j.knosys.2018.11.024>

Dhiman, G., Oliva, D., Kaur, A., Singh, K. K., Vimal, S., Sharma, A., & Cengiz, K. (2021). BEPO: A novel binary emperor penguin optimizer for automatic feature selection.

Knowledge-Based Systems, 211, 106560. <https://doi.org/10.1016/j.knosys.2020.106560>

Dorigo, M. (1992). Optimization, learning and natural algorithms. Ph. D. Thesis, Politecnico di Milano.

Dorigo, M., Maniezzo, V., & Coloni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE transactions on systems, man, and cybernetics, part b (cybernetics)*, 26(1), 29-41.

Du, H., Wu, X., & Zhuang, J. (2006). Small-World Optimization Algorithm for Function Optimization. *Advances in Natural Computation*, 264–273.
https://doi.org/10.1007/11881223_33

Eberhart, R., & Kennedy, J. (n.d.). A new optimizer using particle swarm theory. MHS'95. *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*.
<https://doi.org/10.1109/mhs.1995.494215>

Elminaam, D. S. A., Nabil, A., Ibraheem, S. A., & Houssein, E. H. (2021). An Efficient Marine Predators Algorithm for Feature Selection. *IEEE Access*, 9, 60136–60153.
<https://doi.org/10.1109/access.2021.3073261>

Faramarzi, A., Heidarinejad, M., Mirjalili, S., & Gandomi, A. H. (2020). Marine Predators Algorithm: A nature-inspired metaheuristic. *Expert Systems with Applications*, 152, 113377.
<https://doi.org/10.1016/j.eswa.2020.113377>

Farhat, M., Kamel, S., Atallah, A. M., & Khan, B. (2021). Optimal Power Flow Solution Based on Jellyfish Search Optimization Considering Uncertainty of Renewable Energy Sources. *IEEE Access*, 9, 100911–100933. <https://doi.org/10.1109/access.2021.3097006>

Filho, C. J. A. B., de Lima Neto, F. B., Lins, A. J. C. C., Nascimento, A. I. S., & Lima, M. P. (2009). Fish School Search. *Studies in Computational Intelligence*, 261–277.
https://doi.org/10.1007/978-3-642-00267-0_9

Forsati, R., Moayedikia, A., Keikha, A., & Shamsfard, M. (2012). A Novel Approach for Feature Selection based on the Bee Colony Optimization. *International Journal of Computer Applications*, 43(8), 13–16. <https://doi.org/10.5120/6122-8329>

Gen, M., & Cheng, R. (1999). Genetic algorithms and engineering optimization (Vol. 7). John Wiley & Sons.

- Gheraibia, Y., & Moussaoui, A. (2013). Penguins Search Optimization Algorithm (PeSOA). *Lecture Notes in Computer Science*, 222–231. https://doi.org/10.1007/978-3-642-38577-3_23
- Ghojogh, B., & Sharifian, S. (2018). Pontogammarus Maeoticus Swarm Optimization: A Metaheuristic Optimization Algorithm. *arXiv preprint arXiv:1807.01844*.
- Ghosh, K. K., Ahmed, S., Singh, P. K., Geem, Z. W., & Sarkar, R. (2020). Improved Binary Sailfish Optimizer Based on Adaptive β -Hill Climbing for Feature Selection. *IEEE Access*, 8, 83548–83560. <https://doi.org/10.1109/access.2020.2991543>
- Ghosh, M., Guha, R., Mondal, R., Singh, P. K., Sarkar, R., & Nasipuri, M. (2018). Feature Selection Using Histogram-Based Multi-objective GA for Handwritten Devanagari Numeral Recognition. *Intelligent Engineering Informatics*, 471–479. https://doi.org/10.1007/978-981-10-7566-7_46
- Ghosh, M., Guha, R., Sarkar, R., & Abraham, A. (2019). A wrapper-filter feature selection technique based on ant colony optimization. *Neural Computing and Applications*, 32(12), 7839–7857. <https://doi.org/10.1007/s00521-019-04171-3>
- Glover, F., Laguna, M., & Taillard, E. (1993). *Tabu search*. Basel, Switzerland.
- Hammouti, I. E., Lajjam, A., Merouani, M. E., & Tabaa, Y. (2019). A modified sailfish optimizer to solve dynamic berth allocation problem in conventional container terminal. *International Journal of Industrial Engineering Computations*, 491–504. <https://doi.org/10.5267/j.ijiec.2019.4.002>
- Harifi, S., Khalilian, M., Mohammadzadeh, J., & Ebrahimnejad, S. (2019). Emperor Penguins Colony: a new metaheuristic algorithm for optimization. *Evolutionary Intelligence*, 12(2), 211–226. <https://doi.org/10.1007/s12065-019-00212-x>
- Hatamlou, A. (2013). Black hole: A new heuristic optimization approach for data clustering. *Information Sciences*, 222, 175–184. <https://doi.org/10.1016/j.ins.2012.08.023>
- Hof, P. R., & Van Der Gucht, E. (2007). Structure of the cerebral cortex of the humpback whale, *Megaptera novaeangliae* (Cetacea, Mysticeti, Balaenopteridae). *The Anatomical Record: Advances in Integrative Anatomy and Evolutionary Biology*, 290(1), 1–31. <https://doi.org/10.1002/ar.20407>

