Prospect of Quantum Inspired Algorithms for Optimum Feature Subset Selection in Machine Learning

Ashis kumar Mandal and Basabi Chakraborty

Presenter: Basabi Chakraborty

Iwate Prefectural University, Japan

Madanapalle Institute of Technology and Science, AP, India

basabi@iwate-pu.ac.jp, drbasabic@mits.ac.in

About Me

- Received B. Tech, M. Tech and Ph. D degrees in Radio Physics and Electronics from Calcutta University
- Worked in Indian Statistical Institute, Kolkata as Computer Engineer
- Visiting Researcher in AIC (Advanced Intelligent Communication) Systems in Sendai, Japan (1991-1993)
- Doctoral and Post doctoral research In RIEC (Research Institute of Electrical Communication) , Tohoku University, Japan (1993-1998)
- Ph. D in Information Science (1996)
- Faculty in Dept. of Software and Information Science ,Iwate Prefectural university (1998-2022)
- Visiting faculty in Dept. of Electrical and Computer Engineering, University of Western Ontario, Canada (Oct. 2006 –March 2007)
- Professor Emeritus and Distinguished Professor in Iwate Prefectural University (April, 2022 ~)
- Dean, School of Computing , Madanapalle Institute of Science and Technology

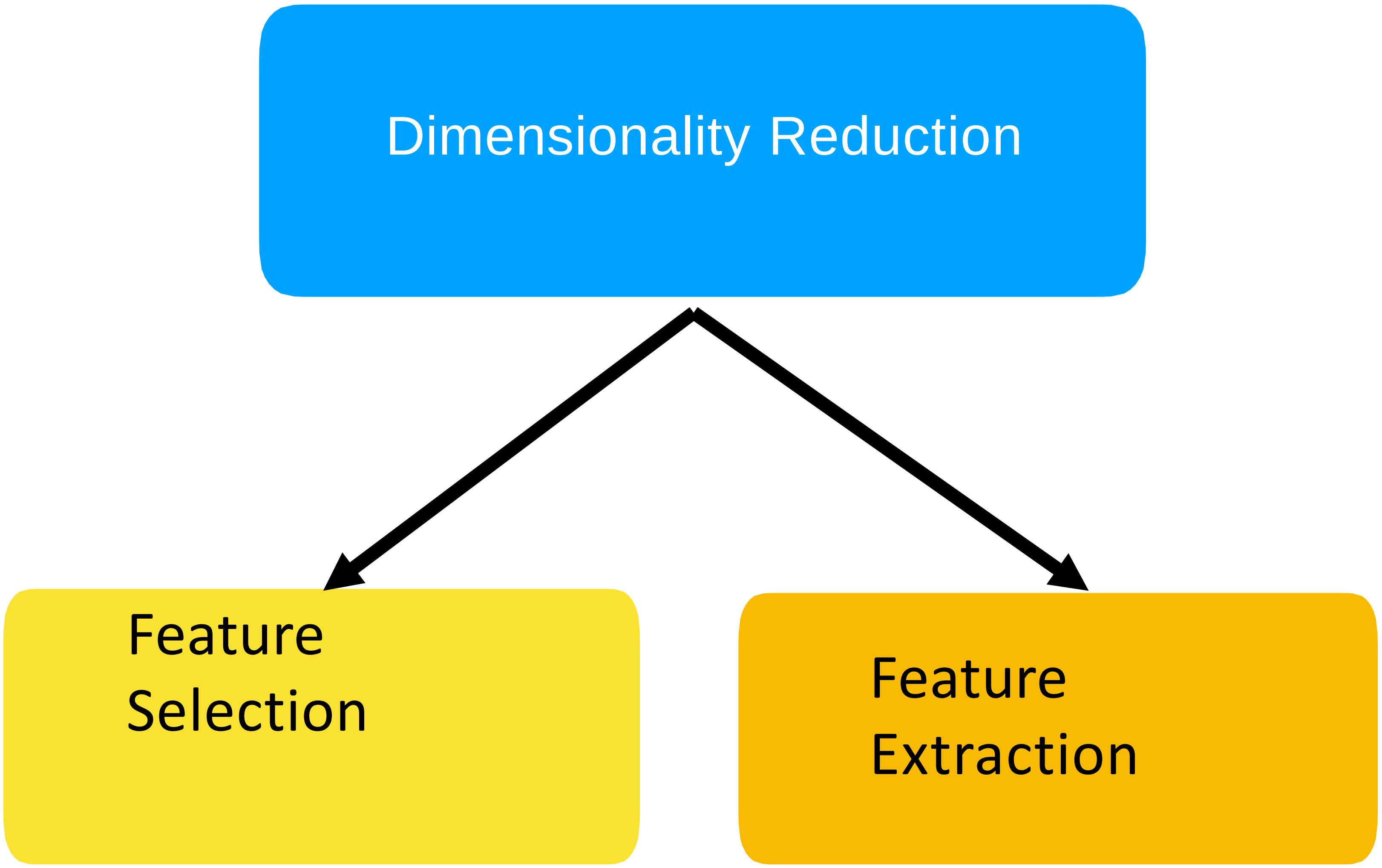
About My Research

- Development of Computing Techniques for solving Cognitive problems.
- Tools: Mathematics, Statistics, ANN, Fuzzy logic, Rough set theory, GA, EC,PSO etc. and their hybrids
- Problems: Pattern Recognition and Machine Learning : handwriting recognition, face and face expression recognition, signature verification, gait recognition, activity recognition, person authentication etc..
- Feature Evaluation and Selection
- Time series data analysis, classification, clustering, prediction
- Data mining, Bioinformatics
- Online Social Data mining
- Quantum Machine learning

Why do we need Feature Subset Selection?

- Pre processing step in Pattern Recognition , Data mining problems

- Dimensionality Reduction, Feature Extraction or Generation and Feature Selection/Subset Selection



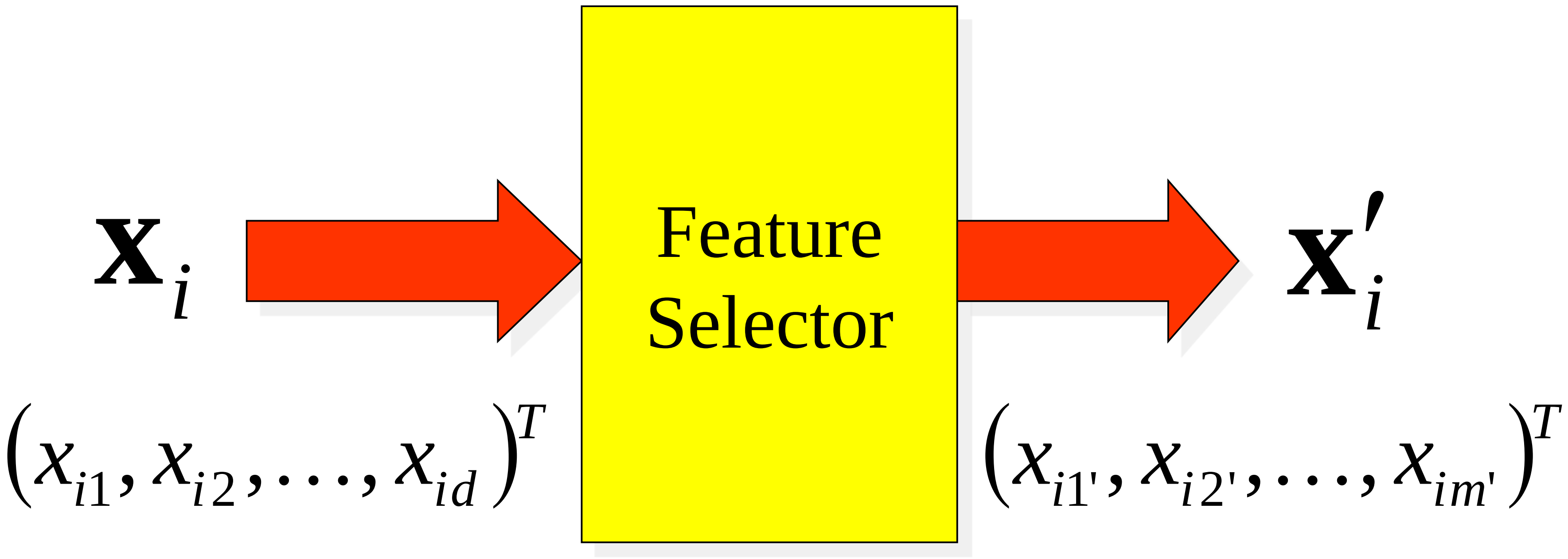
Dimensionality Reduction

Feature Selection

Feature Extraction

# possible Selections $\binom{d}{m}$

Feature Selector



$\mathbf{x}_i$

Feature Selector

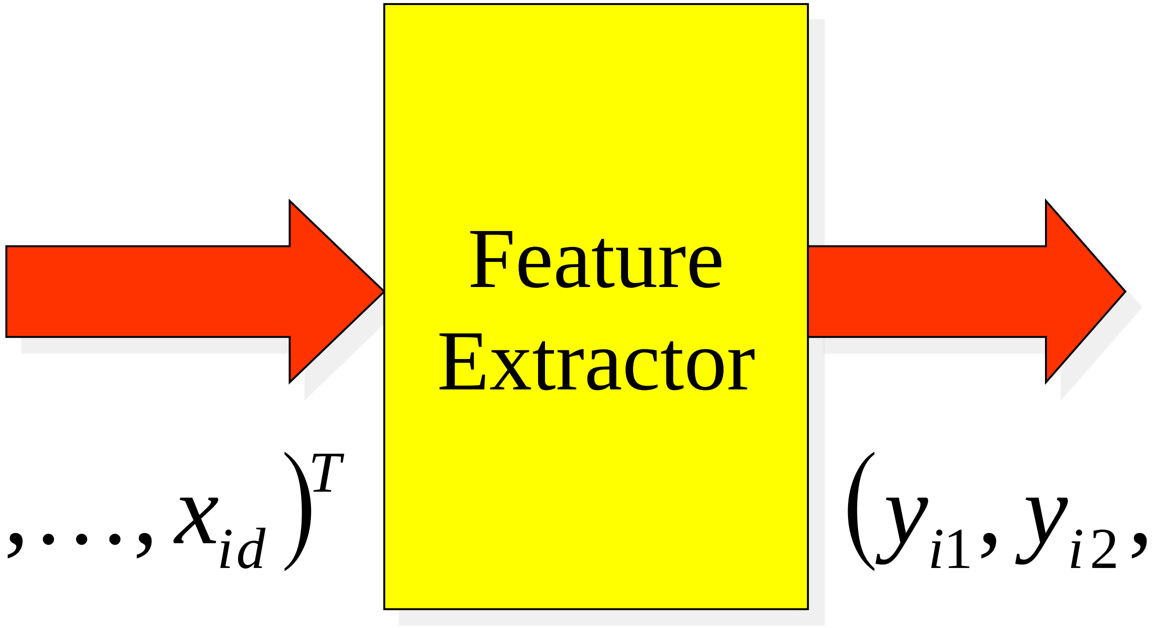
$\mathbf{x}_i'$

$(x_{i1}, x_{i2}, \ldots, x_{id})^T$

$(x_{i1'}, x_{i2'}, \ldots, x_{im'})^T$

$m \leq d$, usually

Feature Extractor/Generator



$\mathbf{x}_i$

Feature Extractor

$\mathbf{y}_i$

$(x_{i1}, x_{i2}, \ldots, x_{id})^T$

$(y_{i1}, y_{i2}, \ldots, y_{im})^T$

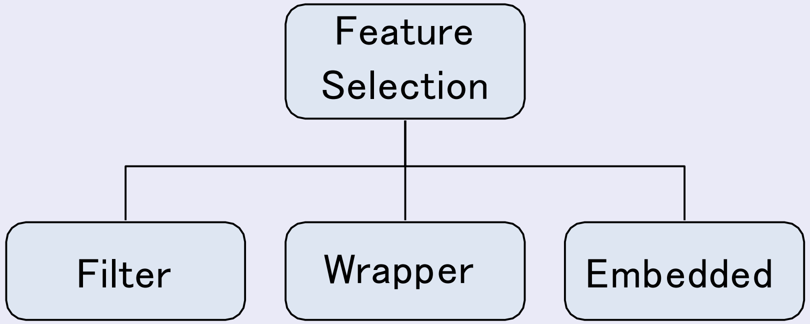
$m \leq d$, usually

Feature Subset Selection

- Given a set of *N* features, select the optimized subset of *m* features that leads to the best performance of the classifier
- Two Important tasks
- Feature Evaluation by a suitable metric
- Optimum feature subset selection by proper search strategy

