Feature Selection with Learning Tabu Search

LUCIEN MOUSIN, LAETITIA JOURDAN, MARIE-ELEONORE MARMION, CLARISSE DHAENENS

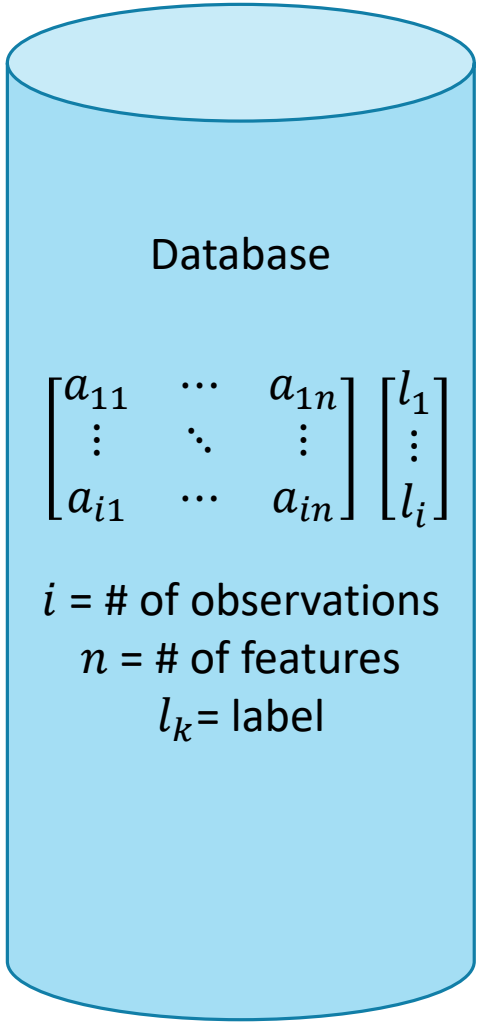
DOLPHIN TEAM

30 MAY 2016

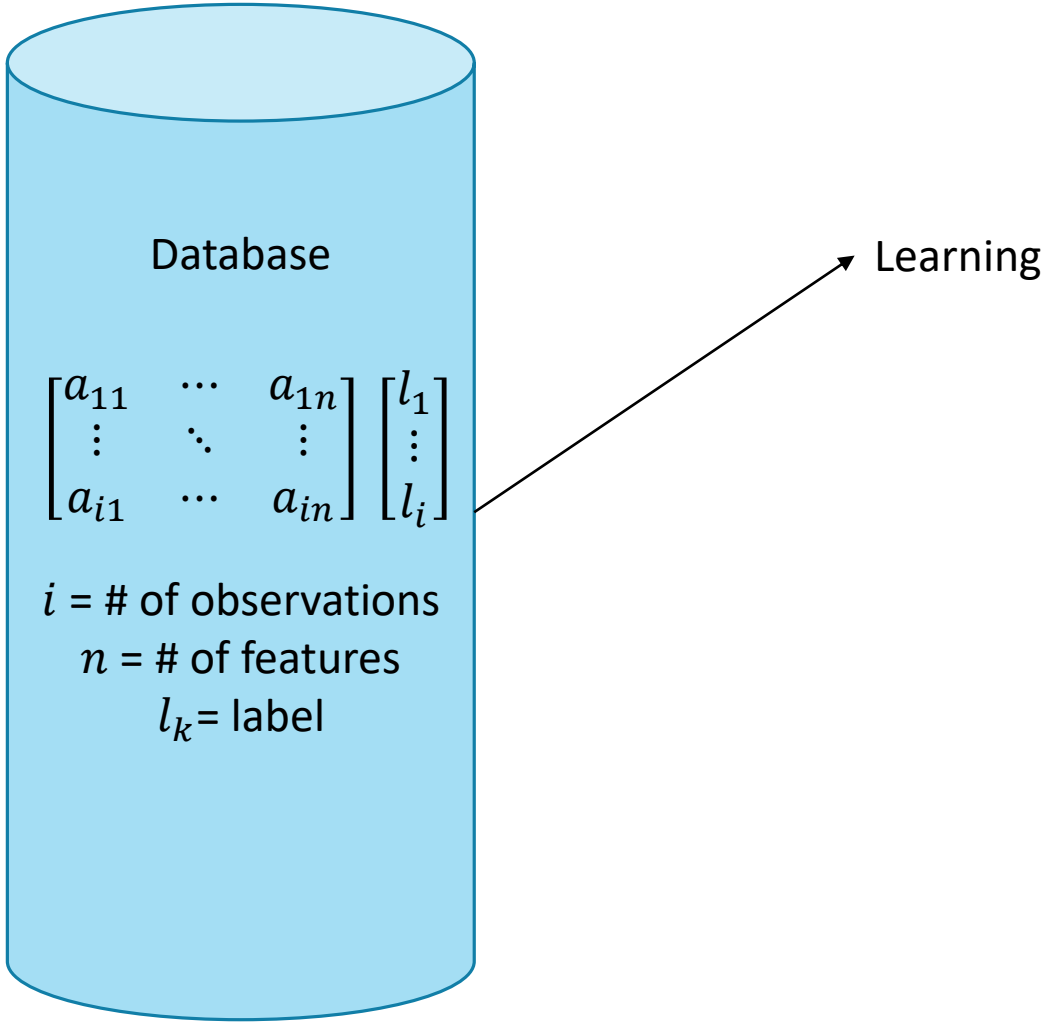
Plan

- Introduction
- Learning Tabu Search
- Experiments

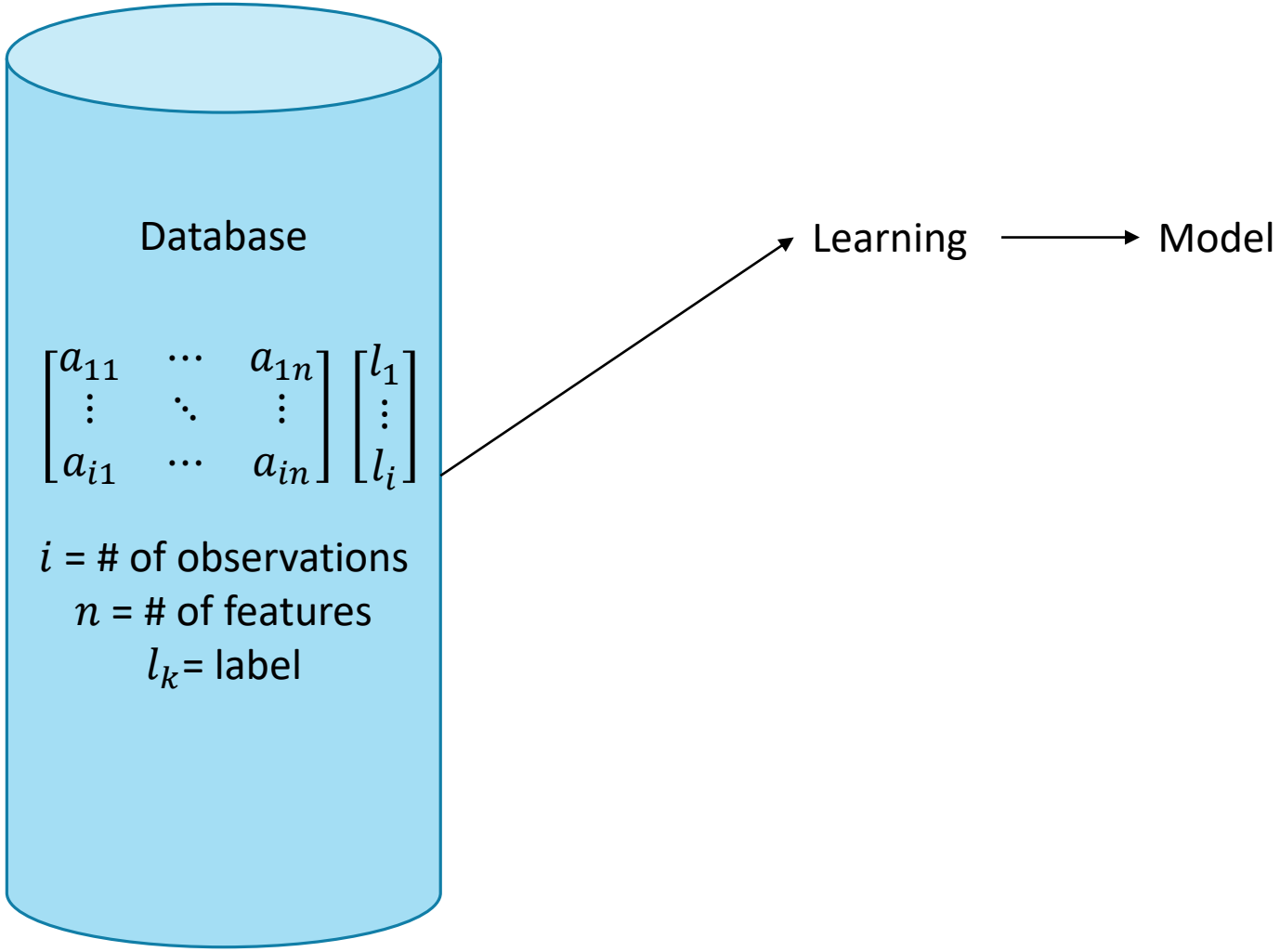
Context – Feature selection in classification