Holland, J.H. (1975) *Adaptation in Natural and Artificial Systems*, The University of Michigan Press.

Hussein, W. A., Sahran, S., & Sheikh Abdullah, S. N. H. (2014). Patch-Levy-based initialization algorithm for Bees Algorithm. *Applied Soft Computing*, 23, 104–121. <https://doi.org/10.1016/j.asoc.2014.06.004>

Ibrahim, R. A., Ewees, A. A., Oliva, D., Abd Elaziz, M., & Lu, S. (2018). Improved salp swarm algorithm based on particle swarm optimization for feature selection. *Journal of Ambient Intelligence and Humanized Computing*, 10(8), 3155–3169. <https://doi.org/10.1007/s12652-018-1031-9>

Ibrahim, R. A., Ewees, A. A., Oliva, D., Abd Elaziz, M., & Lu, S. (2018). Improved salp swarm algorithm based on particle swarm optimization for feature selection. *Journal of Ambient Intelligence and Humanized Computing*, 10(8), 3155–3169. <https://doi.org/10.1007/s12652-018-1031-9>

Ikram, R. M. A., Dehrashid, A. A., Zhang, B., Chen, Z., Le, B. N., & Moayedi, H. (2023). A novel swarm intelligence: cuckoo optimization algorithm (COA) and SailFish optimizer (SFO) in landslide susceptibility assessment. *Stochastic Environmental Research and Risk Assessment*, 37(5), 1717–1743. <https://doi.org/10.1007/s00477-022-02361-5>

Improved marine predators algorithm for feature selection and SVM optimization. (2022). *KSII Transactions on Internet and Information Systems*, 16(4). <https://doi.org/10.3837/tiis.2022.04.003>

J.R. Koza, “Genetic programming,” 1992.

Jahani, E., & Chizari, M. (2018). Tackling global optimization problems with a novel algorithm – Mouth Brooding Fish algorithm. *Applied Soft Computing*, 62, 987–1002. <https://doi.org/10.1016/j.asoc.2017.09.035>

Jia, H., Peng, X., & Lang, C. (2021). Remora optimization algorithm. *Expert Systems with Applications*, 185, 115665. <https://doi.org/10.1016/j.eswa.2021.115665>

Jia, H., Xing, Z., & Song, W. (2019). A New Hybrid Seagull Optimization Algorithm for Feature Selection. *IEEE Access*, 7, 49614–49631.

<https://doi.org/10.1109/access.2019.2909945>

Jianhao, W., Long, W., Lijie, C., & Tian, G. (2021). Enhanced Whale Optimization Algorithm for Large-Scale Global Optimization Problems. 2021 International Conference on Computer Communication and Artificial Intelligence (CCAI).

<https://doi.org/10.1109/ccai50917.2021.9447541>

Johnson, J. M., & Rahmat-Samii, Y. (1994, June). Genetic algorithm optimization and its application to antenna design. In *Proceedings of IEEE Antennas and Propagation Society International Symposium and URSI National Radio Science Meeting* (Vol. 1, pp. 326-329). IEEE.

Jovanovic, L., Zivkovic, M., Antonijevic, M., Jovanovic, D., Ivanovic, M., & Jassim, H. S. (2022). An emperor penguin optimizer application for medical diagnostics. 2022 IEEE Zooming Innovation in Consumer Technologies Conference (ZINC).

