

APS-VSS: Accelerated Pattern Search with Variable Solution Size for Simultaneous Instance Selection and Generation

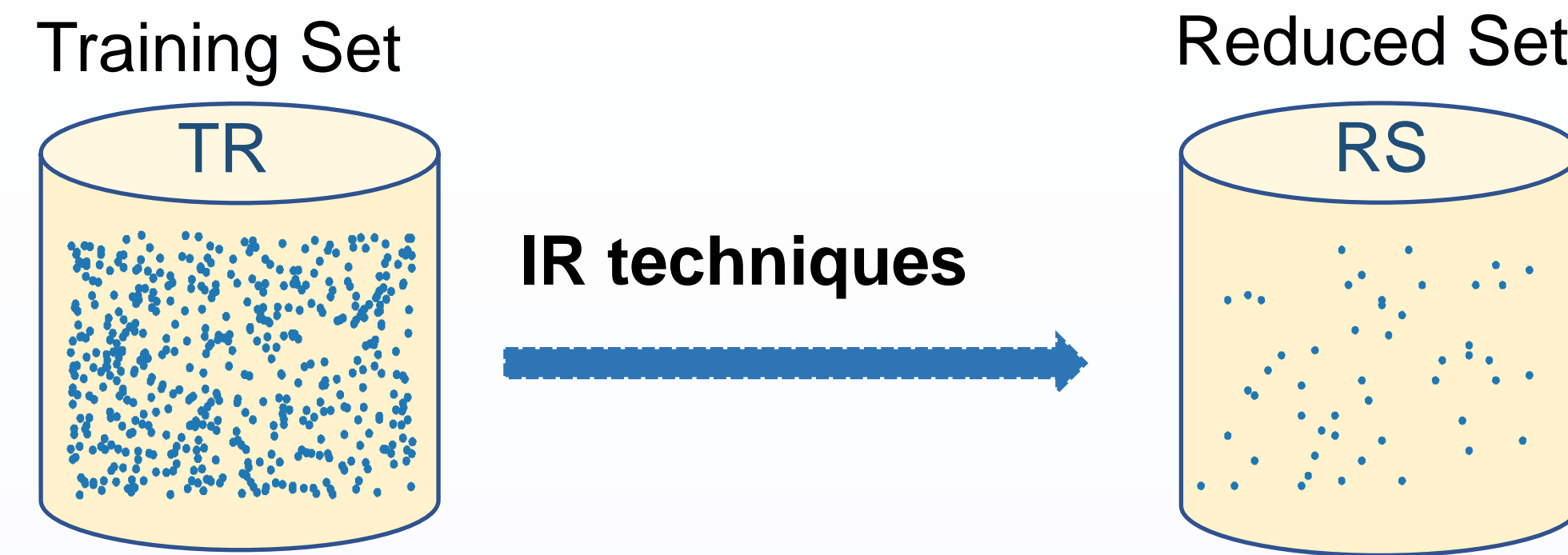
Hoang Lam Le^{1,*}, Ferrante Neri², Dario Landa-Silva¹, Isaac Triguero^{1,3}

¹COL, School of Computer Science, University of Nottingham, UK
²NICE, Department of Computer Science, University of Surrey, UK
³DaSCI Andalusian Institute in Data Science and Computational Intelligence, University of Granada, Spain
 *Email: hoang.le@nottingham.ac.uk
lehoanglam2000@gmail.com



Introduction

With the explosion in the size of training set (TR), potentially having more valuable information but also more noise and imperfections. Data reduction techniques including feature selection, instance reduction (IR), and discretisation are important for a data mining process.



Research about IR can be categorised into instance selection (IS) and instance generation (IG). IS chooses representative examples in the available source while IG creates artificial ones, if needed. IS has frequently been modelled as a binary combinatorial optimisation problem as it deals with the decision whether or not to include a sample in the final subset, whilst IG can be modelled as a continuous optimisation problem, considering generating new examples non-existing in the source but better to represent TR.

Benefits of RS over TR

- Cleaner and smaller
- Freer of noise, redundant or irrelevant samples (the so-called **Smart Data**)
- Green AI, sustainable AI

Challenges

State-of-the-art IR solutions are based on evolutionary search methods, which are time-consuming due to:

- High fitness evaluation cost
→ **Surrogate model** [1]
- Algorithmic design complexity
→ **Single-Point Memetic Structure** [2]