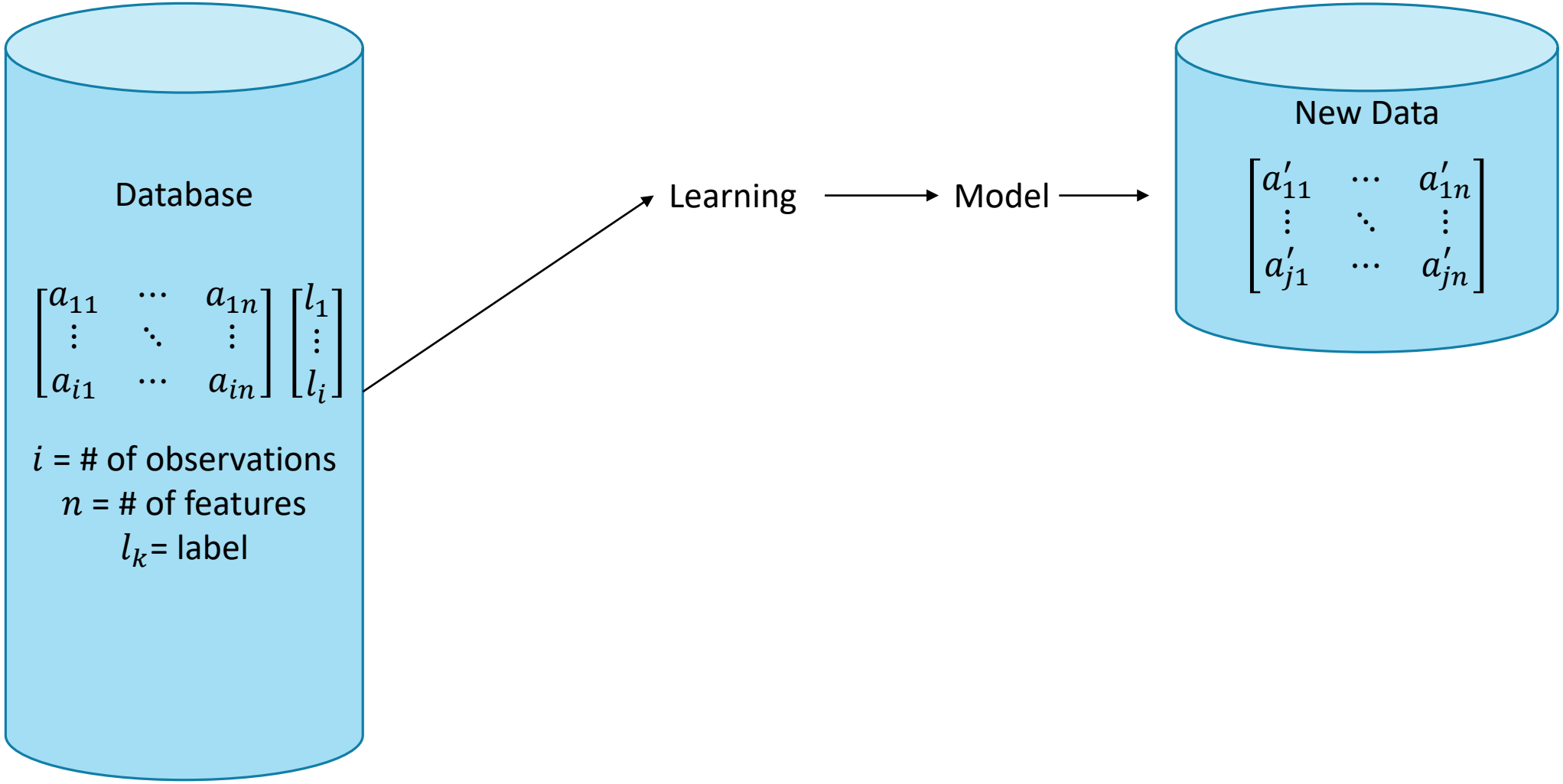
Context – Feature selection in classification



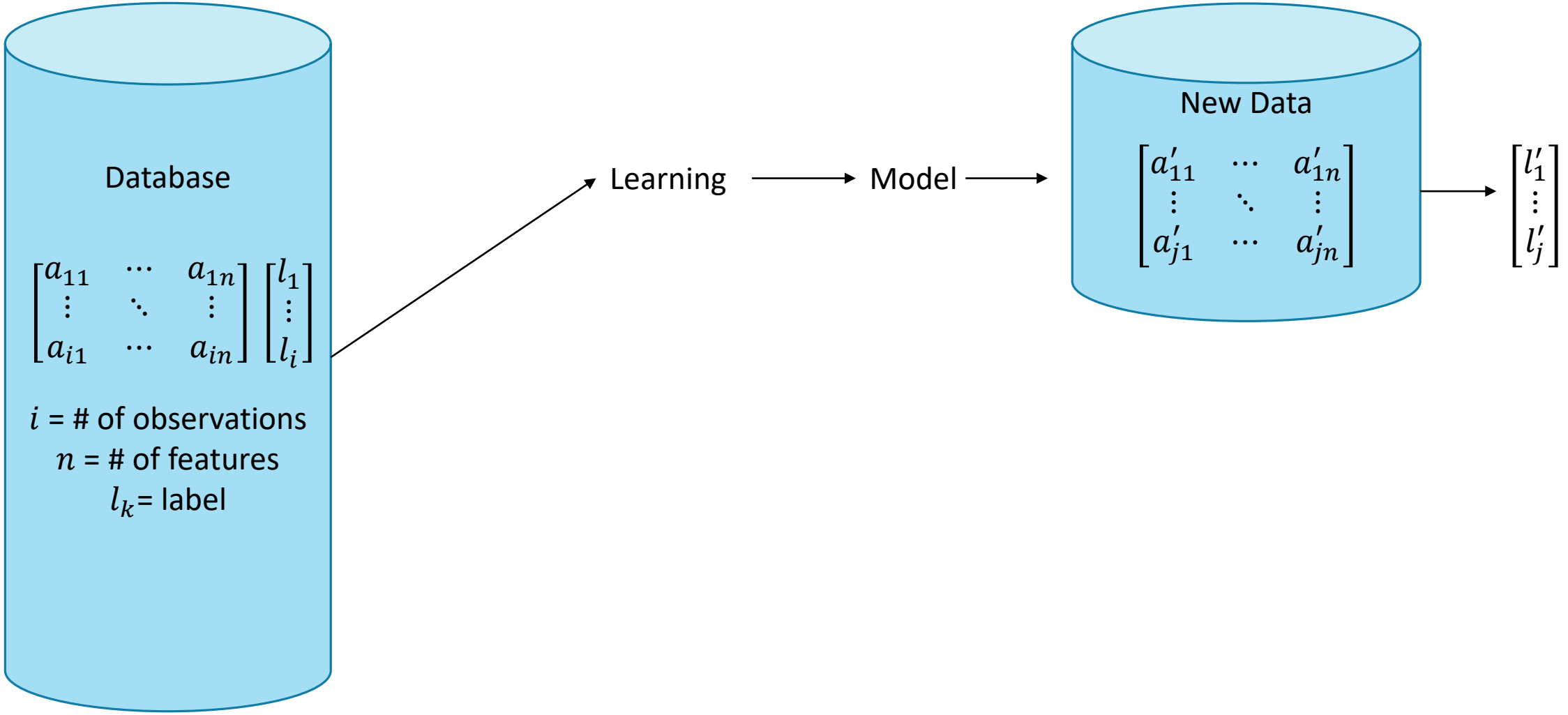
Context – Feature selection in classification



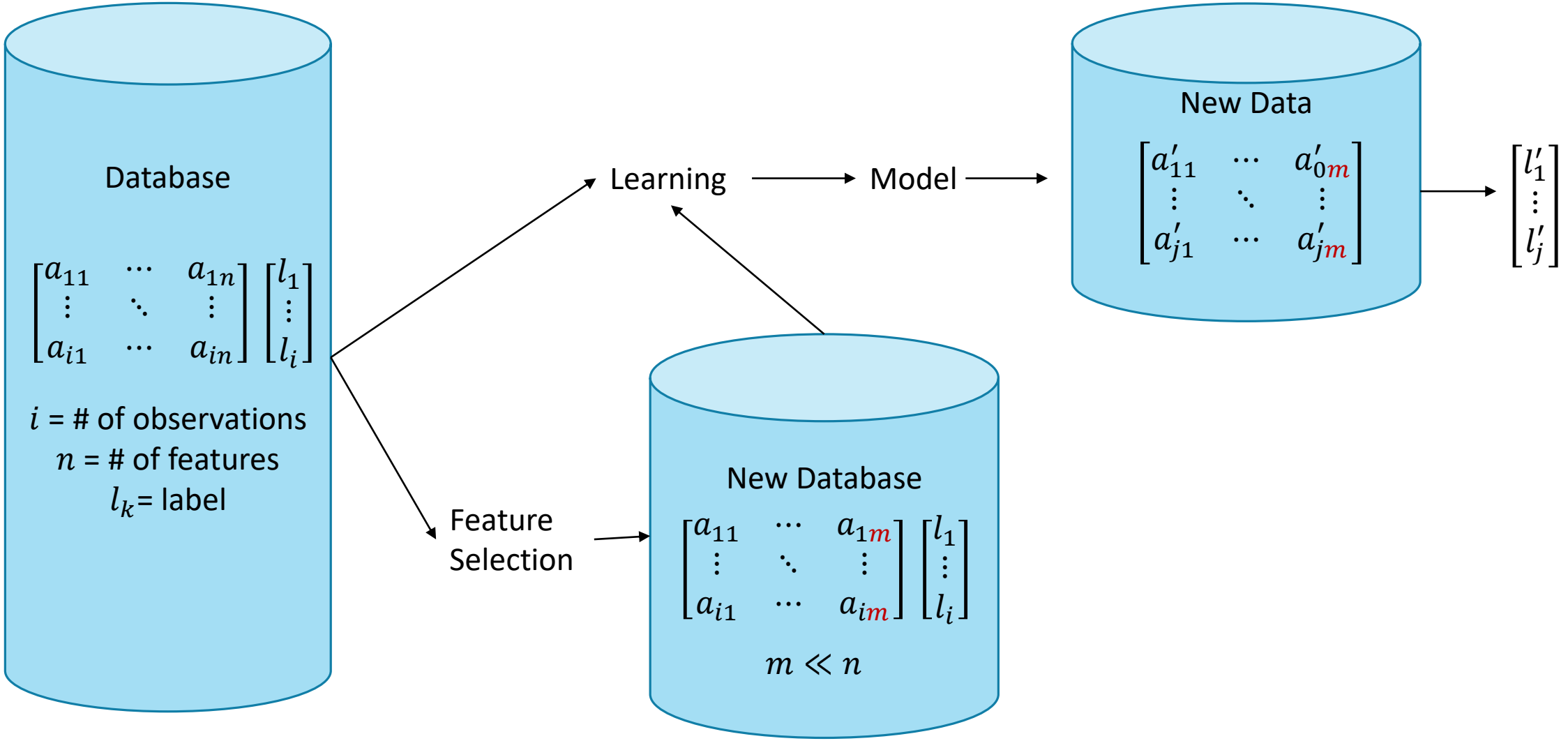
Context – Feature selection in classification