<https://doi.org/10.1109/zinc55034.2022.9840612>

Jovic, A., Brkic, K., & Bogunovic, N. (2015). A review of feature selection methods with applications. 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO).

<https://doi.org/10.1109/mipro.2015.7160458>

Kaçabeti, A. (2021). A new hybrid metaheuristic algorithm applied on a classification problem (Master's thesis, İstanbul Sabahattin Zaim Üniversitesi).

Kalayci, G. (2019). Application of Metaheuristics for the Feature Selection Problem (Doctoral dissertation, Marmara Üniversitesi (Turkey)).

Kalra, M., Kumar, V., Kaur, M., Ahmed Idris, S., Öztürk, Ş., & Alshazly, H. (2022). A Novel Binary Emperor Penguin Optimizer for Feature Selection Tasks. *Computers, Materials & Continua*, 70(3), 6239–6255. <https://doi.org/10.32604/cmc.2022.020682>

Karasu, S., & Altan, A. (2019). Recognition Model for Solar Radiation Time Series based on Random Forest with Feature Selection Approach. 2019 11th International Conference on

Electrical and Electronics Engineering (ELECO).
<https://doi.org/10.23919/eleco47770.2019.8990664>

Karimi, F., Dowlatshahi, M. B., & Hashemi, A. (2023). SemiACO: A semi-supervised feature selection based on ant colony optimization. *Expert Systems with Applications*, 214, 119130.
<https://doi.org/10.1016/j.eswa.2022.119130>

Kashan, A. H. (2009). League Championship Algorithm: A New Algorithm for Numerical Function Optimization. 2009 International Conference of Soft Computing and Pattern Recognition. <https://doi.org/10.1109/socpar.2009.21>

Kaveh, A., & Farhoudi, N. (2013). A new optimization method: Dolphin echolocation. *Advances in Engineering Software*, 59, 53–70.
<https://doi.org/10.1016/j.advengsoft.2013.03.004>

Kaveh, A., Biabani Hamedani, K., Kamalinejad, M., & Joudaki, A. (2021). Quantum-based jellyfish search optimizer for structural optimization. *Int J Optim Civil Eng*, 11(2), 329-56.

Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks* (Vol. 4, pp. 1942-1948). IEEE.

Kirkpatrick, S., Gelatt Jr, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598), 671-680.

Kumar, B. S., Santhi, S. G., & Narayana, S. (2021). Sailfish optimizer algorithm (SFO) for optimized clustering in wireless sensor network (WSN). *Journal of Engineering, Design and Technology*, 20(6), 1449–1467. <https://doi.org/10.1108/jedt-02-2021-0087>

Kumar, V., Kumar, D., Kaur, M., Singh, D., Idris, S. A., & Alshazly, H. (2021). A Novel Binary Seagull Optimizer and its Application to Feature Selection Problem. *IEEE Access*, 9, 103481–103496. <https://doi.org/10.1109/access.2021.3098642>

Le Maho, Y. (1977). The emperor penguin: a strategy to live and breed in the cold: morphology, physiology, ecology, and behavior distinguish the polar emperor penguin from other penguin species, particularly from its close relative, the king penguin. *American Scientist*, 65(6), 680-693.