Feature Subset Selection Approaches

Taxonomy of Feature Selection: evaluation



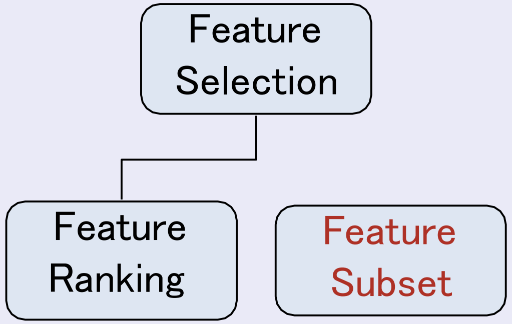
Feature Selection

Filter

Wrapper

Embedded

Taxonomy of Feature Selection: selection type



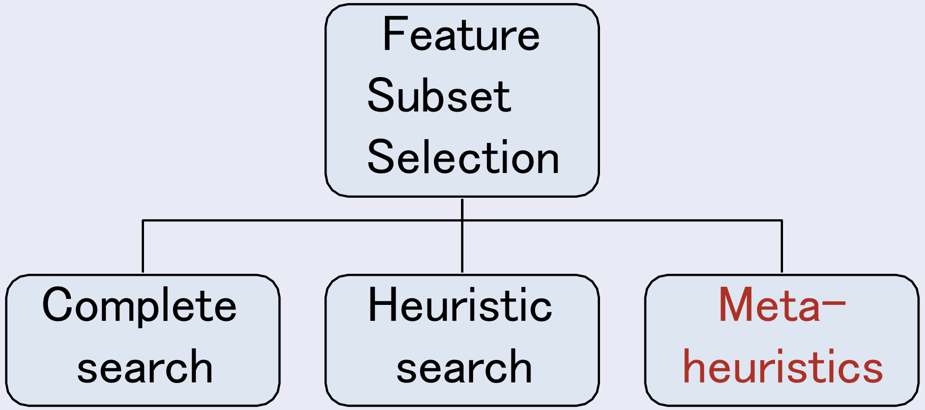
Feature Selection

Feature Ranking

Feature Subset

- Feature Subset Selection
  - Finding best subset is NP-hard problem
  - Combinatorial optimization problem
  - For n feature, total Possible number of subset is $2^n - 1$

Search Strategy for Feature subset



Feature Subset Selection

Complete search

Heuristic search

Meta–heuristics

Objective

Key Objective

- Development of effective feature subset selection algorithms using
  - Meta heuristics
  - Quantum computing and quantum inspired strategies

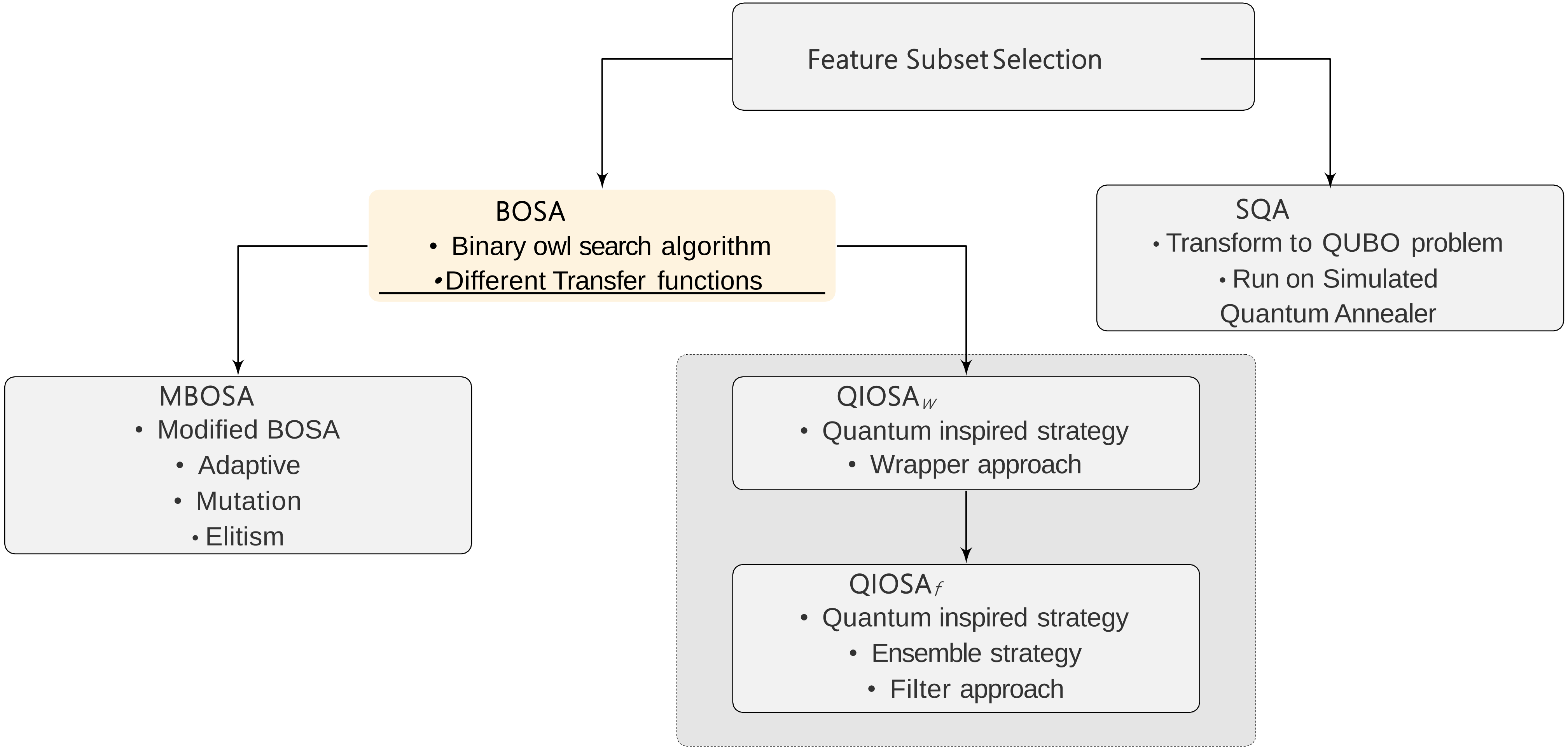
How to measure quality of a feature subset algorithm

- Classification accuracy
- Number of features
- Stability
- Computational time

Datasets:

- Numerical data
- Number of features: 4 to 22,283

Our Approach



Feature Subset Selection

BOSA
- Binary owl search algorithm
- Different Transfer functions

SQA
- Transform to QUBO problem
- Run on Simulated Quantum Annealer

MBOSA
- Modified BOSA
- Adaptive
- Mutation
- Elitism

$QIOSA_W$
- Quantum inspired strategy
- Wrapper approach

$QIOSA_f$
- Quantum inspired strategy
- Ensemble strategy
- Filter approach

Owl Search Algorithm (OSA):Introduction

1. Owl Search Algorithm (OSA)
   1. Recently proposed
      Population-based meta-heuristic
   3. Inspiration: Auditory map of prey's
   - sound generated by a owl's brain

2. Motivation
   1. Solve continuous optimization problem effectively
   2. Less parameter than other meta heuristics
   3. OSA was not used for feature subset selection

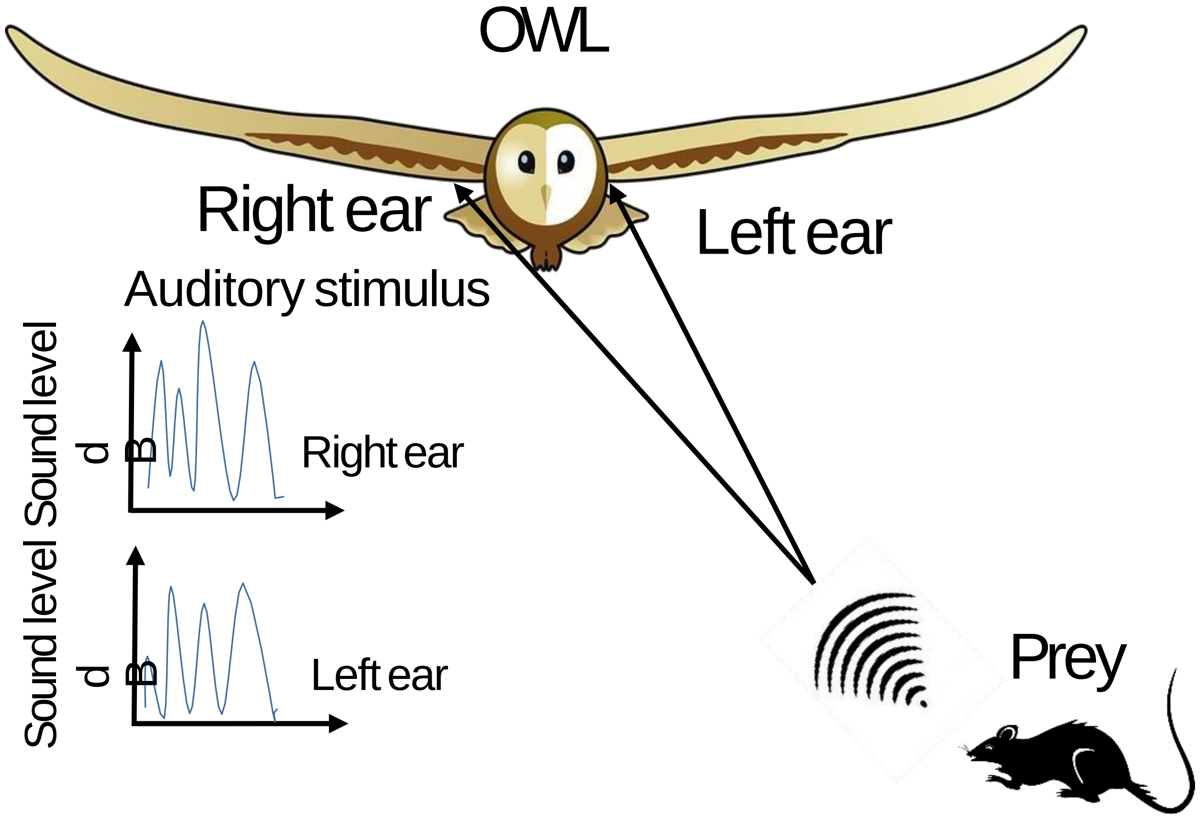


Figure 1:The distance of prey is estimated on the basis of time and intensity differences of sound wave arrival

OSA: Equation

Initialization

$$O_i = O_L + U(0,1) \times (O_U - O_L) \tag{1}$$

O_i = i^{th} owl in d dimension space, $U(0, 1)$=uniform random number between 0 and 1. O_U upper and O_L = lower bound of i^{th} owl