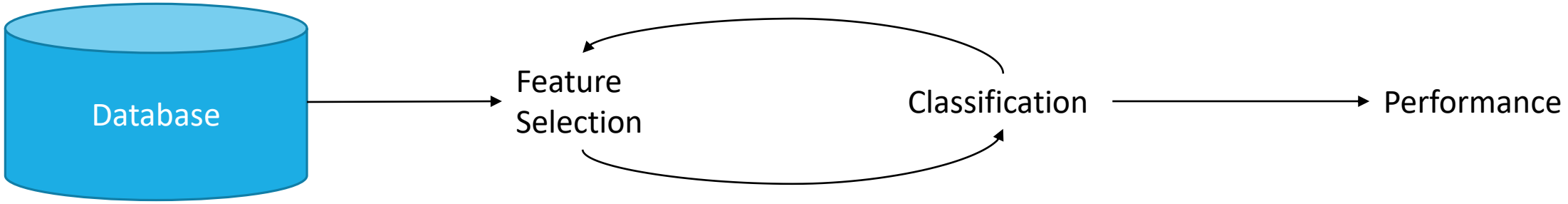
Context – Feature selection in classification



Context – Feature selection in classification



Wrapper Approach

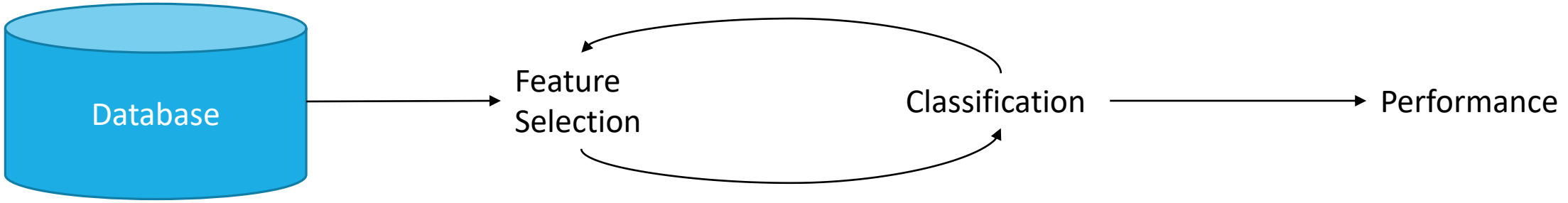


Feature Selection = Chooses m features among n ($m \ll n$)

Classification = Expensive evaluation

Search space = $O(2^n)$ → exhaustive search is impractical unless n is small

Wrapper Approach



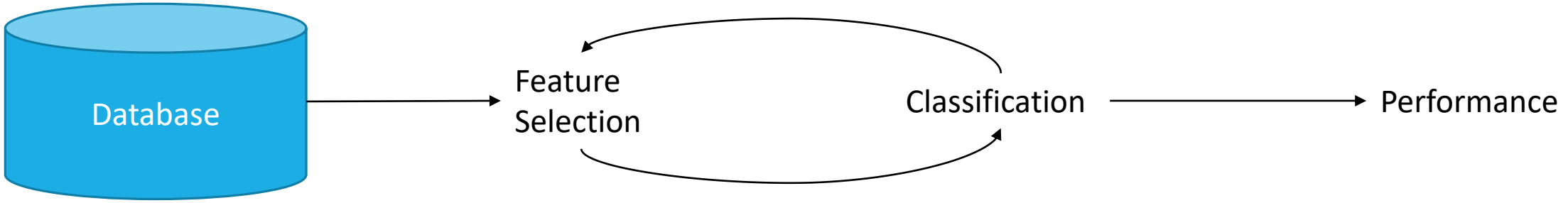
Feature Selection = Chooses m features among n ($m \ll n$)

Classification = Expensive evaluation

Search space = $O(2^n)$ → exhaustive search is impractical unless n is small

→ Estimation

Wrapper Approach



Feature Selection = Chooses m features among n ($m \ll n$)

Classification = Expensive evaluation

Search space = $O(2^n)$ → exhaustive search is impractical unless n is small

→ Estimation

Solve Feature Selection problem with a metaheuristic with knowledge incorporation

Learning Tabu Search

FS modeling – Representation and Evaluation

REPRESENTATION OF SOLUTIONS

Bit string of size n:

$$s = (a_1, \dots, a_n) \; with \; \forall i \in \{1, \dots, n\}, a_i \in \{0,1\}$$

a_i indicates if the feature i is chosen $(a_i = 1)$ or not $(a_i = 0)$

FS modeling – Representation and Evaluation

REPRESENTATION OF SOLUTIONS

Bit string of size n:

$$s = (a_1, \dots, a_n) \; with \; \forall i \in \{1, \dots, n\}, a_i \in \{0,1\}$$

a_i indicates if the feature i is chosen $(a_i = 1)$ or not $(a_i = 0)$

EVALUATION OF SOLUTIONS

$$accuracy = \frac{number\ of\ well\_classified\ observations}{total\ number\ of\ observations}$$

$$features = 1 - \frac{number\ of\ selected\ features}{total\ number\ of\ features}$$

Two maximization criteria

FS modeling – Representation and Evaluation

REPRESENTATION OF SOLUTIONS

Bit string of size n:

$$s = (a_1, \dots, a_n) \; with \; \forall i \in \{1, \dots, n\}, a_i \in \{0,1\}$$

a_i indicates if the feature i is chosen ($a_i = 1$) or not ($a_i = 0$)

EVALUATION OF SOLUTIONS

$$accuracy = \frac{number\ of\ well\_classified\ observations}{total\ number\ of\ observations}$$

$$features = 1 - \frac{number\ of\ selected\ features}{total\ number\ of\ features}$$

Two maximization criteria

$$f = \alpha * accuracy + (1 - \alpha) * features$$
$$\alpha \in [0,1]$$

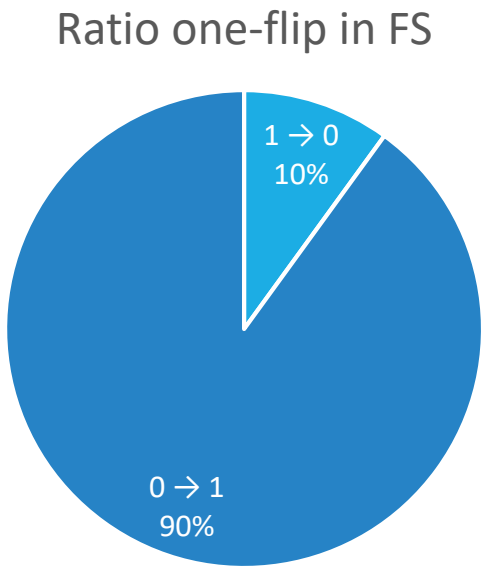
Single-objective

FS modeling - Neighborhood

$$\mathcal{N}_1^0(s) = \{\, s' | \exists i \in \{1, \ldots, n\}\, s.t.\, a_i' \neq a_i \; and \; \forall j \neq i, a_j' = a_i \}$$

FS modeling - Neighborhood

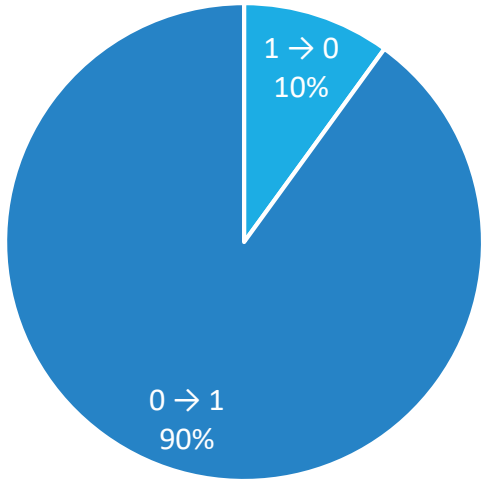
$$\mathcal{N}_1^0(s) = \{\, s' | \exists i \in \{1, \ldots, n\}\, s.t.\, a_i' \neq a_i \; and \; \forall j \neq i, a_j' = a_i \}$$



FS modeling - Neighborhood

$$\mathcal{N}_1^0(s) = \{\, s' | \exists i \in \{1, \dots, n\}\, s.t.\, a_i' \neq a_i \; and \; \forall j \neq i, a_j' = a_i \}$$