- Liang, H., Zou, J., Zuo, K., & Khan, M. J. (2020). An improved genetic algorithm optimization fuzzy controller applied to the wellhead back pressure control system. *Mechanical Systems and Signal Processing*, 142, 106708. <https://doi.org/10.1016/j.ymssp.2020.106708>
- Lin, S.-W., Ying, K.-C., Chen, S.-C., & Lee, Z.-J. (2008). Particle swarm optimization for parameter determination and feature selection of support vector machines. *Expert Systems with Applications*, 35(4), 1817–1824. <https://doi.org/10.1016/j.eswa.2007.08.088>
- Liu, W., Wang, Z., Yuan, Y., Zeng, N., Hone, K., & Liu, X. (2021). A Novel Sigmoid-Function-Based Adaptive Weighted Particle Swarm Optimizer. *IEEE Transactions on Cybernetics*, 51(2), 1085–1093. <https://doi.org/10.1109/tcyb.2019.2925015>
- Liu, X., Wang, N., Molina, D., & Herrera, F. (2022). A least square support vector machine approach based on bvRNA-GA for modeling photovoltaic systems. *Applied Soft Computing*, 117, 108357. <https://doi.org/10.1016/j.asoc.2021.108357>
- Liu, Y., Wang, G., Chen, H., Dong, H., Zhu, X., & Wang, S. (2011). An improved particle swarm optimization for feature selection. *Journal of Bionic Engineering*, 8(2), 191–200. [https://doi.org/10.1016/s1672-6529\(11\)60020-6](https://doi.org/10.1016/s1672-6529(11)60020-6)
- Lutvica, K., & Konjicija, S. (2017). Alternative pheromone laying strategy — An improvement for the ACO algorithm. 2017 XXVI International Conference on Information, Communication and Automation Technologies (ICAT). <https://doi.org/10.1109/icat.2017.8171607>
- Mafarja, M. M., & Mirjalili, S. (2017). Hybrid Whale Optimization Algorithm with simulated annealing for feature selection. *Neurocomputing*, 260, 302–312. <https://doi.org/10.1016/j.neucom.2017.04.053>
- Mafarja, M., & Mirjalili, S. (2018). Whale optimization approaches for wrapper feature selection. *Applied Soft Computing*, 62, 441–453. <https://doi.org/10.1016/j.asoc.2017.11.006>
- Mahdavi, M., Fesanghary, M., & Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation*, 188(2), 1567–1579. <https://doi.org/10.1016/j.amc.2006.11.033>

- Maleki, N., Zeinali, Y., & Niaki, S. T. A. (2021). A k-NN method for lung cancer prognosis with the use of a genetic algorithm for feature selection. *Expert Systems with Applications*, 164, 113981. <https://doi.org/10.1016/j.eswa.2020.113981>
- Mansour, R. F., Alhumyani, H., Khalek, S. A., Saeed, R. A., & Gupta, D. (2022). Design of cultural emperor penguin optimizer for energy-efficient resource scheduling in green cloud computing environment. *Cluster Computing*, 26(1), 575–586. <https://doi.org/10.1007/s10586-022-03608-0>
- Masadeh, R., A., B., & Sharieh, A. (2019). Sea Lion Optimization Algorithm. *International Journal of Advanced Computer Science and Applications*, 10(5). <https://doi.org/10.14569/ijacsa.2019.0100548>
- Meier, D., Tullumi, I., Stauffer, Y., Dornberger, R., & Hanne, T. (2017). A novel backup path planning approach with ACO. 2017 5th International Symposium on Computational and Business Intelligence (ISCBI). <https://doi.org/10.1109/iscbi.2017.8053543>
- Mirjalili, S., & Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software*, 95, 51–67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
- Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, 114, 163–191. <https://doi.org/10.1016/j.advengsoft.2017.07.002>
- Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*, 69, 46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
- Mirjalili, S., Mirjalili, S. M., & Yang, X.-S. (2013). Binary bat algorithm. *Neural Computing and Applications*, 25(3–4), 663–681. <https://doi.org/10.1007/s00521-013-1525-5>
- Moghaddam, F. F., Moghaddam, R. F., & Cheriet, M. (2012). Curved space optimization: a random search based on general relativity theory. *arXiv preprint arXiv:1208.2214*.
- Mostafa Bozorgi, S., & Yazdani, S. (2019). IWOA: An improved whale optimization algorithm for optimization problems. *Journal of Computational Design and Engineering*, 6(3), 243–259. <https://doi.org/10.1016/j.jcde.2019.02.002>

Nadimi-Shahraki, M. H., Zamani, H., & Mirjalili, S. (2022). Enhanced whale optimization algorithm for medical feature selection: A COVID-19 case study. *Computers in Biology and Medicine*, 148, 105858. <https://doi.org/10.1016/j.compbiomed.2022.105858>

Nasiri, J., & Khiyabani, F. M. (2018). A whale optimization algorithm (WOA) approach for clustering. *Cogent Mathematics & Statistics*, 5(1), 1483565. <https://doi.org/10.1080/25742558.2018.1483565>

Niknam, T., & Amiri, B. (2010). An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis. *Applied Soft Computing*, 10(1), 183–197. <https://doi.org/10.1016/j.asoc.2009.07.001>

Novakovic, J. (2010, November). The impact of feature selection on the accuracy of naïve bayes classifier. In *18th Telecommunications forum TELFOR* (Vol. 2, pp. 1113-1116).