Fitness value of i^{th} owl

$$f_i = f([O_{i1}, O_{i2}, ..., O_{id}]) \tag{2}$$

Intensity

$$I_i = \frac{(f_i - w)}{(b - w)} \tag{3}$$

w=min intensity, b=max intensity

OSA: Equation

$$I_{ci} = \frac{I_i}{R_i^2} + randomnoise \tag{4}$$

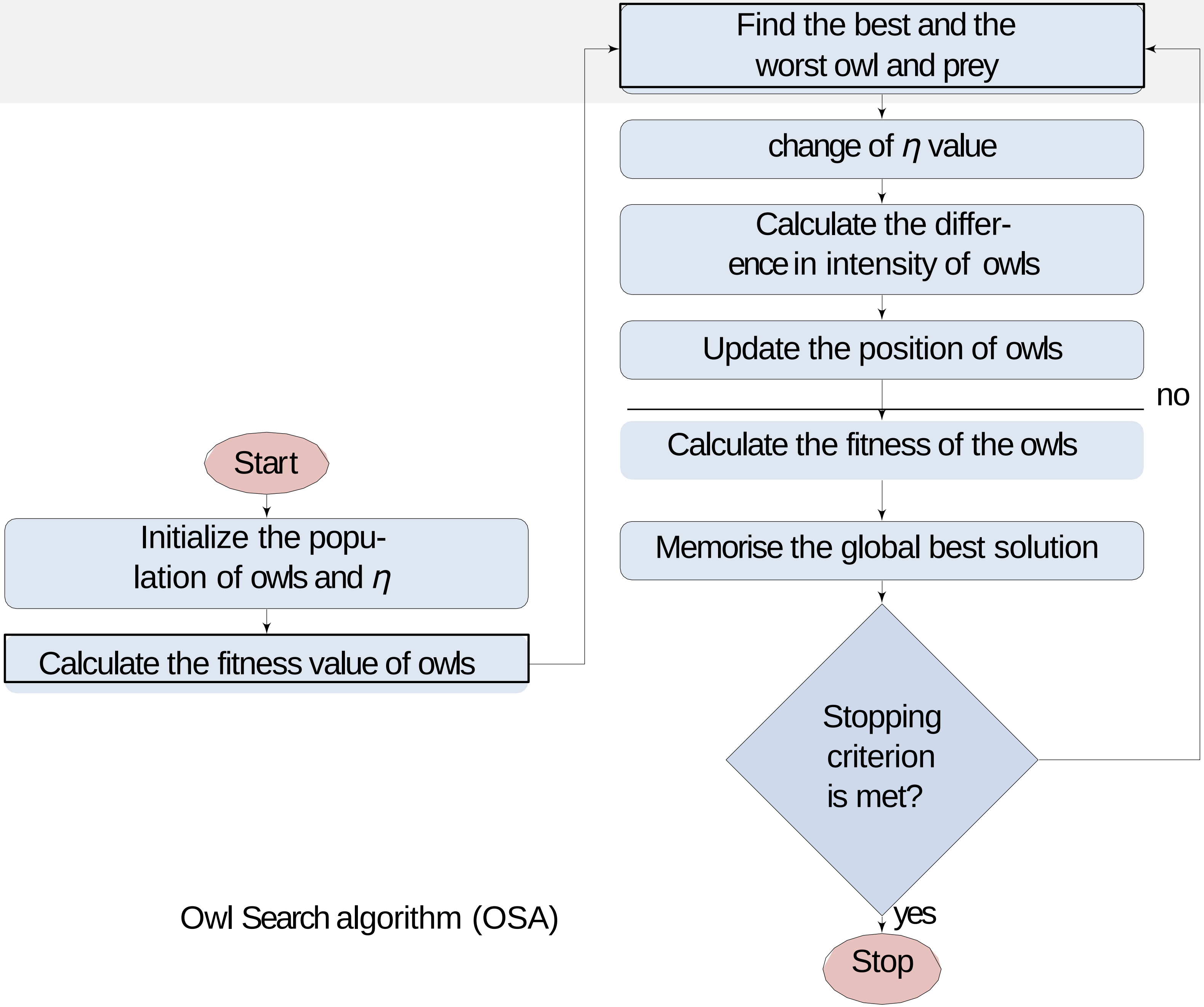
R_i^2 is distance measure

Update position

$$O_i(t+1) = \begin{cases} O_i(t) + \eta \times I_{ci} \times |\zeta V - O_i(t)|, & \text{if } p_{vm} < 0.5 \\ O_i(t - \eta \times I_{ci} \times |\zeta V - O_i(t)|, & \text{if } p_{vm} \geq 0.5 \end{cases} \tag{5}$$

V is prey (i.e. nearest owl near to prey), ζ is random number, η is user derived function, p_{vm}=Uniform random number

- η parameter: important for controlling exploration and exploitation


Start
Initialize the population of owls and η
Calculate the fitness value of owls
Find the best and the worst owl and prey
change of η value
Calculate the difference in intensity of owls
Update the position of owls
no
Calculate the fitness of the owls
Memorise the global best solution
Stopping criterion is met?
yes
Stop

Owl Search algorithm (OSA)

BOSA for feature subset selection

1. Convert OSA to binary variation of OSA (BOSA)
2. Use transfer functions for mapping

Initialization: Feature is either selected (1) or not (0)

$$O_i = U_b(0, 1) \qquad (6)$$

$U_b(0, 1)$=random number either 0 and 1

Updating position

$$\Delta O_i(t+1) = \begin{cases} O_i(t) + \eta \times I_{c\,i} \times |\zeta V - O_i(t)|, & \text{if } p_{vm} < 0.5 \\ O_i(t) - \eta \times I_{c\,i} \times |\zeta V - O_i(t)|, & \text{if } p_{vm} \geq 0.5 \end{cases} \qquad (7)$$

$\Delta O_i(t+1)$ is the step vector in continuous space of i^{th} owl at iteration $t+1$.

BOSA for feature subset selection (cont.)

Update of η parameter

$$\eta = \rho - (t \times \rho)/T \tag{8}$$

ρ is constant and it is 2.0, $T=$ total no. iteration, t =current no. iteration

Distance

$$R_i = \sum_{j=1}^{d} |O_i^j - V^j| \quad \text{where} \quad O_i^j, V^j \in \{0,1\} \tag{9}$$

R_i is hamming distance

BOSA for feature subset selection: Transfer function

1. Transfer functions (TF) convert continuous to binary space
2. 11 transfer functions are used to build models
   - 4 S-shaped (BOSA-S1 to BOSA-s4)
   - 4 V-shaped (BOSA-V1 to BOSA-V4)
   - 3 Q-shaped (BOSA-Q1 to BOSA-Q4)

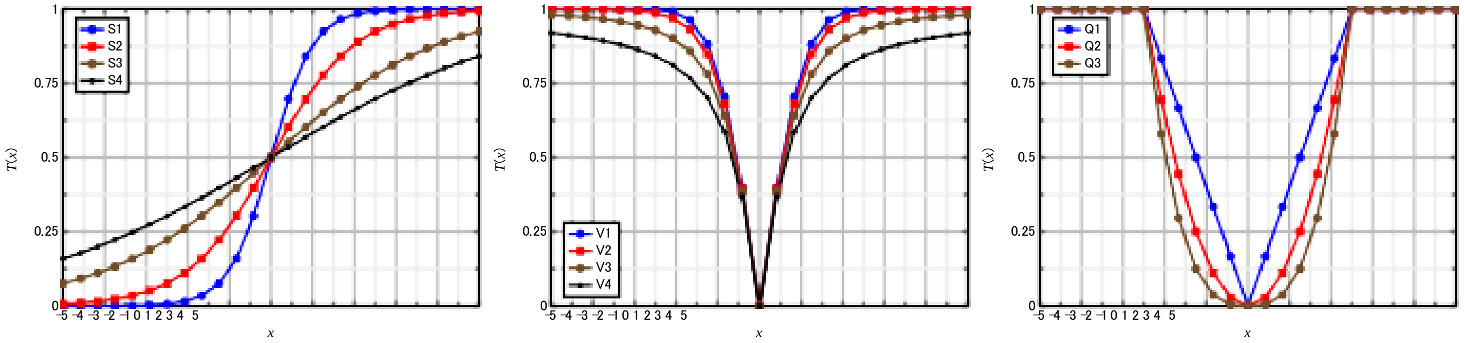


Figure 2:Shape of different transfer functions

BOSA for feature subset selection

Fitness function: classification accuracy + no. of selected

$$F = \omega \times Acc + (1 - \omega) \times (1 - \frac{F_S}{F_T}) \tag{10}$$

Acc =classifier accuracy, F_S = cardinality, F_T=no. of original feature, and ω is a weighted value (.90).

- K-nearest neighbor (KNN,k=3) for fitness function
- Support vector machine (SVM) for evaluation of the results
- 20 UCI datasets are used
- Best model is compared with Harmony Search (HS), Binary Particle Swarm Optimization(BPSO), Binary Genetic Algorithm (BGA)

Results and Discussion

Table 1:Number of best values obtained

Measure (Avg)	BOSA S1	BOSA S2	BOSA S3	BOSA S4	BOSA V1	BOSA V2	BOSA V3	BOSA V4	BOSA Q1	BOSA Q2	BOSA Q3
Fitness	0	0	2	4	0	1	0	0	0	4	15
Accuracy	4	3	1	1	3	4	0	2	3	2	5
Feature selection	0	0	1	3	0	0	0	0	0	2	16
Time	0	3	2	6	0	0	0	0	0	0	9

- BOSA-Q3 has demonstrated high success regarding fitness and feature section
- BOSA-Q3 has competitive classification accuracy

Results and Discussion (cont.)

Table 2:Average Classification Accuracy of BOSA-Q3 Vs. others

Dataset	Arrhythmia	Breast-w	Clean1	Dermatology	Hepatitis	Ilpd	Libras-move	Lung-cancer	Parkinsons	Pendigits
HS	60.053	95.854	75.944	95	78.298	**71.429**	61.944	48	85.763	**95.024**
BPSO	60.735	**96.488**	75.105	**96.296**	78.936	**71.429**	63.611	43	**86.61**	94.867
BGA	61.985	96.439	**77.762**	**96.296**	78.936	**71.429**	**64.444**	44	85.932	95.361
BOSA-Q3	**64.282**	**96.488**	76.084	95.833	**80**	**71.429**	63.704	**51**	85.932	94.873

Dataset	Promoters	Qsar-biodeg	Semeion	Sonar	Spambase	Spect	Spectf	Vehicle	Wine	Wisconsin
HS	70.625	82.965	89.854	73.492	89.768	69.877	80	72.008	**92.778**	95.263
BPSO	77.813	83.88	90.251	71.27	90.905	70.37	81.429	71.89	91.667	95.146
BGA	74.063	**84.006**	90.669	73.333	91.195	72.346	81.524	**72.913**	92.037	95.205
BOSA-Q3	**79.375**	82.776	**91.423**	**74.603**	**91.253**	**73.21**	**82.19**	72.441	92.037	**95.38**

Results and Discussion (cont.)

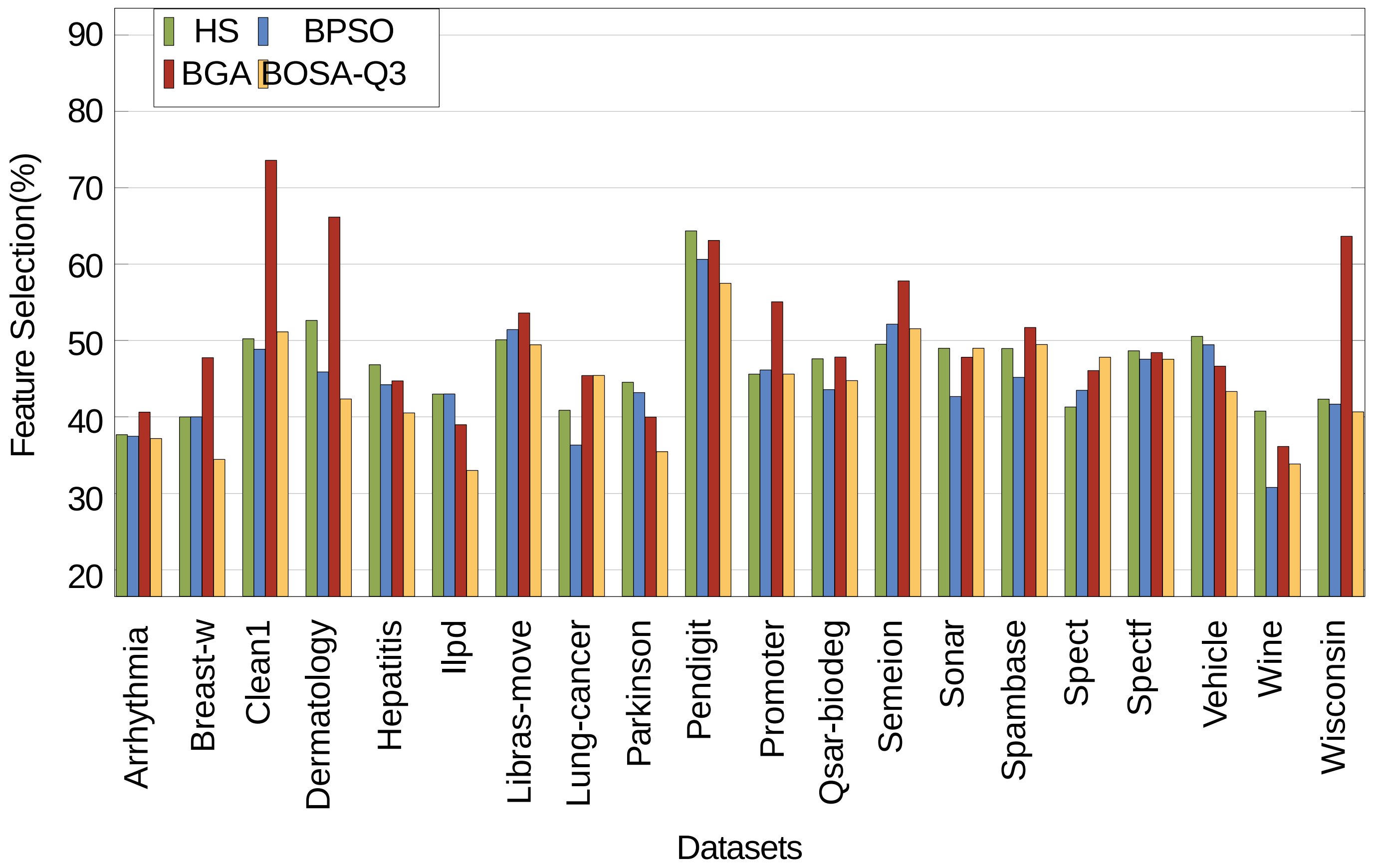


Figure 3:Average feature selection rate of BOSA-Q3 compared to other algorithms

- The less the percentage of feature selection, the better the approach is

Results and Discussion (cont.)