Ratio one-flip in FS



$$\mathcal{N}_{Add}(s) = \{\, s' | \exists i \in \{1, \dots, n\} \; with \; a_i' = 1 \; and \; a_i = 0 \; and \; \forall j \neq i, a_j' = a_i \} \quad (0 \rightarrow 1)$$

$$\mathcal{N}_{Drop}(s) = \{\, s' | \exists i \in \{1, \dots, n\} \; with \; a_i' = 0 \; and \; a_i = 1 \; and \; \forall j \neq i, a_j' = a_i \} \quad (1 \rightarrow 0)$$

Learning Tabu Search

Algorithm 1: Learning Tabu Search (LTS)

```
begin
    s ← initial solution;
    s* ← s;
    repeat
        Estimate the quality of non-tabu neighbors of 𝒩(s);
        N_Q ← Q most promising neighbors of 𝒩(s) according to the
        diversification policy;
        s ← max_{s'∈N_Q} f(s');
        if s > s* then
            s* ← s;
        if s > ŝ then
            ŝ ← s;
        Update the tabu list;
        if End of cycle then
            Update trails of each combination with ŝ;
    until Stopping condition is met;
    return s*
```

s = *current solution*
s' = *neighboring solution of* s
s^* = *best solution from the begining*
$\hat{s}$ = *best solution from the last cycle*

D. Schindl and N. Zufferey - Solution Methods for Fuel Supply of Trains. (2013)

Learning Tabu Search

Algorithm 1: Learning Tabu Search (LTS)

```
begin
    s ← initial solution;
    s* ← s;
    repeat
        Estimate the quality of non-tabu neighbors of N(s);
        N_Q ← Q most promising neighbors of N(s) according to the
        diversification policy;
        s ← max_{s'∈N_Q} f(s');
        if s > s* then
            s* ← s;
        if s > ŝ then
            ŝ ← s;
        Update the tabu list;
        if End of cycle then
            Update trails of each combination with ŝ;
    until Stopping condition is met;
    return s*
```

s = *current solution*
s' = *neighboring solution of* s
s^* = *best solution from the begining*
$\hat{s}$ = *best solution from the last cycle*

Estimation

D. Schindl and N. Zufferey - Solution Methods for Fuel Supply of Trains. (2013)

Learning Tabu Search

Algorithm 1: Learning Tabu Search (LTS)

```
begin
    s ← initial solution;
    s* ← s;
    repeat
        Estimate the quality of non-tabu neighbors of N(s);
        N_Q ← Q most promising neighbors of N(s) according to the
        diversification policy;
        s ← max_{s'∈N_Q} f(s');
        if s > s* then
            s* ← s;
        if s > ŝ then
            ŝ ← s;
        Update the tabu list;
        if End of cycle then
            Update trails of each combination with ŝ;
    until Stopping condition is met;
    return s*
```

s = *current solution*
s' = *neighboring solution of* s
s^* = *best solution from the begining*
$\hat{s}$ = *best solution from the last cycle*

Estimation

Selection + Diversification

D. Schindl and N. Zufferey - Solution Methods for Fuel Supply of Trains. (2013)

Learning Tabu Search

Algorithm 1: Learning Tabu Search (LTS)

```
begin
    s ← initial solution;
    s* ← s;
    repeat
        Estimate the quality of non-tabu neighbors of N(s);
        N_Q ← Q most promising neighbors of N(s) according to the
        diversification policy;
        s ← max_{s'∈N_Q} f(s');
        if s > s* then
            s* ← s;
        if s > ŝ then
            ŝ ← s;
        Update the tabu list;
        if End of cycle then
            Update trails of each combination with ŝ;
    until Stopping condition is met;
    return s*
```

s = *current solution*
s' = *neighboring solution of* s
s^* = *best solution from the begining*
$\hat{s}$ = *best solution from the last cycle*

Estimation

Selection + Diversification

Update

D. Schindl and N. Zufferey - Solution Methods for Fuel Supply of Trains. (2013)

Learning Tabu Search

Algorithm 1: Learning Tabu Search (LTS)

```
begin
    s ← initial solution;
    s* ← s;
    repeat
        Estimate the quality of non-tabu neighbors of N(s);
        N_Q ← Q most promising neighbors of N(s) according to the
        diversification policy;
        s ← max_{s'∈N_Q} f(s');
        if s > s* then
            s* ← s;
        if s > ŝ then
            ŝ ← s;
        Update the tabu list;
        if End of cycle then
            Update trails of each combination with ŝ;
    until Stopping condition is met;
    return s*
```

s = *current solution*
s' = *neighboring solution of* s
s^* = *best solution from the begining*
$\hat{s}$ = *best solution from the last cycle*

Tabu Search

Estimation

Selection + Diversification

Update

D. Schindl and N. Zufferey - Solution Methods for Fuel Supply of Trains. (2013)

LTS – Definition of trail and Update procedure

DEFINITION

$$Trail = \begin{bmatrix} t_{11} & \cdots & t_{1n} \\ \vdots & \ddots & \vdots \\ t_{n1} & \cdots & t_{nn} \end{bmatrix}$$

where t_{ij} indicates if the combination of features t_i and t_j is promising

LTS – Definition of trail and Update procedure

DEFINITION

$$Trail = \begin{bmatrix} t_{11} & \cdots & t_{1n} \\ \vdots & \ddots & \vdots \\ t_{n1} & \cdots & t_{nn} \end{bmatrix}$$

where t_{ij} indicates if the combination of features i and j is promising

UPDATE PROCEDURE

Each $cycle$ (*i.e.* X iterations before updating)

$$t_{ij} = \rho * t_{ij} + \Delta t_{ij}$$

Reinforcement
if a_i and a_j are present in $\hat{s}$

Evaporation rate ($\rho \in [0,1]$)

LTS – Estimation and Selection of solutions

ESTIMATION

$$Tr(s, a_i) = \sum_j tr(a_i, a_j) \quad where\ a_j = 1\ in\ s$$

ADD (a_i = 0 → 1) : the higher $Tr(s, a_i)$, the more information a_i brings to solution s

DROP (a_i = 1 → 0) : the lower $Tr(s, a_i)$, the less information a_i removes to solution s

LTS – Estimation and Selection of solutions

ESTIMATION

$$Tr(s, a_i) = \sum_j tr(a_i, a_j) \quad where\ a_j = 1\ in\ s$$

ADD (a_i = 0 → 1) : the higher $Tr(s, a_i)$, the more information a_i brings to solution s

DROP (a_i = 1 → 0) : the lower $Tr(s, a_i)$, the less information a_i removes to solution s

SELECTION

ADD (a_i = 0 → 1) : chooses the q neigbors with highest estimations (A_q)

DROP (a_i = 1 → 0) : chooses the q neigbors with lowest estimations (D_q)

→ Only $2q$ neighbors are evaluated ($A_q \cup D_q$)

LTS - Diversification

DIVERSIFICATION

INTENSIFICATION **VS** DIVERSIFICATION

LTS - Diversification

DIVERSIFICATION

INTENSIFICATION **VS** DIVERSIFICATION

TRAIL

$A_q =$ q highest estimated neighbors

$D_q =$ q lowest estimated neighbors

LTS - Diversification

DIVERSIFICATION

INTENSIFICATION

↓

TRAIL

↓

$A_q =$ q highest estimated neighbors

$D_q =$ q lowest estimated neighbors

VS

DIVERSIFICATION

↓

TRAIL

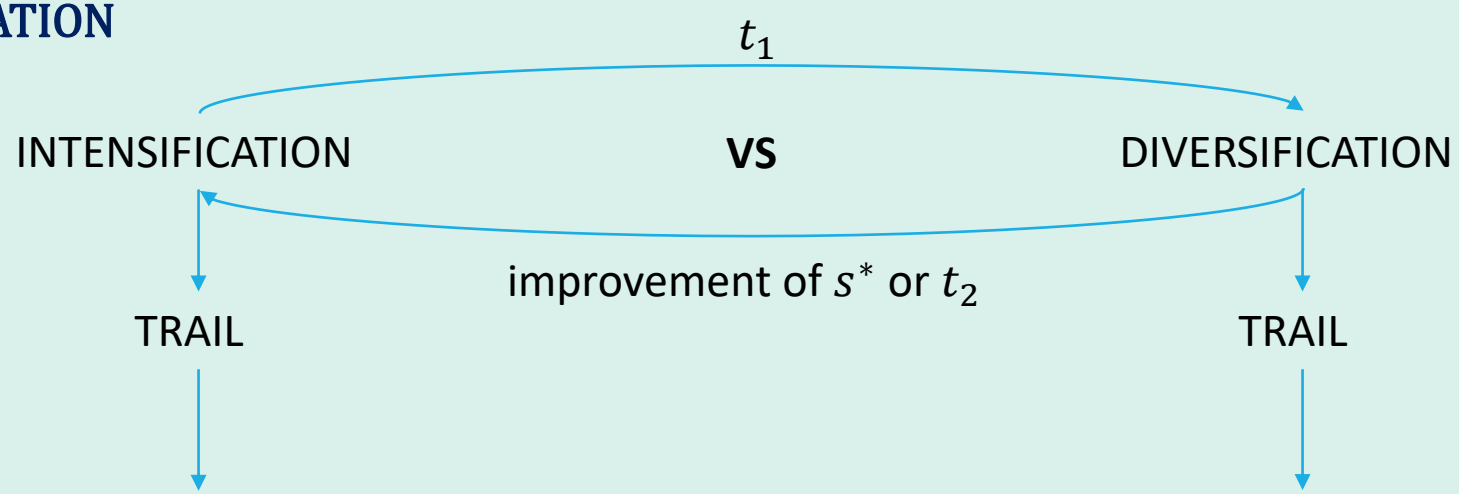
↓

$A_q =$ q lowest estimated neighbors

$D_q =$ q highest estimated neighbors

LTS - Diversification

DIVERSIFICATION

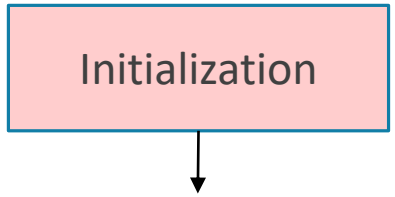


$A_q =$ q highest estimated neighbors
$D_q =$ q lowest estimated neighbors

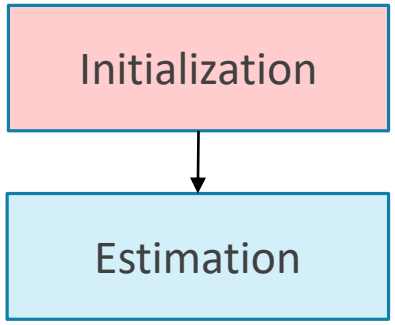
$A_q =$ q lowest estimated neighbors
$D_q =$ q highest estimated neighbors

t_1 : Number of iterations without improving s^*
t_2 : Number of iterations with diversification