Paniri, M., Dowlatshahi, M. B., & Nezamabadi-pour, H. (2020). MLACO: A multi-label feature selection algorithm based on ant colony optimization. *Knowledge-Based Systems*, 192, 105285. <https://doi.org/10.1016/j.knosys.2019.105285>

Parouha, R. P., & Das, K. N. (2016). A memory based differential evolution algorithm for unconstrained optimization. *Applied Soft Computing*, 38, 501–517. <https://doi.org/10.1016/j.asoc.2015.10.022>

Peng, H., Ying, C., Tan, S., Hu, B., & Sun, Z. (2018). An Improved Feature Selection Algorithm Based on Ant Colony Optimization. *IEEE Access*, 6, 69203–69209. <https://doi.org/10.1109/access.2018.2879583>

Pozna, C., Precup, R.-E., Horvath, E., & Petriu, E. M. (2022). Hybrid Particle Filter–Particle Swarm Optimization Algorithm and Application to Fuzzy Controlled Servo Systems. *IEEE Transactions on Fuzzy Systems*, 30(10), 4286–4297. <https://doi.org/10.1109/tfuzz.2022.3146986>

Qais, M. H., Hasanien, H. M., & Alghuwainem, S. (2019). Enhanced salp swarm algorithm: Application to variable speed wind generators. *Engineering Applications of Artificial Intelligence*, 80, 82–96. <https://doi.org/10.1016/j.engappai.2019.01.011>

Qiu, X., Zhang, L., Nagaratnam Suganthan, P., & Amaratunga, G. A. J. (2017). Oblique random forest ensemble via Least Square Estimation for time series forecasting. *Information Sciences*, 420, 249–262. <https://doi.org/10.1016/j.ins.2017.08.060>

Rai, A., & Das, D. K. (2022). The development of a fuzzy tilt integral derivative controller based on the sailfish optimizer to solve load frequency control in a microgrid, incorporating energy storage systems. *Journal of Energy Storage*, 48, 103887. <https://doi.org/10.1016/j.est.2021.103887>

Rai, R., Dhal, K. G., Das, A., & Ray, S. (2023). An Inclusive Survey on Marine Predators Algorithm: Variants and Applications. *Archives of Computational Methods in Engineering*, 30(5), 3133–3172. <https://doi.org/10.1007/s11831-023-09897-x>

Rajamoorthy, R., Arunachalam, G., Kasinathan, P., Devendiran, R., Ahmadi, P., Pandiyan, S., Muthusamy, S., Panchal, H., Kazem, H. A., & Sharma, P. (2022). A novel intelligent transport system charging scheduling for electric vehicles using Grey Wolf Optimizer and Sail Fish Optimization algorithms. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, 44(2), 3555–3575. <https://doi.org/10.1080/15567036.2022.2067268>

Ramezani, M., Bahmanyar, D., & Razmjooy, N. (2021). A New Improved Model of Marine Predator Algorithm for Optimization Problems. *Arabian Journal for Science and Engineering*, 46(9), 8803–8826. <https://doi.org/10.1007/s13369-021-05688-3>

Rani, R. R., & Ramyachitra, D. (2017). Krill Herd Optimization algorithm for cancer feature selection and random forest technique for classification. 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS). <https://doi.org/10.1109/icsess.2017.8342875>

Rashedi, E., Nezamabadi-pour, H., & Saryazdi, S. (2009). GSA: A Gravitational Search Algorithm. *Information Sciences*, 179(13), 2232–2248. <https://doi.org/10.1016/j.ins.2009.03.004>

Reddy, G. T., Reddy, M. P. K., Lakshmanan, K., Rajput, D. S., Kaluri, R., & Srivastava, G. (2019). Hybrid genetic algorithm and a fuzzy logic classifier for heart disease diagnosis. *Evolutionary Intelligence*, 13(2), 185–196. <https://doi.org/10.1007/s12065-019-00327-1>