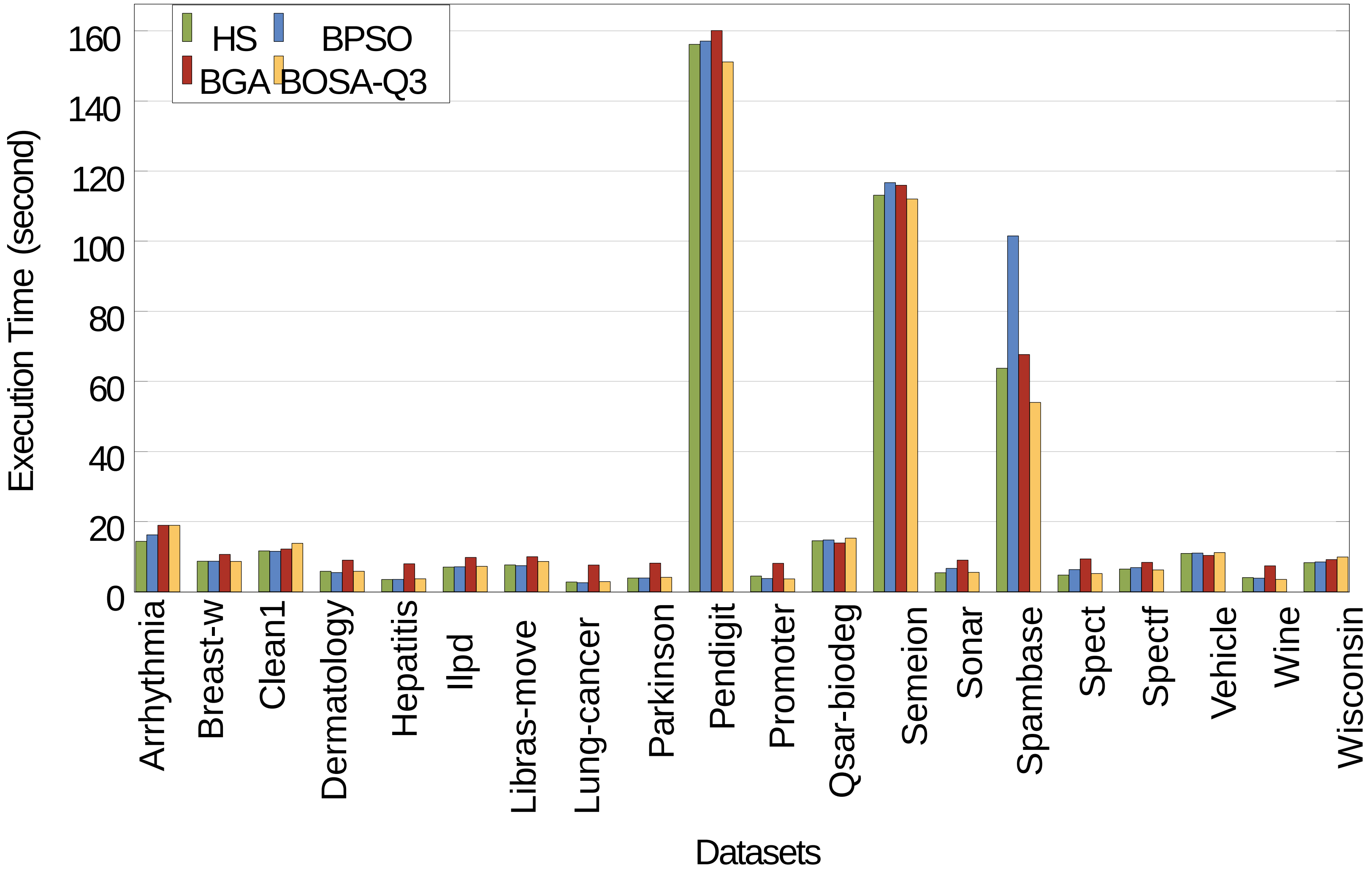


Figure 4:Average CPU time of BOSA-Q3 compared to other algorithms

Results and Discussion (cont.)

Table 3:Summary results (Avg. value)

Methods	Fitness	Accuracy	no. of Feature
HS	0.865	79.20%	44.73%
BPSO	0.901	79.58%	44.68%
BGA	0.898	79.99%	50.77%
BOSA-Q3	**0.901**	**80.71%**	**44%**

Table 4:Wilconxon’s test (p-values <0.05 is bold)

Comparison	fitness	Accuracy	No. of Feature
BOSA-Q3 vs HS	**0.00034**	**0.00148**	**0.02491**
BOSA-Q3 vs BPSO	0.63036	**0.00956**	0.51966
BOSA-Q3 vs BGA	0.16319	0.19299	**0.00029**

[1] **A. K. Mandal**, R. Sen and B. Chakraborty, (2019) ”Binary Owl Search Algorithm for Feature Subset Selection,” 2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST),Morioka, Japan, pp. 1–6

[2] **A. K. Mandal**, R. Sen, and B. Chakraborty (2020) ”Analysis of various transfer functions for binary owl search algorithm in feature selection problem” International Journal of Applied Science and Engineering,Chaoyang University of Technology, vol. 17, no. 3, pp. 281–297

MBOSA

1. Extended BOSA and proposed Modified BOSA (MBOSA)
   - Self-adaptive strategy for parameter tuning
   - Elitism mechanism
   - Mutation operation
2. Comparison with BOSA, Binary Bat Algorithm (BBA), Binary Particle Swarm Optimization (BPSO), and Binary Genetic Algorithm (BGA)
3. Use 20 UCI datasets with some high dimensional datasets
4. Decision Tree (DT) with Gini Index in wrapper fitness
5. SVM as final Classification evaluation

Methodology: Self-adaptive strategy

η is defined dynamically

$$\eta_0 = \frac{f_e}{(\sum_{i=1}^{d} f_i)/d} \tag{13}$$

where η_0 is an initial weight of η, expected fitness $f_e = 1.0$ (maximum) fitness value

η update at time t

$$\eta(t) = \eta_0 e^{-10a\frac{t}{T}} \tag{14}$$

where T is total iteration, a controls η

Modify the parameter a, τ is stuck time

$$a(\tau) = \tau/T \tag{15}$$

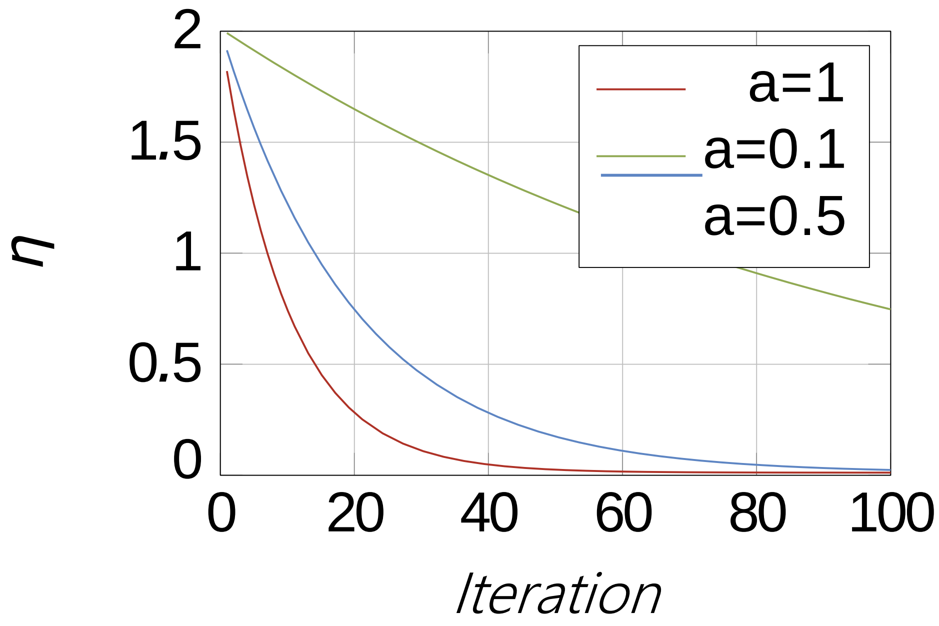


Figure 5: Shape of η during iteration with 3 different a values.

1. Stuck condition: The highest and the lowest fitness values are the same in successive iterations

Methodology: Elitism and Mutation strategy

1. Elitism
   - best individual is selected for the next generation

$$O_i(t) = \begin{cases} O_i(t), & \text{if } f(O_i(t)) \geq f(O_i(t-1)) \\ O_i(t-1), & otherwise \end{cases} \tag{11}$$

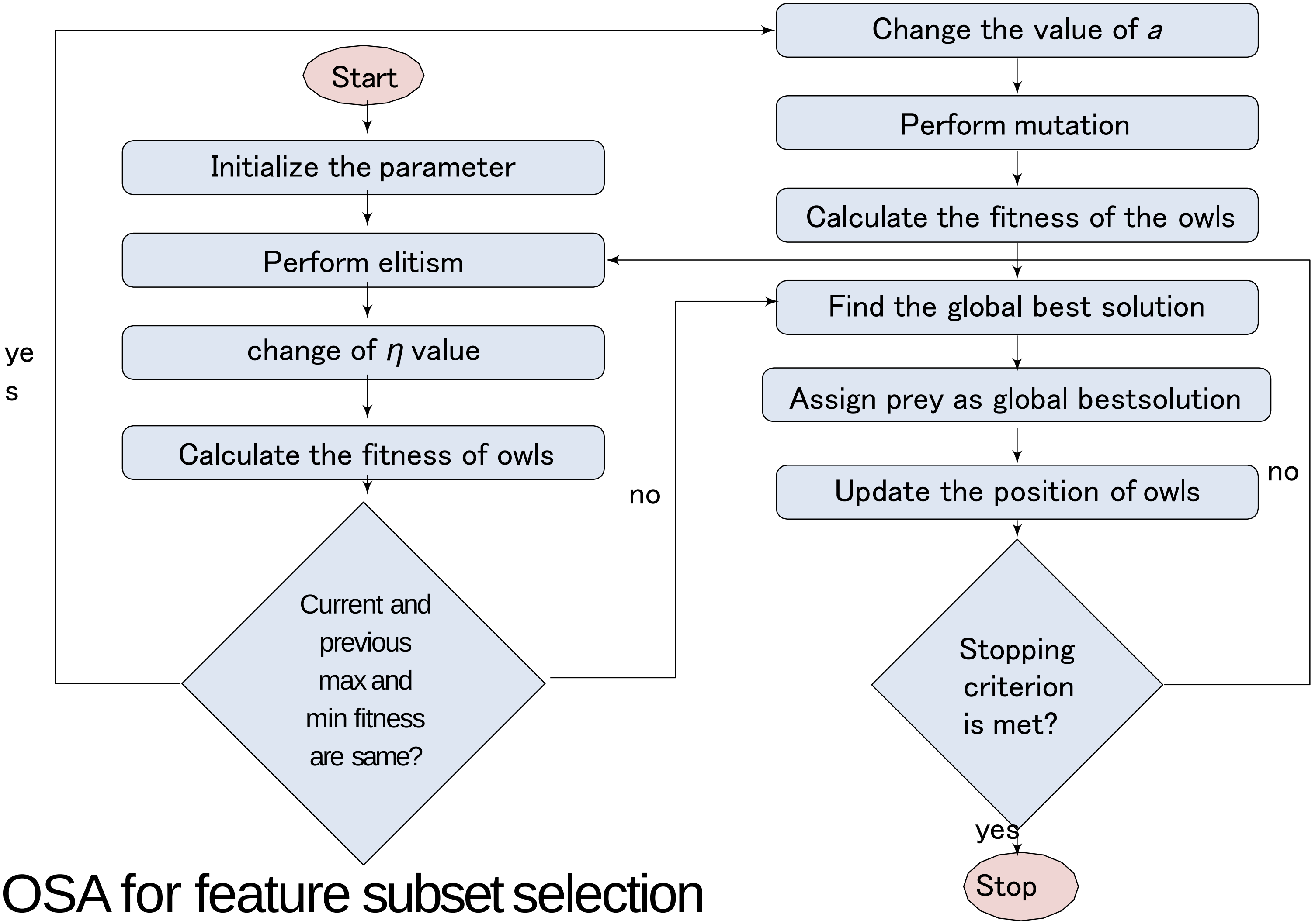
   where $f(O_i(t-1))$ and $f(O_i(t))$ are the fitness of O_i owl at $(t-1)^{th}$ and t^{th} iteration, respectively.