Summary of LTS method



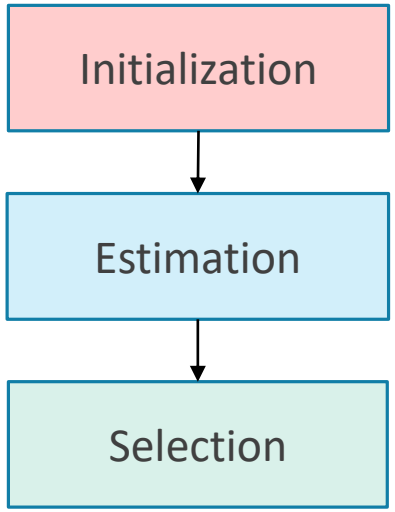
Summary of LTS method



A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
0	1	1	0	0	0	1	1	0	1

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	Estimation
A1	-	**28,9**	**35,9**	8,3	2,3	9,6	**90,6**	**30,9**	40,9	**28,9**	215,2
A2	28,9	-	**45,9**	95,6	7,5	28,9	**3,5**	**24,9**	53,3	**45,9**	138,2
A3	35,9	**45,9**	-	12,6	9,8	36,9	**27,9**	**66,9**	6,8	**26,7**	167,4
A4	8,3	**7,5**	**9,8**	-	10,9	40,9	**75,9**	**45,9**	4,9	**53,9**	283,9
A5	2,3	**7,5**	**9,8**	10,9	-	67,6	**90,6**	**54,1**	86,6	**4,6**	166,6
A6	9,6	**28,9**	**36,9**	40,9	67,6	-	**100,6**	**23,6**	74,9	**70,4**	260,4
A7	90,6	**3,5**	**27,9**	75,9	90,6	100,6	-	**32,3**	89,6	**85,6**	149,3
A8	30,9	**24,9**	**66,9**	45,9	54,1	23,6	**32,3**	-	85,6	**37,6**	161,7
A9	40,9	**53,3**	**6,8**	4,9	86,6	74,9	**89,6**	**85,6**	-	**35,4**	270,7
A10	28,9	**45,9**	**26,7**	53,9	4,6	70,4	**85,6**	**37,6**	35,4	-	195,8

Summary of LTS method



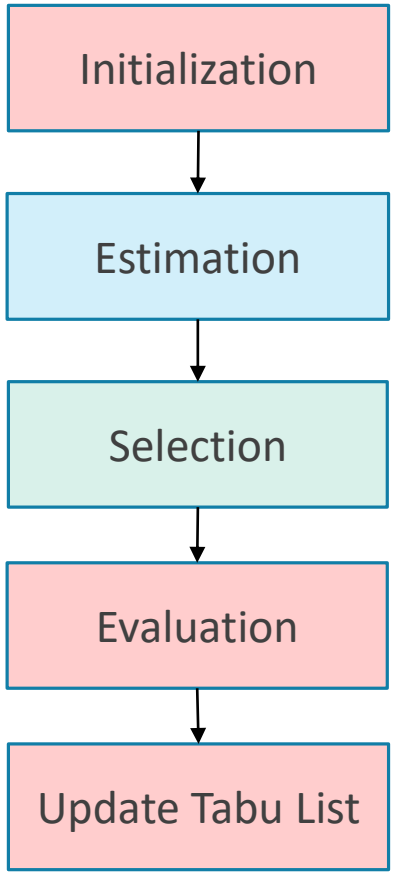
A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
0	1	1	0	0	0	1	1	0	1

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	Estimation
A1	-	**28,9**	**35,9**	8,3	2,3	9,6	**90,6**	**30,9**	40,9	**28,9**	215,2
A2	28,9	-	**45,9**	95,6	7,5	28,9	**3,5**	**24,9**	53,3	**45,9**	138,2
A3	35,9	**45,9**	-	12,6	9,8	36,9	**27,9**	**66,9**	6,8	**26,7**	167,4
A4	8,3	**7,5**	**9,8**	-	10,9	40,9	**75,9**	**45,9**	4,9	**53,9**	283,9
A5	2,3	**7,5**	**9,8**	10,9	-	67,6	**90,6**	**54,1**	86,6	**4,6**	166,6
A6	9,6	**28,9**	**36,9**	40,9	67,6	-	**100,6**	**23,6**	74,9	**70,4**	260,4
A7	90,6	**3,5**	**27,9**	75,9	90,6	100,6	-	**32,3**	89,6	**85,6**	149,3
A8	30,9	**24,9**	**66,9**	45,9	54,1	23,6	**32,3**	-	85,6	**37,6**	161,7
A9	40,9	**53,3**	**6,8**	4,9	86,6	74,9	**89,6**	**85,6**	-	**35,4**	270,7
A10	28,9	**45,9**	**26,7**	53,9	4,6	70,4	**85,6**	**37,6**	35,4	-	195,8

A : A4 > A9 > A6 > A1 > A5
D : A2 < A7 < A8 < A3 < A10

q = 2 : $A_q \cup D_q$
Intensification : A4 A9 A2 A7
Diversification : A1 A5 A3 A10

Summary of LTS method



A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
0	1	1	0	0	0	1	1	0	1

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	Estimation
A1	-	**28,9**	**35,9**	8,3	2,3	9,6	**90,6**	**30,9**	40,9	**28,9**	215,2
A2	28,9	-	**45,9**	95,6	7,5	28,9	**3,5**	**24,9**	53,3	**45,9**	138,2
A3	35,9	**45,9**	-	12,6	9,8	36,9	**27,9**	**66,9**	6,8	**26,7**	167,4
A4	8,3	**7,5**	**9,8**	-	10,9	40,9	**75,9**	**45,9**	4,9	**53,9**	283,9
A5	2,3	**7,5**	**9,8**	10,9	-	67,6	**90,6**	**54,1**	86,6	**4,6**	166,6
A6	9,6	**28,9**	**36,9**	40,9	67,6	-	**100,6**	**23,6**	74,9	**70,4**	260,4
A7	90,6	**3,5**	**27,9**	75,9	90,6	100,6	-	**32,3**	89,6	**85,6**	149,3
A8	30,9	**24,9**	**66,9**	45,9	54,1	23,6	**32,3**	-	85,6	**37,6**	161,7
A9	40,9	**53,3**	**6,8**	4,9	86,6	74,9	**89,6**	**85,6**	-	**35,4**	270,7
A10	28,9	**45,9**	**26,7**	53,9	4,6	70,4	**85,6**	**37,6**	35,4	-	195,8

A : A4 > A9 > A6 > A1 > A5

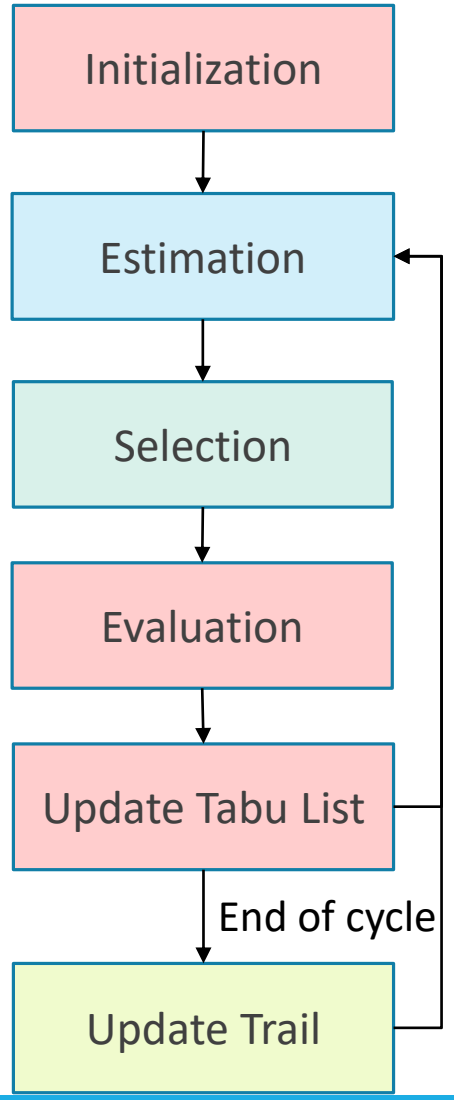
D : A2 < A7 < A8 < A3 < A10

q = 2 : $A_q \cup D_q$

Intensification : A4 A9 A2 A7

Diversification : A1 A5 A3 A10

Summary of LTS method



A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
0	1	1	0	0	0	1	1	0	1

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	Estimation
A1	-	**28,9**	**35,9**	8,3	2,3	9,6	**90,6**	**30,9**	40,9	**28,9**	215,2
A2	28,9	-	**45,9**	95,6	7,5	28,9	**3,5**	**24,9**	53,3	**45,9**	138,2
A3	35,9	**45,9**	-	12,6	9,8	36,9	**27,9**	**66,9**	6,8	**26,7**	167,4
A4	8,3	**7,5**	**9,8**	-	10,9	40,9	**75,9**	**45,9**	4,9	**53,9**	283,9
A5	2,3	**7,5**	**9,8**	10,9	-	67,6	**90,6**	**54,1**	86,6	**4,6**	166,6
A6	9,6	**28,9**	**36,9**	40,9	67,6	-	**100,6**	**23,6**	74,9	**70,4**	260,4
A7	90,6	**3,5**	**27,9**	75,9	90,6	100,6	-	**32,3**	89,6	**85,6**	149,3
A8	30,9	**24,9**	**66,9**	45,9	54,1	23,6	**32,3**	-	85,6	**37,6**	161,7
A9	40,9	**53,3**	**6,8**	4,9	86,6	74,9	**89,6**	**85,6**	-	**35,4**	270,7
A10	28,9	**45,9**	**26,7**	53,9	4,6	70,4	**85,6**	**37,6**	35,4	-	195,8