Ren, J., Lee, S. D., Chen, X., Kao, B., Cheng, R., & Cheung, D. (2009). Naive Bayes Classification of Uncertain Data. 2009 Ninth IEEE International Conference on Data Mining. <https://doi.org/10.1109/icdm.2009.90>

Reza Naji, H., Shadravan, S., Mousa Jafarabadi, H., & Momeni, H. (2022). Accelerating sailfish optimization applied to unconstrained optimization problems on graphical processing unit. *Engineering Science and Technology, an International Journal*, 32, 101077. <https://doi.org/10.1016/j.jestch.2021.11.003>

Rizk-Allah, R. M., Hassanien, A. E., Elhoseny, M., & Gunasekaran, M. (2018). A new binary salp swarm algorithm: development and application for optimization tasks. *Neural Computing and Applications*, 31(5), 1641–1663. <https://doi.org/10.1007/s00521-018-3613-z>

Rokbani, N., Kumar, R., Abraham, A., Alimi, A. M., Long, H. V., Priyadarshini, I., & Son, L. H. (2020). Bi-heuristic ant colony optimization-based approaches for traveling salesman problem. *Soft Computing*, 25(5), 3775–3794. <https://doi.org/10.1007/s00500-020-05406-5>

Salcedo-Sanz, S., Del Ser, J., Landa-Torres, I., Gil-López, S., & Portilla-Figueras, J. A. (2014). The Coral Reefs Optimization Algorithm: A Novel Metaheuristic for Efficiently Solving Optimization Problems. *The Scientific World Journal*, 2014, 1–15. <https://doi.org/10.1155/2014/739768>

Salgotra, R., Singh, U., Singh, S., Singh, G., & Mittal, N. (2021). Self-adaptive salp swarm algorithm for engineering optimization problems. *Applied Mathematical Modelling*, 89, 188–207. <https://doi.org/10.1016/j.apm.2020.08.014>

Saremi, S., Mirjalili, S. M., & Mirjalili, S. (2014). Chaotic Krill Herd Optimization Algorithm. *Procedia Technology*, 12, 180–185. <https://doi.org/10.1016/j.protcy.2013.12.473>

Saremi, S., Mirjalili, S., & Lewis, A. (2014). How important is a transfer function in discrete heuristic algorithms. *Neural Computing and Applications*, 26(3), 625–640. <https://doi.org/10.1007/s00521-014-1743-5>

Sayed, G. I., Khoriba, G., & Haggag, M. H. (2018). A novel chaotic salp swarm algorithm for global optimization and feature selection. *Applied Intelligence*, 48(10), 3462–3481. <https://doi.org/10.1007/s10489-018-1158-6>

Sayed, S., Nassef, M., Badr, A., & Farag, I. (2019). A Nested Genetic Algorithm for feature selection in high-dimensional cancer Microarray datasets. *Expert Systems with Applications*, 121, 233–243. <https://doi.org/10.1016/j.eswa.2018.12.022>

Seyyedabbasi, A., Aliyev, R., Kiani, F., Gulle, M. U., Basyildiz, H., & Shah, M. A. (2021). Hybrid algorithms based on combining reinforcement learning and metaheuristic methods to solve global optimization problems. *Knowledge-Based Systems*, 223, 107044. <https://doi.org/10.1016/j.knosys.2021.107044>

Shadravan, S., Naji, H. R., & Bardsiri, V. K. (2019). The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. *Engineering Applications of Artificial Intelligence*, 80, 20–34. <https://doi.org/10.1016/j.engappai.2019.01.001>

Sharawi, M., Zawbaa, H. M., Emary, E., Zawbaa, H. M., & Emary, E. (2017). Feature selection approach based on whale optimization algorithm. 2017 Ninth International Conference on Advanced Computational Intelligence (ICACI). <https://doi.org/10.1109/icaci.2017.7974502>

Simon, D. (2008). Biogeography-based optimization. *IEEE transactions on evolutionary computation*, 12(6), 702–713.