2. Mutation
   - Perturbation to diversify the search
   - performed during stuck condition

$$O_i^j(t) = \begin{cases} 1 - O_i^j(t), & \text{if } R \leq mp_r \\ O_i^j(t), & otherwise \end{cases} \tag{12}$$

   where R is a random number within [0, 1],

MBOSA for feature Subset selection:Flow chart


Start
Initialize the parameter
Perform elitism
change of η value
Calculate the fitness of owls
Current and previous max and min fitness are same?
ye s
no
Change the value of a
Perform mutation
Calculate the fitness of the owls
Find the global best solution
Assign prey as global bestsolution
Update the position of owls
Stopping criterion is met?
no
yes
Stop

Proposed MBOSA for feature subset selection

Results and Discussion

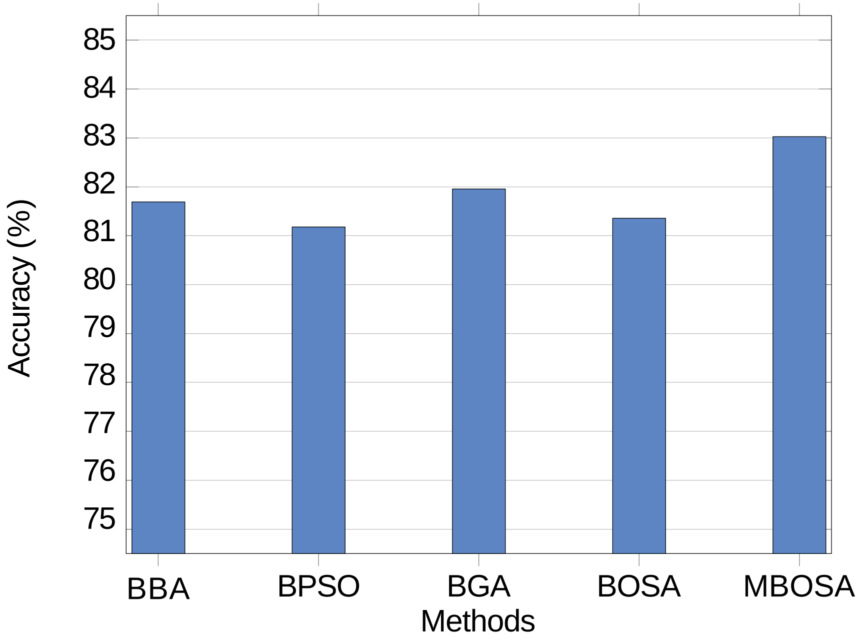


Figure 6:Overall comparison of methods in respect of average classification accuracy.

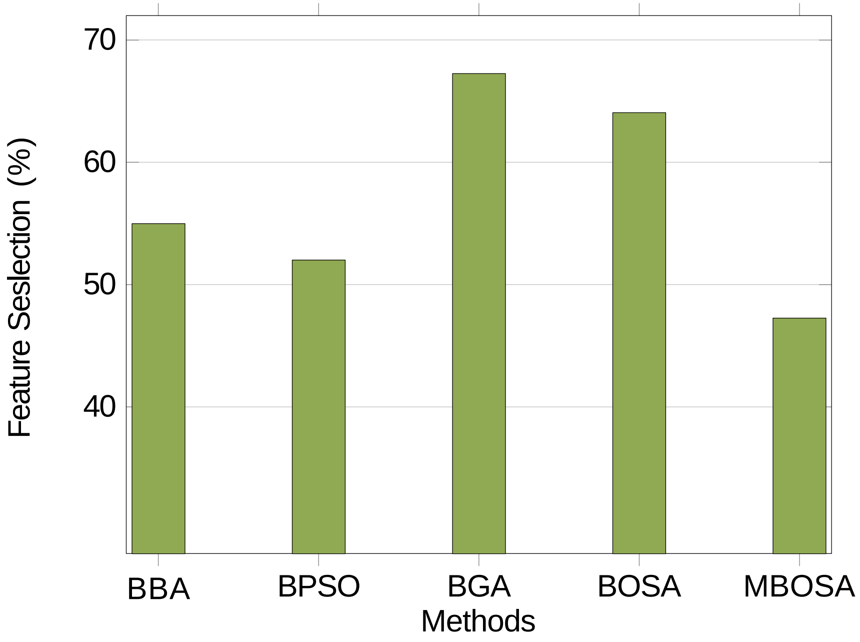
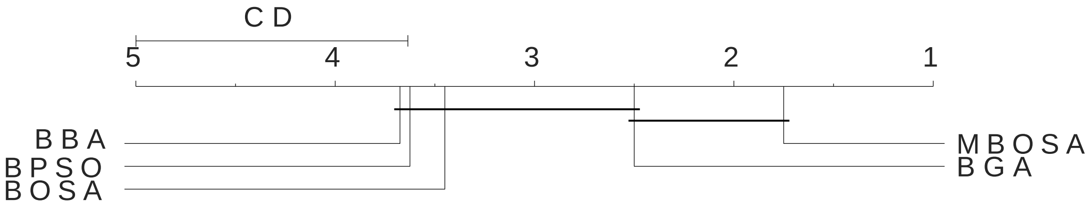


Figure 7:Overall comparison of methods in respect of average feature subset selection ratio.

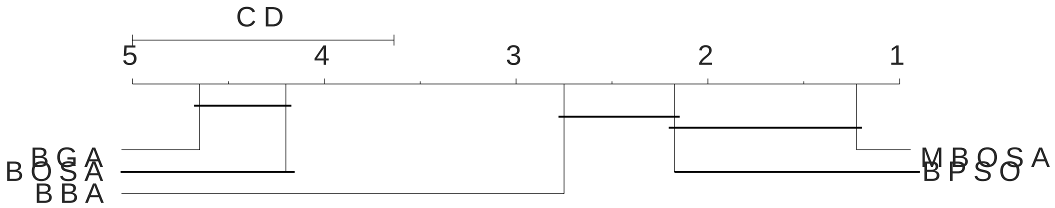
Results and Discussion (cont.)

Table 5:Friedman mean ranks for data sets

Algorithm	mean ranks (Accuracy)	mean ranks (Feature selection)
BBA	3.675	2.750
BPSO	3.625	2.175
BGA	2.500	4.650
BOSA	3.450	4.200
MBOSA	1.750	1.225



(a)Classification Accuracy.



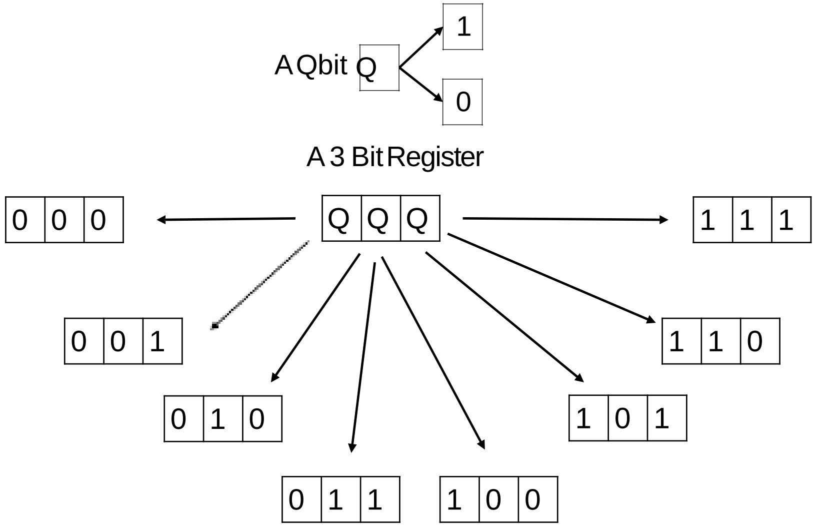
(b)Number of features.

Background: Quantum Computing

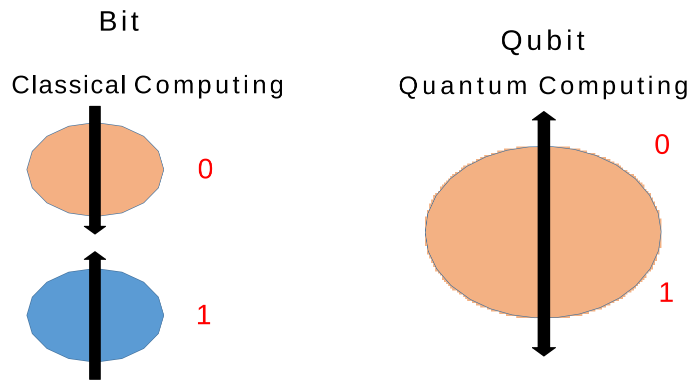
1. Properties of quantum states, such as superposition and entanglement, to perform computation.
2. Speedup Optimization and Machine learning problem

Example

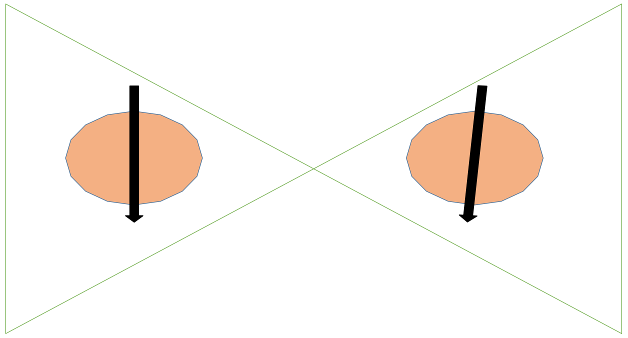
1. $N = 3$ qubit superposition $2^3 = 8$ states at the same time
2. Quantum: Can process these 8 states simultaneously
3. Classical: one state at a time