A : A4 > A9 > A6 > A1 > A5

D : A2 < A7 < A8 < A3 < A10

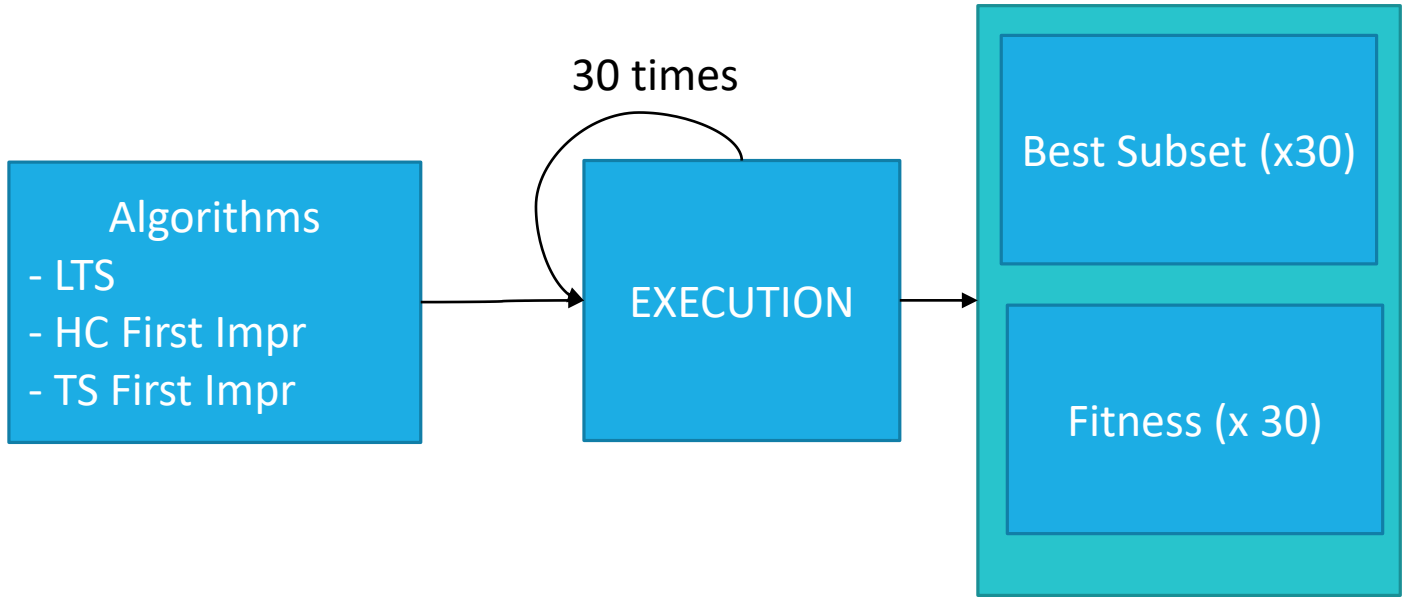
q = 2 : $A_q \cup D_q$

Intensification : A4 A9 A2 A7

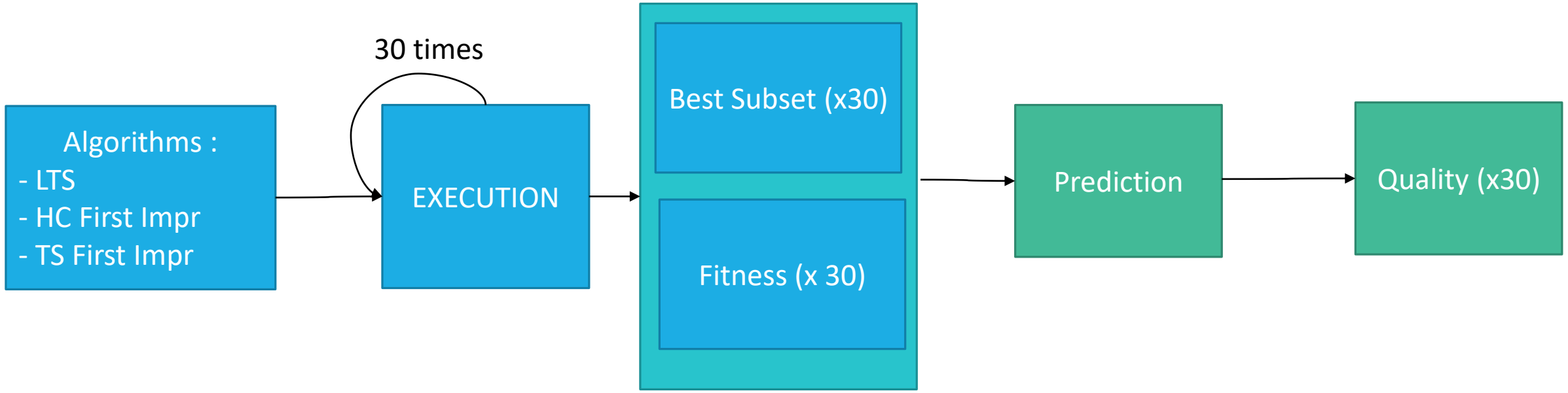
Diversification : A1 A5 A3 A10

Experiments

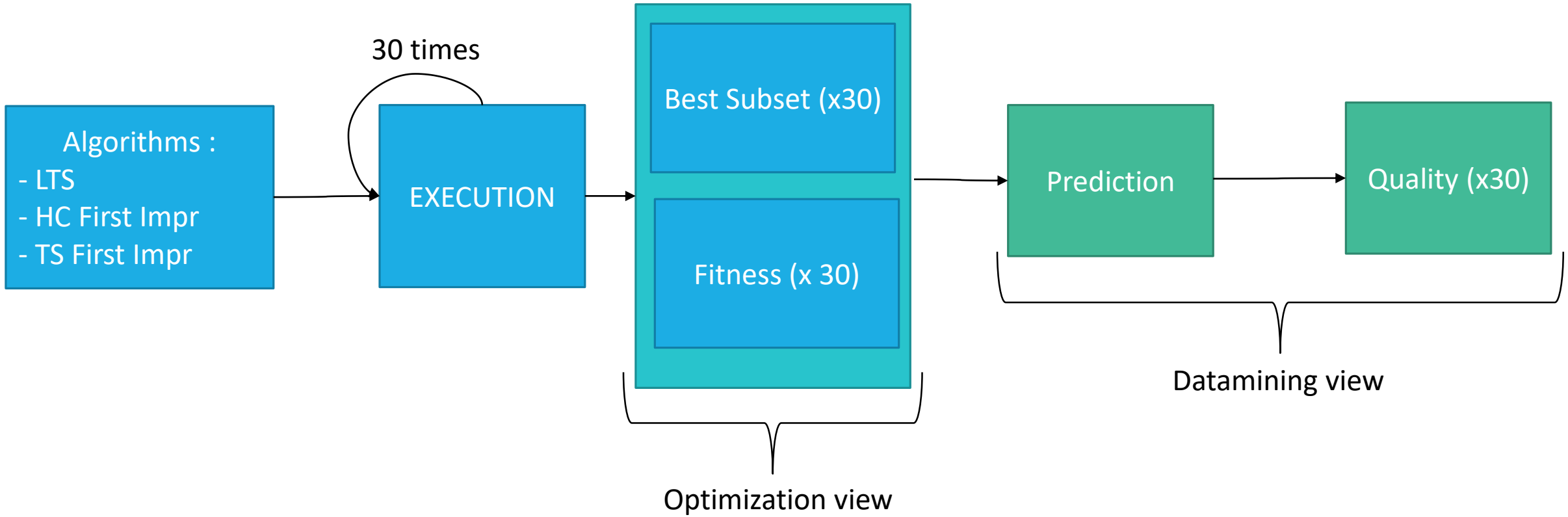
Experimental protocol



Experimental protocol



Experimental protocol

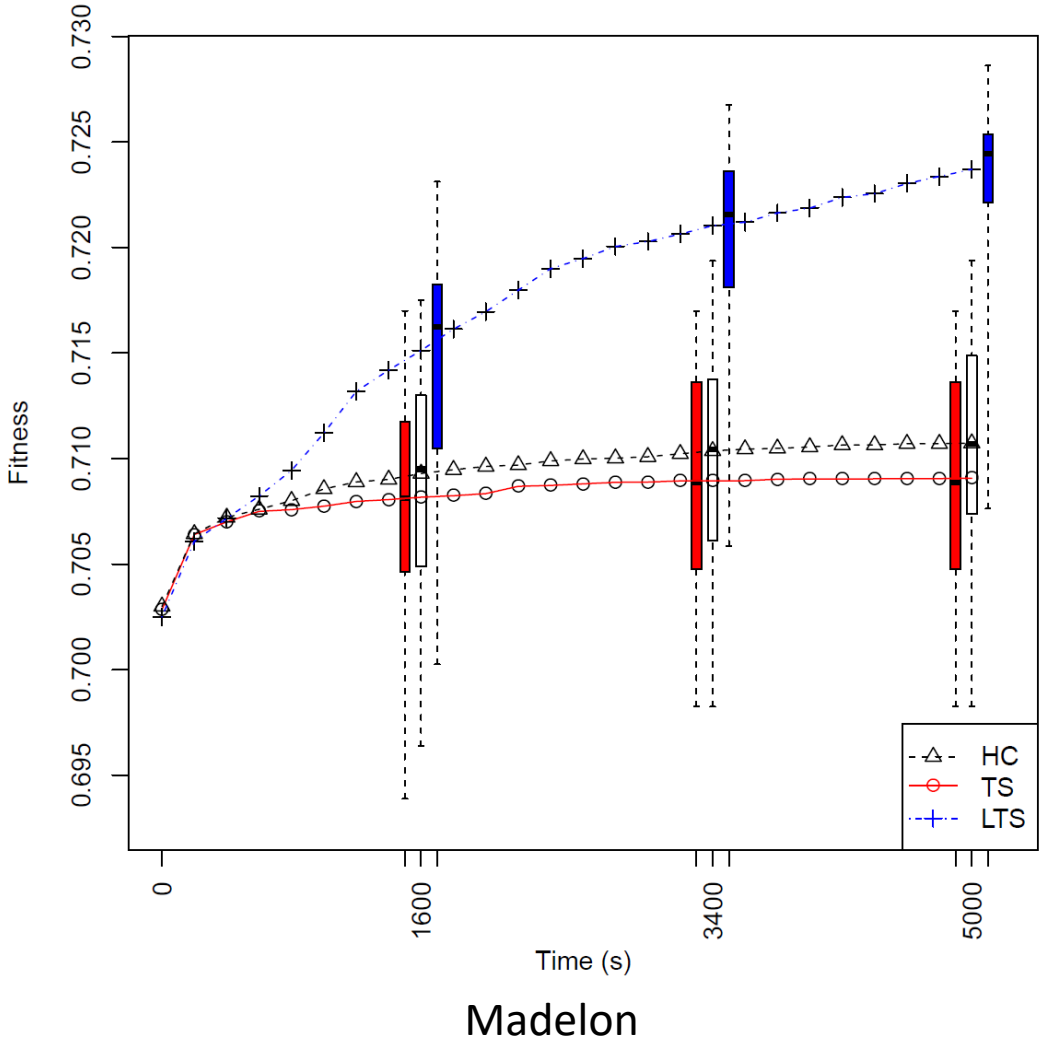


Wilcoxon tests are performed on each part

Description of instances

Name	# Features	Size of training set	Size of validation set	SVM Runtime (sec)	Allocated Runtime (sec)