Singh, L. K., Khanna, M., Garg, H., & Singh, R. (2023). Emperor penguin optimization algorithm- and bacterial foraging optimization algorithm-based novel feature selection approach for glaucoma classification from fundus images. *Soft Computing*. <https://doi.org/10.1007/s00500-023-08449-6>

Singh, N., Singh, S. B., & Houssein, E. H. (2020). Hybridizing salp swarm algorithm with particle swarm optimization algorithm for recent optimization functions. *Evolutionary Intelligence*, 15(1), 23–56. <https://doi.org/10.1007/s12065-020-00486-6>

Sivanandam, S. N., & Deepa, S. N. (2008). Genetic Algorithm Optimization Problems. *Introduction to Genetic Algorithms*, 165–209. https://doi.org/10.1007/978-3-540-73190-0_7

Soliman, M. A., Hasanien, H. M., & Alkuhayli, A. (2020). Marine Predators Algorithm for Parameters Identification of Triple-Diode Photovoltaic Models. *IEEE Access*, 8, 155832–155842. <https://doi.org/10.1109/access.2020.3019244>

Song, X.-F., Zhang, Y., Gong, D.-W., & Gao, X.-Z. (2022). A Fast Hybrid Feature Selection Based on Correlation-Guided Clustering and Particle Swarm Optimization for High-Dimensional Data. *IEEE Transactions on Cybernetics*, 52(9), 9573–9586.

<https://doi.org/10.1109/tcyb.2021.3061152>

Tabakhi, S., Moradi, P., & Akhlaghian, F. (2014). An unsupervised feature selection algorithm based on ant colony optimization. *Engineering Applications of Artificial Intelligence*, 32, 112–123. <https://doi.org/10.1016/j.engappai.2014.03.007>

Tang, F., Li, J., & Zafetti, N. (2020). Optimization of residential building envelopes using an improved Emperor Penguin Optimizer. *Engineering with Computers*, 38(2), 1395–1407.

<https://doi.org/10.1007/s00366-020-01112-w>

Tansui, D., & Thammano, A. (2017). Nature-inspired optimization method: Hydrozoan algorithm for solving continuous problems. 2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD). <https://doi.org/10.1109/snpd.2017.8022695>

Taunk, K., De, S., Verma, S., & Swetapadma, A. (2019). A Brief Review of Nearest Neighbor Algorithm for Learning and Classification. 2019 International Conference on Intelligent Computing and Control Systems (ICCS).

<https://doi.org/10.1109/iccs45141.2019.9065747>

Tawhid, M. A., & Dsouza, K. B. (2018). Hybrid binary bat enhanced particle swarm optimization algorithm for solving feature selection problems. *Applied Computing and Informatics*, 16(1/2), 117–136. <https://doi.org/10.1016/j.aci.2018.04.001>

Thippa Reddy, G., & Khare, N. (2016). FFBAT-Optimized Rule Based Fuzzy Logic Classifier for Diabetes. *International Journal of Engineering Research in Africa*, 24, 137–152. <https://doi.org/10.4028/www.scientific.net/jera.24.137>

Trafalis, T. B., & Ince, H. (2000). Support vector machine for regression and applications to financial forecasting. *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*. <https://doi.org/10.1109/ijcnn.2000.859420>

Tubishat, M., Ja'afar, S., Alswaitti, M., Mirjalili, S., Idris, N., Ismail, M. A., & Omar, M. S. (2021). Dynamic Salp swarm algorithm for feature selection. *Expert Systems with Applications*, 164, 113873. <https://doi.org/10.1016/j.eswa.2020.113873>

Turkoglu, B., Uymaz, S. A., & Kaya, E. (2022). Binary Artificial Algae Algorithm for feature selection. *Applied Soft Computing*, 120, 108630. <https://doi.org/10.1016/j.asoc.2022.108630>

Uymaz, S. A., Tezel, G., & Yel, E. (2015). Artificial algae algorithm (AAA) for nonlinear global optimization. *Applied Soft Computing*, 31, 153–171. <https://doi.org/10.1016/j.asoc.2015.03.003>

Wang, F., Zhang, H., & Zhou, A. (2021). A particle swarm optimization algorithm for mixed-variable optimization problems. *Swarm and Evolutionary Computation*, 60, 100808. <https://doi.org/10.1016/j.swevo.2020.100808>

Wang, Y., & Han, Z. (2021). Ant colony optimization for traveling salesman problem based on parameters optimization. *Applied Soft Computing*, 107, 107439. <https://doi.org/10.1016/j.asoc.2021.107439>