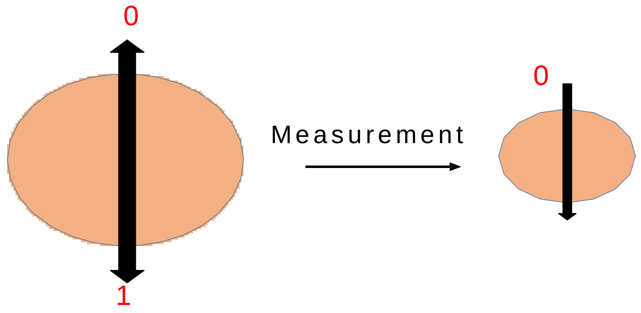
Background: Quantum Computing



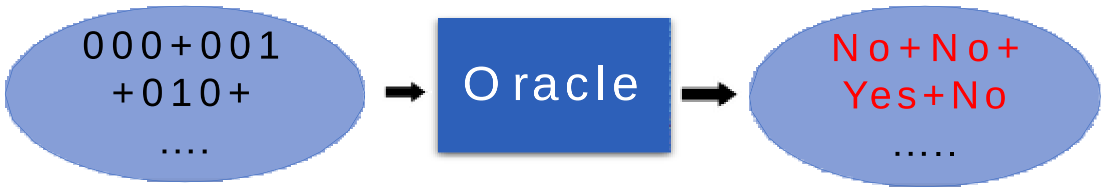
Superposition



Entanglement



Measurement



Oracle

Quantum Inspired OSA (QIOSA)

1. Quantum Inspired strategy in classical computer
    - Each feature is represented by a Q-bit
    - Hadamard gate for initialization
    - Quantum rotation gate drives individual Q-bit string to a better region of solution
    - Wrapper based fitness function
2. QIOSA is tested on twelve datasets
3. Compared with BOSA, BGA, and BPSO
4. K-nearest neighbor (KNN) is used as a classifier both in wrapper and final evaluation

Algorithm

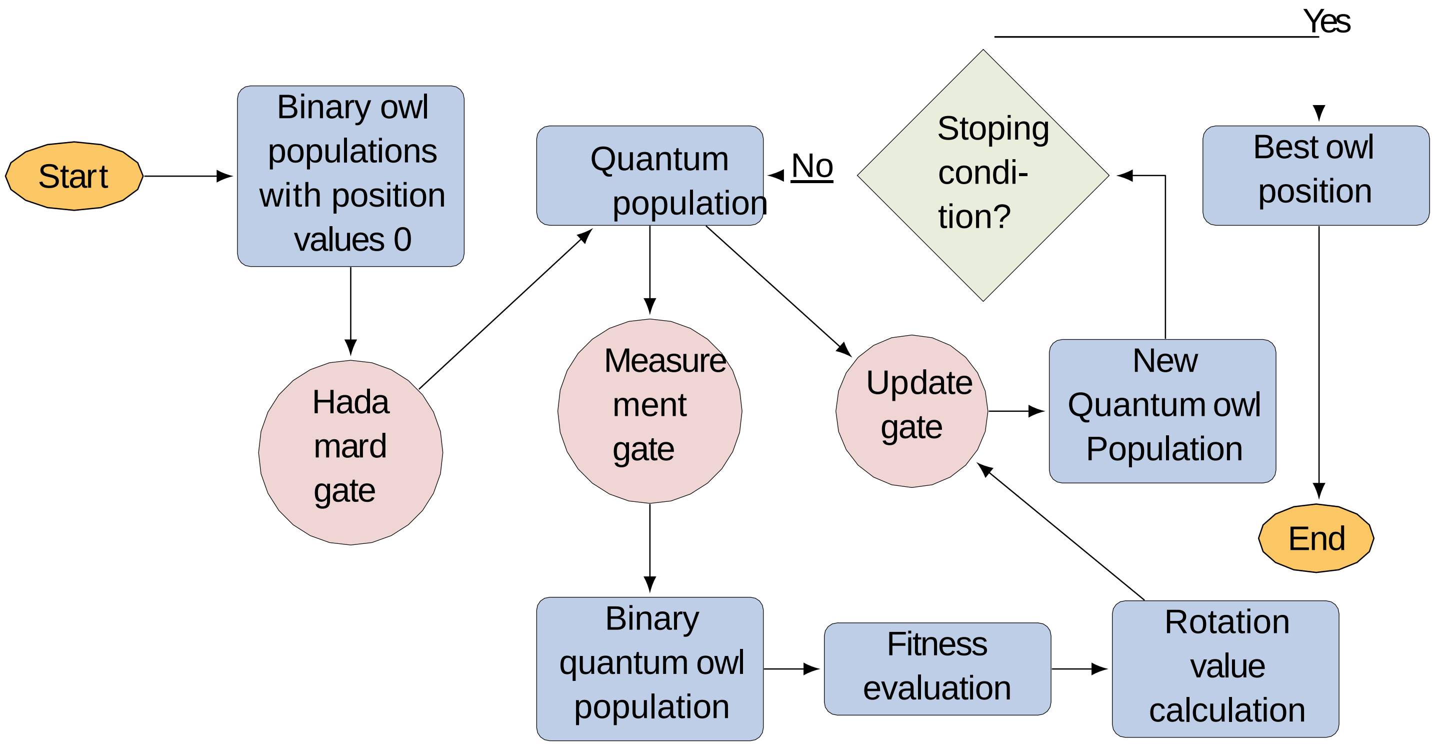


Figure 8:Schematic diagram of involvement of different quantum gates feature subset selection

Quantum gates

Hadamard gate

Used on single Q-bit. It transforms basic states $|0\rangle$ and $|1\rangle$ into superposition states.

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{14}$$

Quantum rotation gate

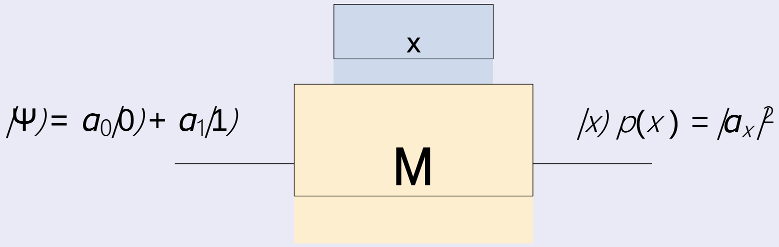
$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{15}$$

where θ indicates rotation angle

Materials and Methods

A Q−bit measurement gate

Collapses a Q-bit to a classic state of either $|0\rangle$ or $|1\rangle$. collapsing value is either 0 or 1.



$|\Psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$
x
M
$|x\rangle\ p(x) = |\alpha_x|^2$

Figure 11:A 1-Q-bit measurement gate.

Quantum Representation of Owls

$$QO_i(t) = \begin{bmatrix} \alpha_i^1(t) & \alpha_i^2(t)\ldots & \alpha_i^d(t) \\ \beta_i^1(t) & \beta_i^2(t)\ldots & \beta_i^d(t) \end{bmatrix} \tag{16}$$

The quantum population of owls is $Q(t) = [QO_1(t), QO_2(t), \ldots.., QO_n(t)]$ at the t^{th} generation. generated from state 0 using Hadamard gate

Materials and Methods (cont.)

Quantum Measurements of Owls

$$BO_i^j(t) = \begin{cases} 0, & \text{if } rand[0,1] < (\alpha_i^j(t))^2 \\ 1, & \text{if } rand[0,1] \geq (\alpha_i^j(t))^2 \end{cases} \tag{17}$$

Rotation Value Calculation

$$\Delta\Theta_i(t+1) = \begin{cases} \Delta\Theta_i(t) + \eta \times lc_i \times |\zeta V - BO_i(t)|, & \text{if } p_{vm} < 0.5 \\ \Delta\Theta_i(t) - \eta \times lc_i \times |\zeta V - BO_i(t)|, & \text{if } p_{vm} \geq 0.5 \end{cases} \tag{18}$$

Results and Discussion

Table 6:Average Classification Accuracies of different methods

Dataset	QIOSA	BOSA	BGA	BPSO
arrhythmia	61.95	60.37	61.8	60.88
breast-w	96.95	96.73	97.10	96.76
clean1	85.59	84.72	85.98	84.86
ionosphere	84.2	83.16	83.35	83.54
libras move	68.89	68.84	68.47	67.78
Parkinsons	89.32	86.95	88.73	88.56
qsar-biodeg	82.85	81.85	82.81	82.62
semeion	87.92	88.32	88.77	87.72
sonar	77.30	76.35	75.16	76.67
vehicle	68.88	68.35	70.43	67.89
wine	89.81	89.63	89.44	89.07
zoo	98.71	96.29	96.77	96.61

Results and Discussion (cont.)

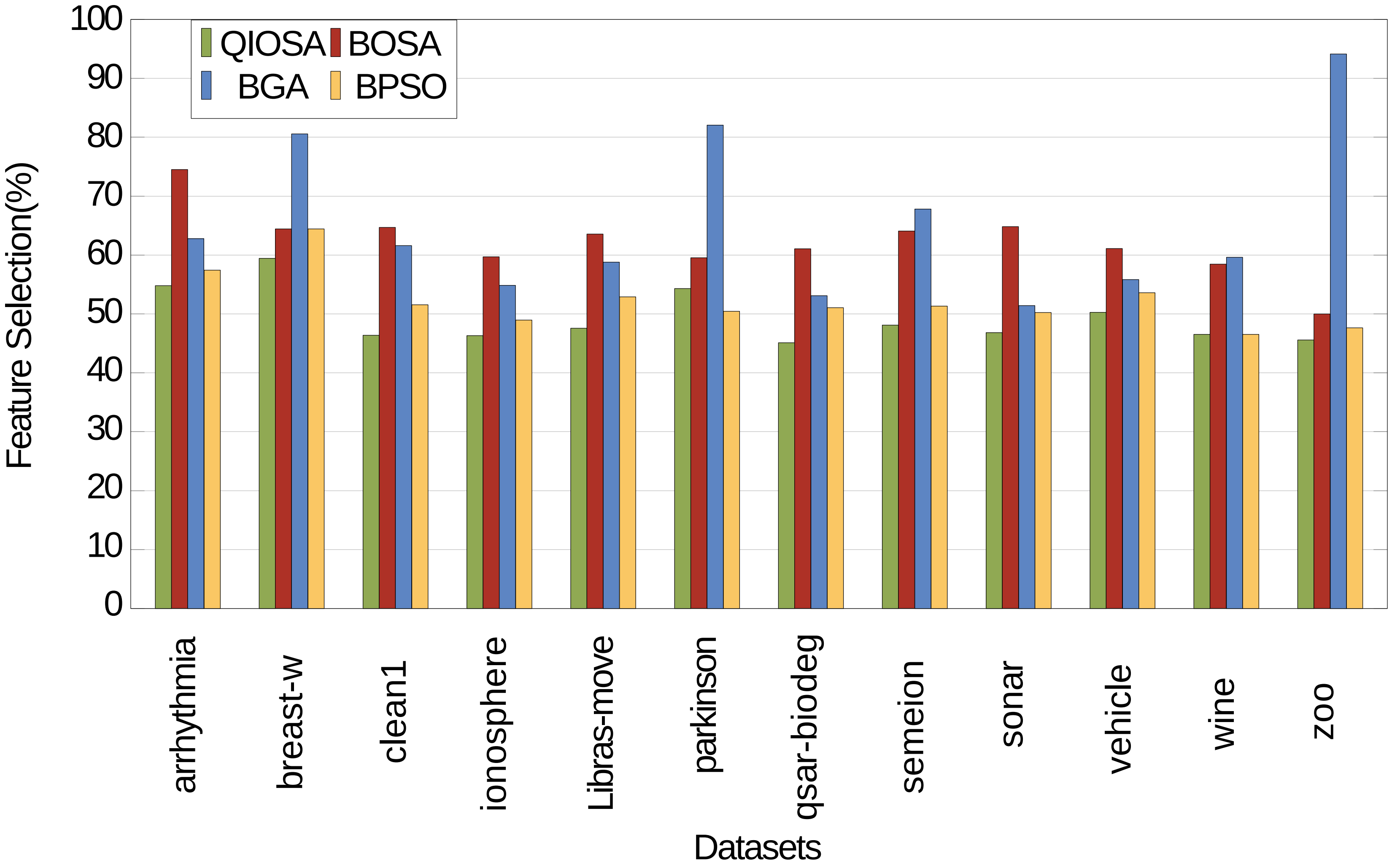


Figure 9:Comparison of approaches in terms of average feature selection (%)

Results and Discussion (cont.)