Schizophrenia	410	56	30	0,001	500
Colon	2000	62	32	0,052	120
Breast	24481	78	26	0,734	500
Arcene	10000	100	100	1,123	3000
DNA	180	1400	600	1,72	500
Madelon	500	2000	600	38,089	5000

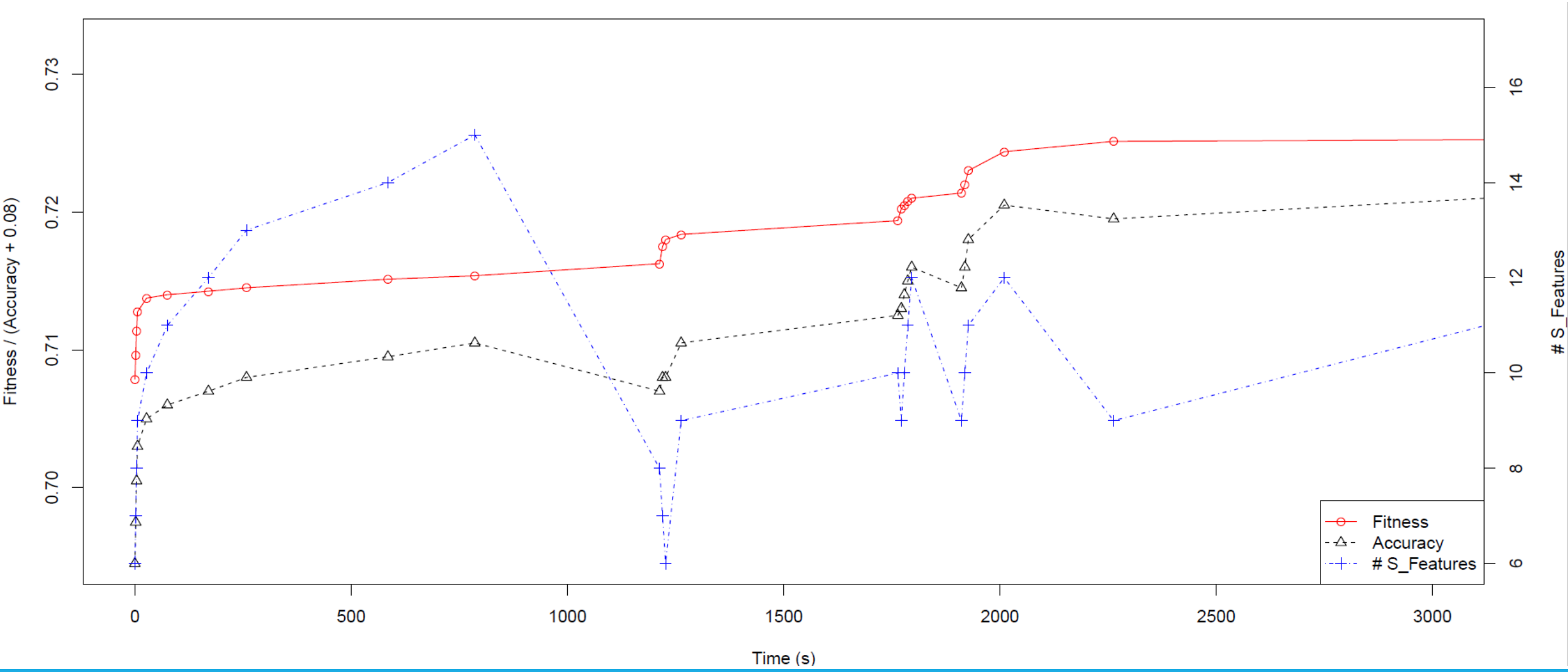
Optimization method



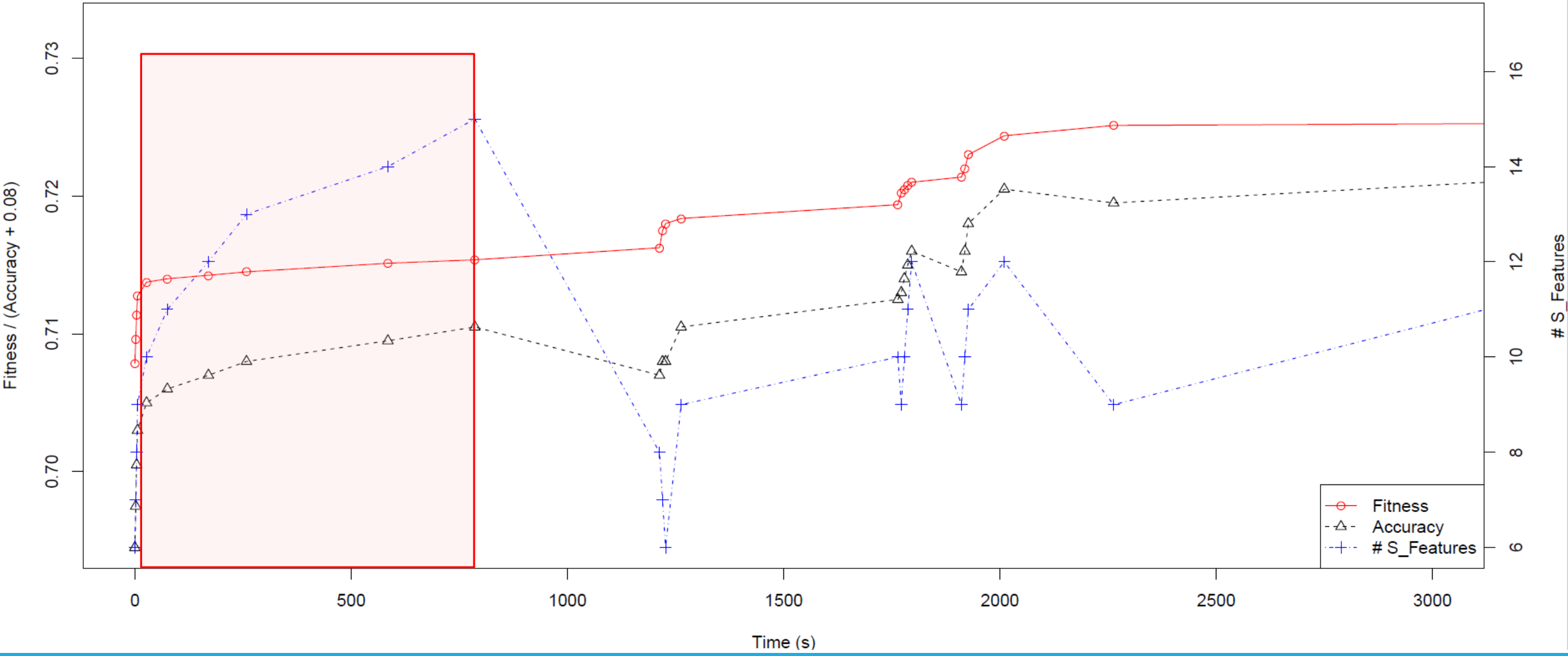
Madelon

Name	Wilcoxon tests
Schizophrenia	**LTS** > HC > TS
Colon	**(LTS = HC)** > TS
Breast	HC > (LTS = TS)
Arcene	**LTS** > HC > TS
DNA	**LTS** > (HC = TS)
Madelon	**LTS** > (HC = TS)