Wei, J., Zhang, R., Yu, Z., Hu, R., Tang, J., Gui, C., & Yuan, Y. (2017). A BPSO-SVM algorithm based on memory renewal and enhanced mutation mechanisms for feature selection. *Applied Soft Computing*, 58, 176–192. <https://doi.org/10.1016/j.asoc.2017.04.061>

Weile, D. S., & Michielssen, E. (1997). Genetic algorithm optimization applied to electromagnetics: a review. *IEEE Transactions on Antennas and Propagation*, 45(3), 343–353. <https://doi.org/10.1109/8.558650>

Wetter, M., & Wright, J. (2003, August). Comparison of a generalized pattern search and a genetic algorithm optimization method. In *Proc. of the 8-th IBPSA Conference* (Vol. 3, pp. 1401-1408).

Widodo, A., & Yang, B.-S. (2007). Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical Systems and Signal Processing*, 21(6), 2560–2574. <https://doi.org/10.1016/j.ymssp.2006.12.007>

Xue, B., Zhang, M., & Browne, W. N. (2013). Particle Swarm Optimization for Feature Selection in Classification: A Multi-Objective Approach. *IEEE Transactions on Cybernetics*, 43(6), 1656–1671. <https://doi.org/10.1109/tsmcb.2012.2227469>

Yang, X.-S., & Deb, S. (2013). Multiobjective cuckoo search for design optimization. *Computers & Operations Research*, 40(6), 1616–1624. <https://doi.org/10.1016/j.cor.2011.09.026>

Yang, X.-S., & Suash Deb. (2009). Cuckoo Search via Lévy flights. 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC). <https://doi.org/10.1109/nabic.2009.5393690>

Youssef, H., Hassan, M. H., Kamel, S., & Elsayed, S. K. (2021). Parameter Estimation of Single Phase Transformer Using Jellyfish Search Optimizer Algorithm. 2021 IEEE International Conference on Automation/XXIV Congress of the Chilean Association of Automatic Control (ICA-ACCA). <https://doi.org/10.1109/icaacca51523.2021.9465279>

Zeng, N., Wang, Z., Liu, W., Zhang, H., Hone, K., & Liu, X. (2022). A Dynamic Neighborhood-Based Switching Particle Swarm Optimization Algorithm. *IEEE Transactions on Cybernetics*, 52(9), 9290–9301. <https://doi.org/10.1109/tcyb.2020.3029748>

Zhang, X., Liu, H., & Tu, L. (2020). A modified particle swarm optimization for multimodal multi-objective optimization. *Engineering Applications of Artificial Intelligence*, 95, 103905. <https://doi.org/10.1016/j.engappai.2020.103905>

Zhang, Y., & Mo, Y. (2021). Dynamic Optimization of Chemical Processes Based on Modified Sailfish Optimizer Combined with an Equal Division Method. *Processes*, 9(10), 1806. <https://doi.org/10.3390/pr9101806>

Zheng, Y., Li, Y., Wang, G., Chen, Y., Xu, Q., Fan, J., & Cui, X. (2019). A Novel Hybrid Algorithm for Feature Selection Based on Whale Optimization Algorithm. *IEEE Access*, 7, 14908–14923. <https://doi.org/10.1109/access.2018.2879848>

Zhou, X., Ma, H., Gu, J., Chen, H., & Deng, W. (2022). Parameter adaptation-based ant colony optimization with dynamic hybrid mechanism. *Engineering Applications of Artificial Intelligence*, 114, 105139. <https://doi.org/10.1016/j.engappai.2022.105139>

Zivkovic, M., Stoean, C., Chhabra, A., Budimirovic, N., Petrovic, A., & Bacanin, N. (2022). Novel Improved Salp Swarm Algorithm: An Application for Feature Selection. *Sensors*, 22(5), 1711. <https://doi.org/10.3390/s22051711>

Zivkovic, M., Stoean, C., Chhabra, A., Budimirovic, N., Petrovic, A., & Bacanin, N. (2022). Novel Improved Salp Swarm Algorithm: An Application for Feature Selection. *Sensors*, 22(5), 1711. <https://doi.org/10.3390/s22051711>

Zorarpacı, E., & Özel, S. A. (2016). A hybrid approach of differential evolution and artificial bee colony for feature selection. *Expert Systems with Applications*, 62, 91–103. <https://doi.org/10.1016/j.eswa.2016.06.004>