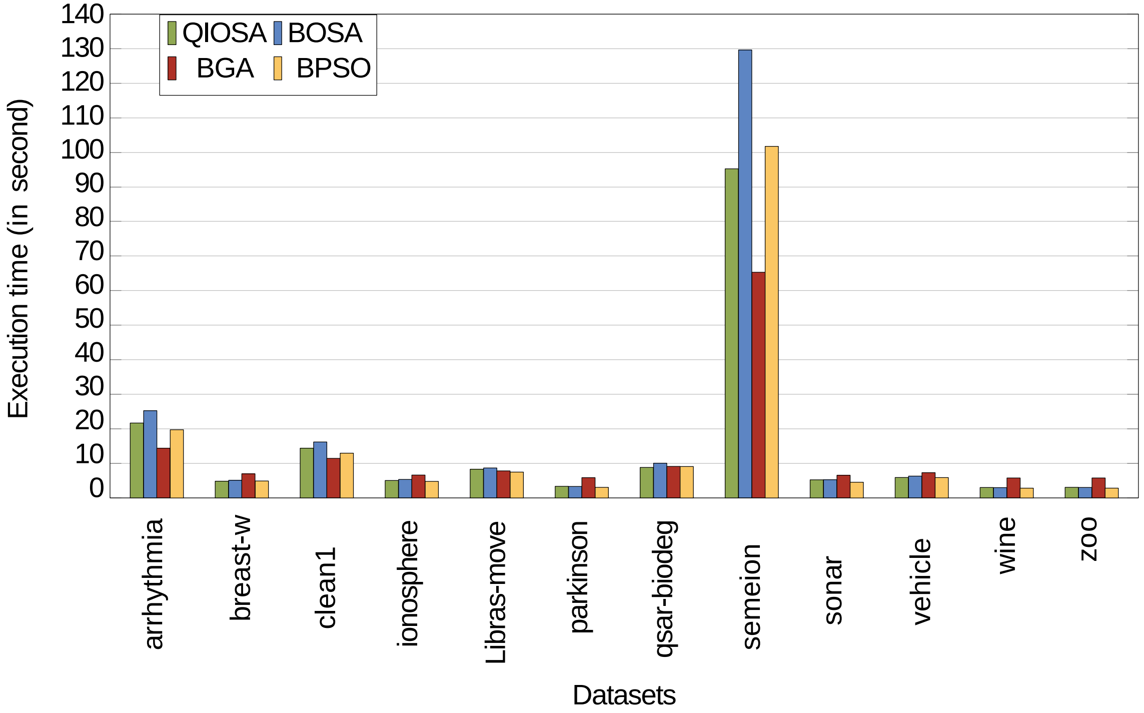


Figure 10:Comparison of approaches in terms of average computational time.

[4] A. K. Mandal, R. Sen, S. Goswami, A. Chakrabarti and B. Chakraborty, (2020)"A New Approach for Feature Subset Selection using Quantum Inspired Owl Search Algorithm," 10th International Conference on Information Science and Technology (ICIST) London, UK , 2020, pp. 266–273.

Gene Subset Selection in Microarray Data with QIOSA$_f$

1. Quantum inspired strategy
2. Two stage strategy
   - Ensemble of filter ranking
   - QIOSA$_f$ in filter subset selection
3. Fitness function based of MIFS
4. QIOSA is tested on twenty-five Microarray datasets
5. Compared with BOSA, BGA, and BPSO , QIOSA$_W$(Quantum inspired owl search wrapper)
6. K -nearest neighbor (KNN) is used as a classifier in wrapper
7. SVM with linear kernel for final evaluation

Methodology

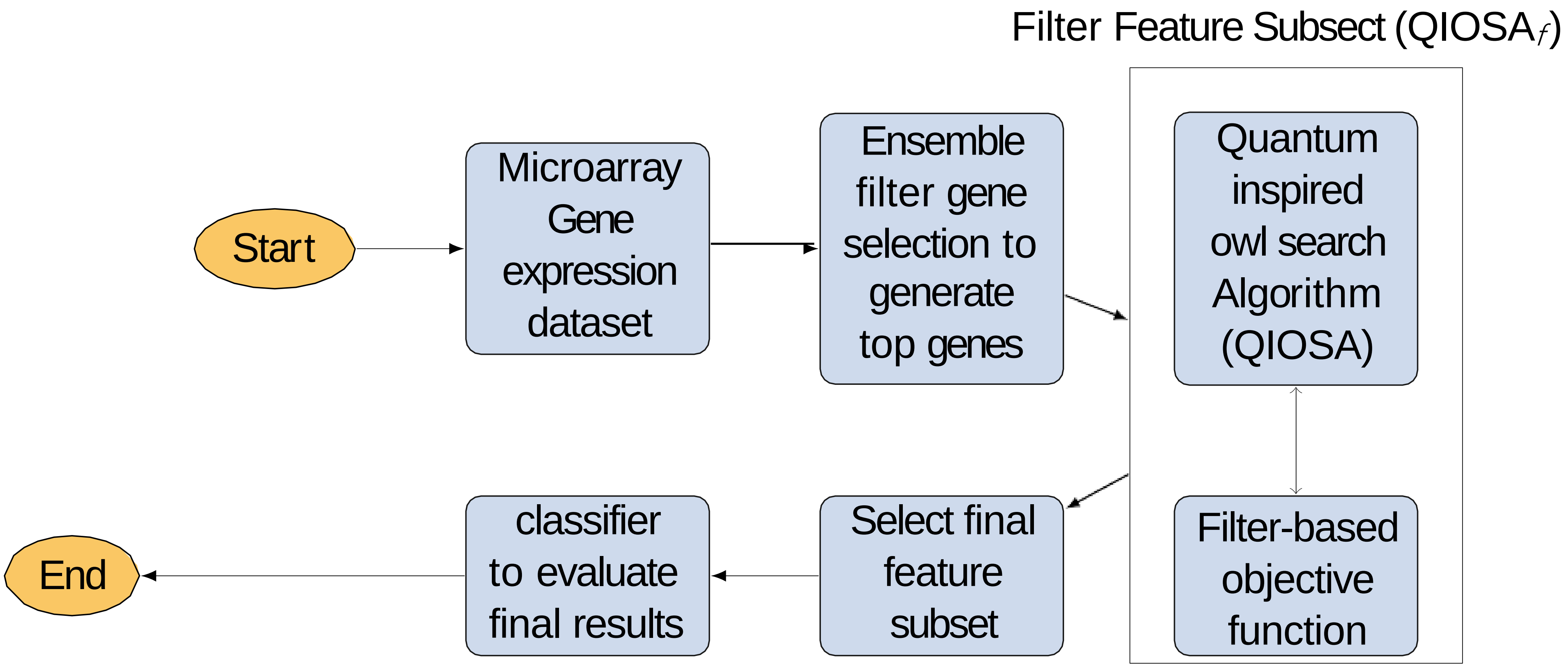

Filter Feature Subsect (QIOSA$_f$)
Start
Microarray Gene expression dataset
Ensemble filter gene selection to generate top genes
Quantum inspired owl search Algorithm (QIOSA)
Filter-based objective function
Select final feature subset
classifier to evaluate final results
End

Figure 11:The outline of the proposed gene selection and classification approach

Methodology(cont.)

1. Owl Search Algorithm based filter approach (QIOSA$_f$) to Gene subset Selection
    1. Quantum Representation
    2. Quantum Measurements
    3. Rotation Value Calculation
    4. Quantum Update Mechanism

Filter Fitness Function: Mutual Information based Feature Selection (MIFS)

Results and Discussion (cont.)

Table 7:win–loss-tie comparison between QIOSA$_f$ with QIOSA$_W$ method

	Measure	QIOSA$_W$
QIOSA$_f$	Accuracy	W(0), T(22), L(3)
	Gene Selection	W(25), T(0), L(0)

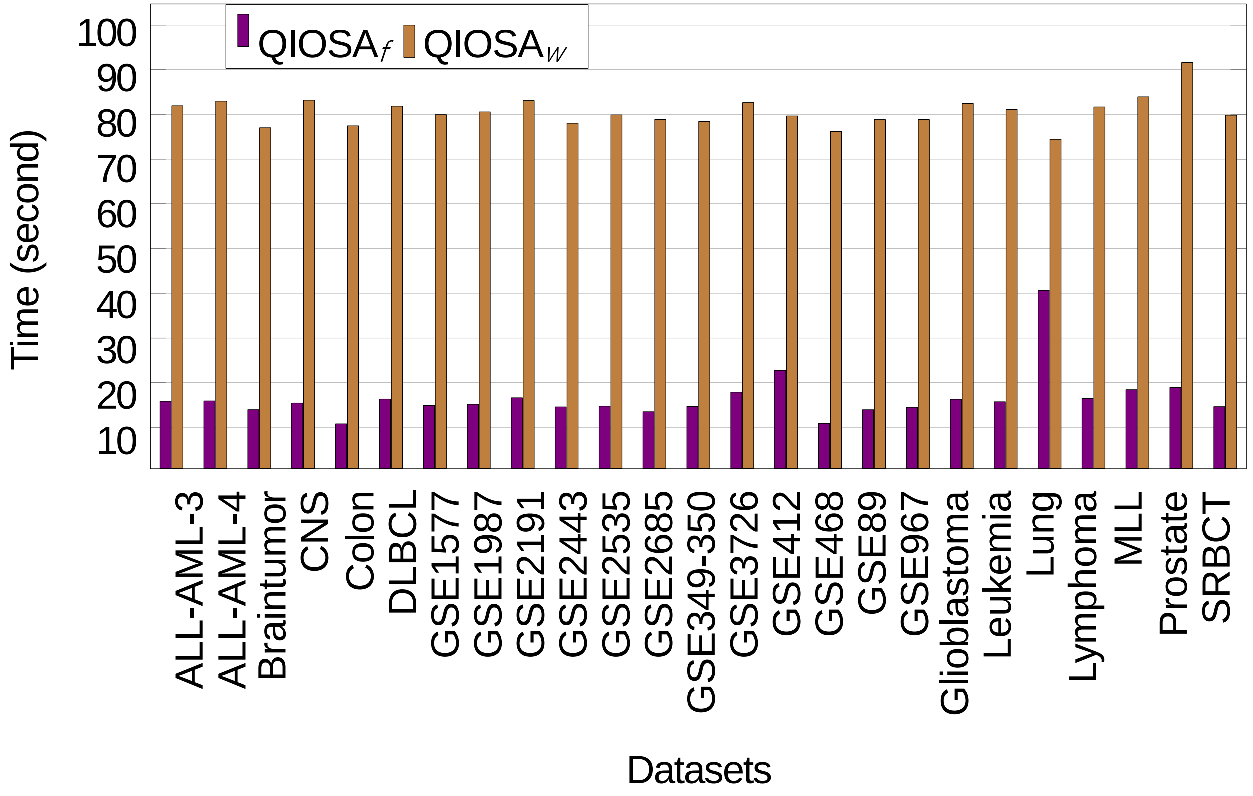


Figure 12:Computational time comparison between QIOSA $_f$ and QIOSA$_W$

Results and Discussion (cont.)

Table 8:win–loss-tie comparison between QIOSA $_f$with other methods

	Measure	BPSO	BGA	BOSA
QIOSA$_f$	Accuracy	W(2), T(23), L(0)	W(2),T(23), L(0)	W(0), T(22), L(3)
	Gene Selection	W(25), T(0), L(0)	W(21), T(4), L(0)	W(25), T(0), L(0)

Results and Discussion (cont.)