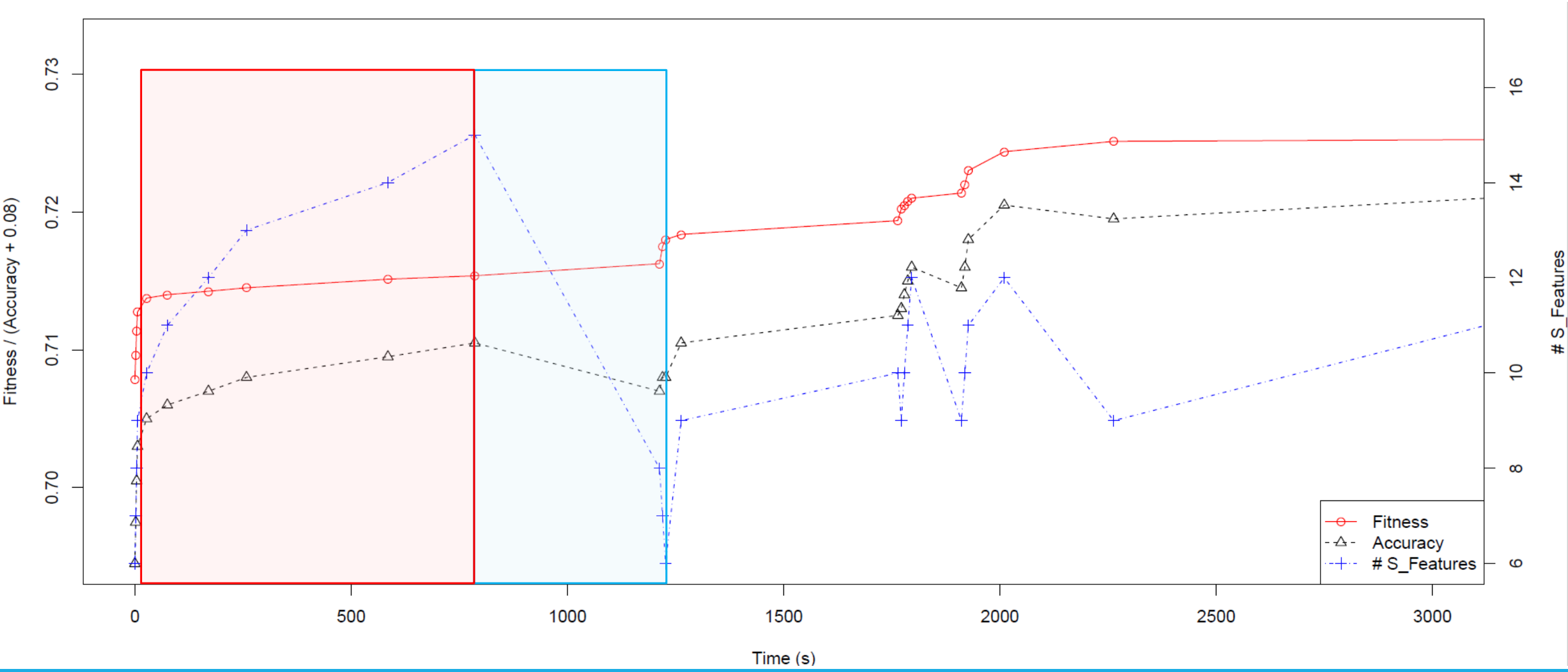
Behavior of LTS on Madelon



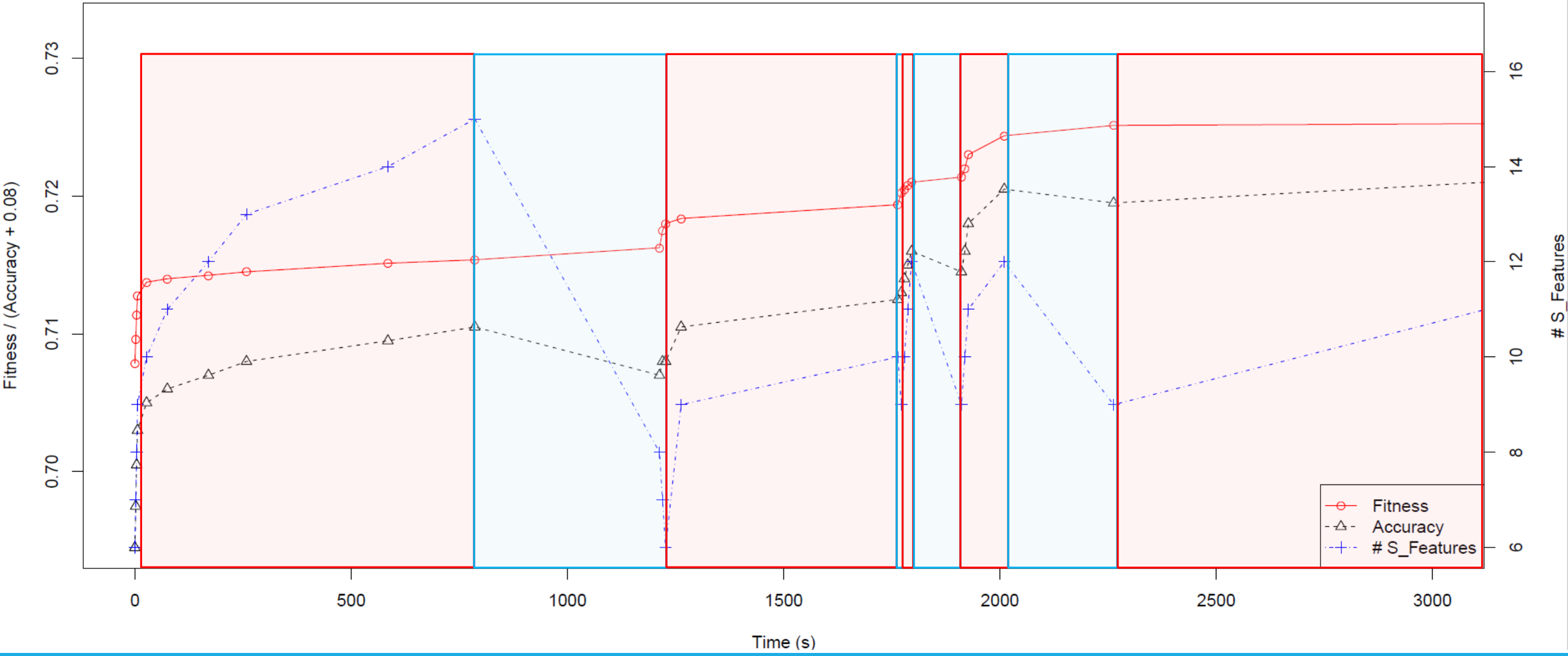
Behavior of LTS



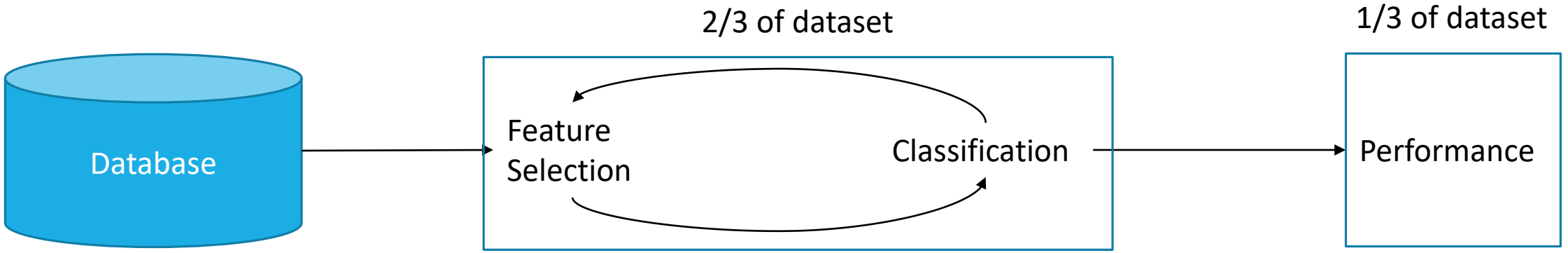
Behavior of LTS



Behavior of LTS



Datamining approach



Dataset: 2 parts

2/3 of dataset = Training
1/3 of dataset = Validation

Name	Wilcoxon
Schizophrenia	LTS > HC > TS
Colon	(LTS = HC) > TS
Breast	LTS = HC = TS
Arcene	LTS > HC > TS
DNA	LTS = HC = TS
Madelon	LTS > (HC = TS)

Conclusion and perspectives

CONCLUSION

Optimization point of view

- LTS improves faster than the other methods
- LTS is always better than TS

Datamining point of view

- LTS is always better than TS

PERSPECTIVES

- Proposing different definitions of trail
- Integrating the learning mechanism into other metaheuristics

Questions ?