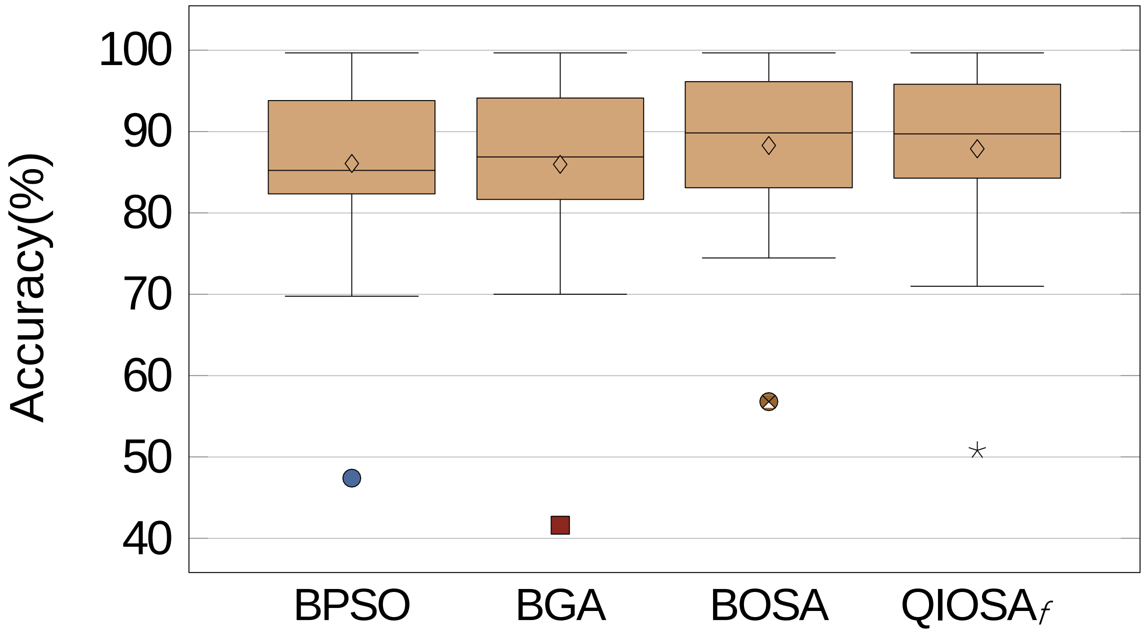


Figure 13:Classification Accuracy

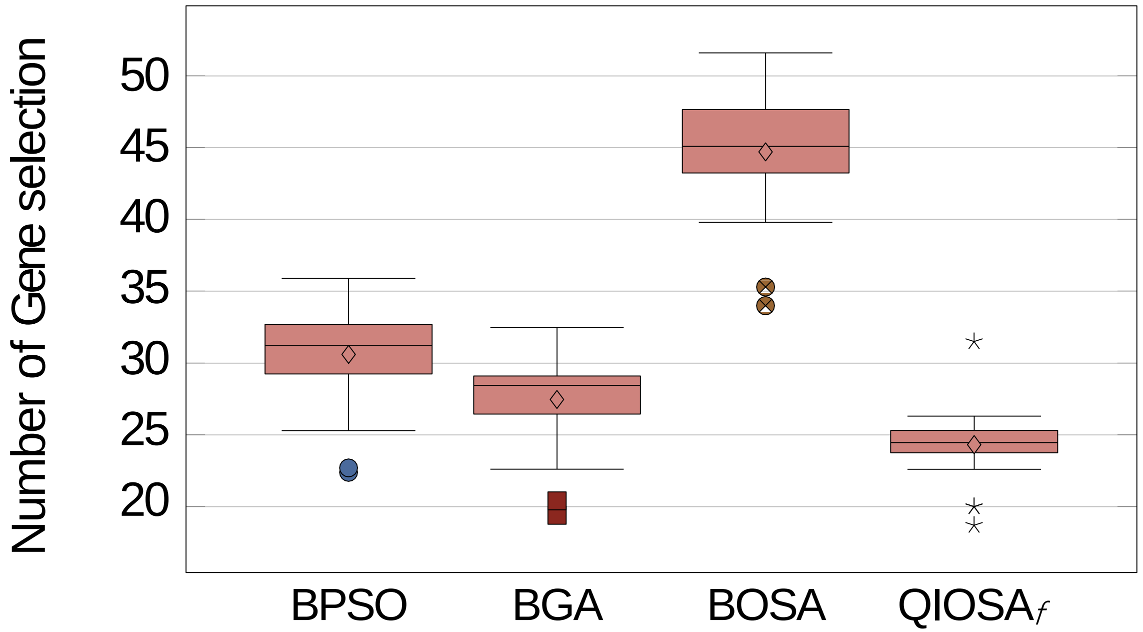


Figure 14:Number of gene selection

Results and Discussion (cont.)

Table 9:Comparison of the suggested technique to existing approaches in the literature

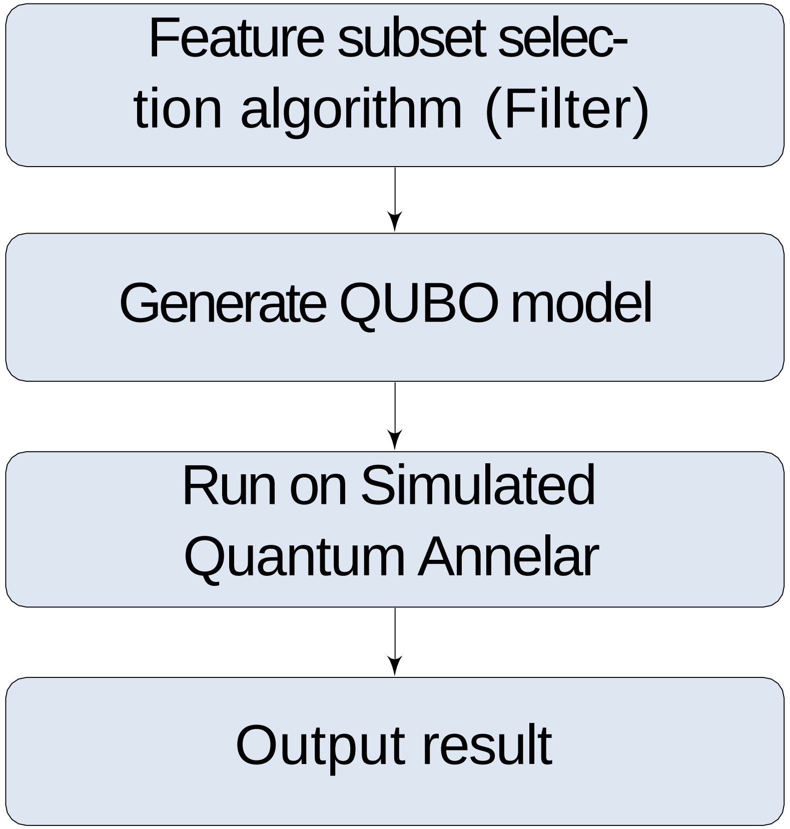
	measure	ALL-AML-3	ALL-AML-4	CNS	Colon	DLBCL	Leukemia	Lymphoma	MLL	SRBCT
ISFLA [4]	ACC	94	90.91	-	93.02	73.33	98.91	-	92.59	93.75
	G	40	32.2	-	35.22	27.42	30.44	-	40.7	43.1
IBCFPA [5]	ACC	-	-	84.82	92.16	-	-	99.6	96.51	98.02
	G	-	-	25.2	25.9	-	-	20.1	47.21	40.8
BCFPA [6]	ACC	-	-	80.33	88.47	-	-	98.43	91	94.78
	G	-	-	22.7	30.9	-	-	24.8	53.9	34.2
BCROSAT [7]	ACC	94.5	90.9	-	-	77.49	-	-	98.04	95.72
	G	32	30.9	-	-	23.16	-	-	35.6	33
MBEGA [8]	ACC	96.64	91.93	72.21	85.66	-	-	97.68	94.33	99.23
	G	18.1	26.2	20.5	24.5	-	-	34.3	32.1	60.7
EBWAS [9]	ACC	84.11	78.54	67.69	80.56	-	-	-	83.16	79.37
	G	49	47.2	52.5	16.2	-	-	-	86.3	17.5
Proposed	ACC	96.59	91.82	**85.67**	85.21	**94.8**	96.29	95.64	**96.63**	**99.38**
	G	23.7	**24**	24.7	20	23.3	**24.6**	24.1	**25**	20

ACC= Classification accuracy, G= Number of selected Genes

Quantum Annealing

1. Optimization problems can be transferred into QUBO
2. Quantum computer based on Quantum annealing can solve QUBO
3. Little attention on feature subset selection in quantum computer
4. Transforming into QUBO is challenging
5. Quantum annealing (simulator) is used

Materials and Methods


Feature subset selection algorithm (Filter)
Generate QUBO model
Run on Simulated Quantum Annelar
Output result

1. Simulated Annealing (SA)
   - metaheuristic technique
   - optimizes based on the physical annealing process
2. Simulated Quantum Annealing (SQA)
   - Nishimori and Kadowaki introduced Quantum annealing
   - Mapping of the quantum annealing in the classical computer
   - Markov Chain Monte Carlo (MCMC) algorithm for simulation
3. We use Sqaod solver as SQA
   - address Ising problems in classical CPU and CUDA (Nvdia GPU)

QUBO for feature subset selection

Materials and Methods: Mapping into QUBO

FCBF

1. FCBF breaks the mRMR objective into 2 stage optimization
2. first is optimizing the relevance
3. next step is optimizing redundancy

FCBF QUBO

1. Maximize Relevancy:

$$X^T[D]X \tag{19}$$

2. Minimize Redundancy:

$$X^T[D - \frac{1}{|S|}M]X \tag{20}$$

Results and Discussion

- We have used Minimum Redundancy Maximum Relevance (mRMR) QUBO

Table 10:Description of Datasets

Datasets	Num. of features	No. of classes	No. of Instances
CMC	9	3	1473
Dermatology	34	6	366
Wisconsin	10	2	699
Ecoli	7	6	336
Iris	4	3	150
Lung-cancer	56	3	32
Lymphography	18	2	148
Vehicle WBDC	18	4	846
Wine	32	2	570
	13	3	178

	F-1	F-2	F-3	F-4	F-5	F-6	F-7
F-1	-0.311	0.038	0.000	0.000	0.011	0.044	0.018
F-2	0.038	-0.339	0.000	0.000	0.007	0.032	0.010
F-3	0.000	0.000	-0.149	0.000	0.000	0.000	0.000
F-4	0.000	0.000	0.000	0.003	0.000	0.000	0.000
F-5	0.011	0.007	0.000	0.000	0.209	0.021	0.024
F-6	0.044	0.032	0.000	0.000	0.021	0.113	0.128
F-7	0.018	0.010	0.000	0.000	0.024	0.128	0.438

Figure 15:QUBO matrix of E. coli dataset

Results and Discussion (cont.)

- SQA can produce less number of features
- SQA produces stable feature subset

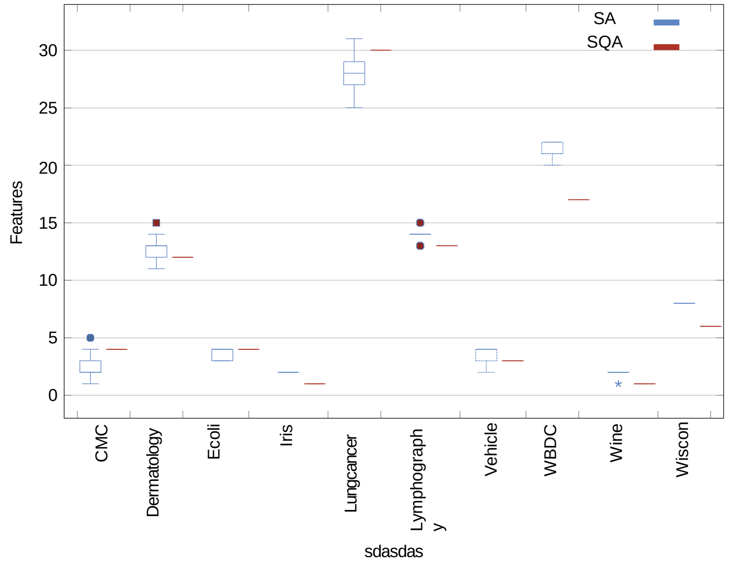


Figure 16:Statistical box plot of the experimental results for SA (left columns) and SQA (right columns)for each dataset

[7]A.K. Mandal , M. Panday ,A. Biswas , S. Goswami, A. Chakrabarti , B. Chakraborty (2021) An Approach of Feature Subset Selection Using Simulated Quantum Annealing. In: Sharma N., Chakrabarti A., Balas V., Martinovic J. (eds) Data Management, Analytics and Innovation. Advances in Intelligent Systems and Computing, vol 1174. Springer,Singapore.

Conclusion

- We developed metaheuristic Owl search based feature selection algorithm with concepts from quantum paradigm.
- Incorporation of quantum inspired concepts /strategies led to more efficient algorithm in terms of reducing number of features without sacrificing classification accuracy.
- Quantum inspired filter algorithms are computationally also comparable to other state of the algorithms using metaheuristics.
- Quantum annealing based algorithm produced more stable features but have high computational cost.