Single-Point Search

Instance I has m features and belongs to class w :

$$I = a_1, a_2, \dots, a_m$$

$$\text{TR} = \begin{bmatrix} f_1 & f_2 & \dots & f_m \\ I_1 & a_{11} & a_{12} & \dots & a_{1m} \\ I_2 & a_{21} & a_{22} & \dots & a_{2m} \\ \dots & \dots & \dots & \dots & \dots \\ I_p & a_{p1} & a_{p2} & \dots & a_{pm} \end{bmatrix} \quad \text{with } p \ll l$$

$$\text{RS} = \begin{bmatrix} f_1 & f_2 & \dots & f_m \\ I_1 & b_{11} & b_{12} & \dots & b_{1m} \\ I_2 & b_{21} & b_{22} & \dots & b_{2m} \\ \dots & \dots & \dots & \dots & \dots \\ I_p & b_{p1} & b_{p2} & \dots & b_{pm} \end{bmatrix}$$

Flatten RS into a n -dimensional vector, $n = m * p$:

$$x = (b_{11}, b_{12}, \dots, b_{1m}, b_{21}, b_{22}, \dots, b_{2m}, \dots, b_{p1}, b_{p2}, \dots, b_{pm})$$

e^i is an n -dimensional vector with all zeros, but 1 at the i th element

$$e^i = (0, 0, \dots, 1, \dots, 0, 0)$$

$i = 1$: indicating the first variable of x : b_{11}

$$x^1 = x - \rho \cdot e^1 \quad \text{First Attempt: Move one exploratory step}$$

$$x = (b_{11}, b_{12}, \dots, b_{1m}, b_{21}, b_{22}, \dots, b_{2m}, \dots, b_{p1}, b_{p2}, \dots, b_{pm})$$

$$x^1 = x + \frac{\rho}{2} \cdot e^1 \quad \text{Second Attempt: Move a half-size exploratory step in the other direction}$$

Acceleration of Fitness Computation

- Accuracy ← considering RS as training data to classify TR as the test set
- Maintains a global distance matrix D : length = size (TR), width = size (RS)
- D can be initialised large (10% size (TR)), but is gradually reduced and remains small (1%-3% size (TR))
- Tailored to the k -nearest neighbour rule and the logic of pattern search

		1	2	3	-	p
Distance matrix	1	0.55	0.12	0.85		1.2
	2					
	3					
	-					
	1	0.21	1.02	3.2		0.98

At each trial of x , only recomputing values of one column, thus saving $l \times (p - 1)$ times of Euclidean distance calculation [2]

Pseudo-code of APS-VSS

```

1: INPUT  $x$ 
2: while local budget and precision conditions are not met do
3:   for  $h = 1 : p$  do
4:      $x_h = x$  after removing the elements  $b_{h1}, b_{h2}, \dots, b_{hm}$ 
5:     if  $f(x_h) \geq f(x)$  then
6:        $x = x_h$ 
7:     end if
8:   end for
9:   for  $i = 1 : n$  ( $n = m \cdot p$ ) do
10:     $x^t = x - \rho \cdot e^i$ 
11:    if  $f(x^t) \geq f(x)$  then
12:       $x = x^t$ 
13:    else
14:       $x^t = x + \frac{\rho}{2} \cdot e^i$ 
15:      if  $f(x^t) \geq f(x)$  then
16:         $x = x^t$ 
17:      end if
18:    end if
19:    if mod( $i, m$ ) = 0 then
20:       $j = i / m$  ▷ Get index of the generated example
21:       $x^t = x^t$  after removing the elements  $b_{j1}, b_{j2}, \dots, b_{jm}$ 
22:      if  $f(x^t) \geq f(x^t)$  then
23:         $x = x^t$ 
24:      end if
25:    end if
26:  end for
27:  if  $x$  has not been updated then
28:    halve the exploratory radius  $\rho$ 
29:    if  $\rho < \epsilon$  then
30:      Randomly generate a candidate solution  $x^r$ 
31:      Apply Crossover between  $x$  and  $x^r$  to generate  $x^t$ 
32:      Reinitialise  $x = x^t$ 
33:    end if
34:  end if
35: end while
36: RETURN  $x$ 
  
```

LS^{eli}: Local search in combinatorial space

LS^{cont}: Local search in continuous space

LS^{asc}: Local search in combinatorial space

Restart the search or halve the exploratory radius

Motivation

State-of-the-art IR techniques employed IS and IG sequentially, usually IS first and then IG. Typically, IS searches for the best distribution of instances per class to feed in IG for further optimisation. Unlike previous studies, **Accelerated Pattern Search with Variable Solution Size (APS-VSS)** performs the selection and generation on both continuous and combinatorial search spaces within a single framework.

Algorithmic Description

An iteration of APS-VSS is summarised as follows:

- LS^{eli} shrinks the initial RS, discarding any element whose absence does not deteriorate the solution quality
- LS^{cont} perturbs features and seeks an accurate solution
- LS^{asc} is embedded within LS^{cont} and confirms whether the presence of the newly generated instance is necessary.
- The crossover re-initialises the candidate solution to explore another search region when the LS^{cont} seems to be no longer effective.

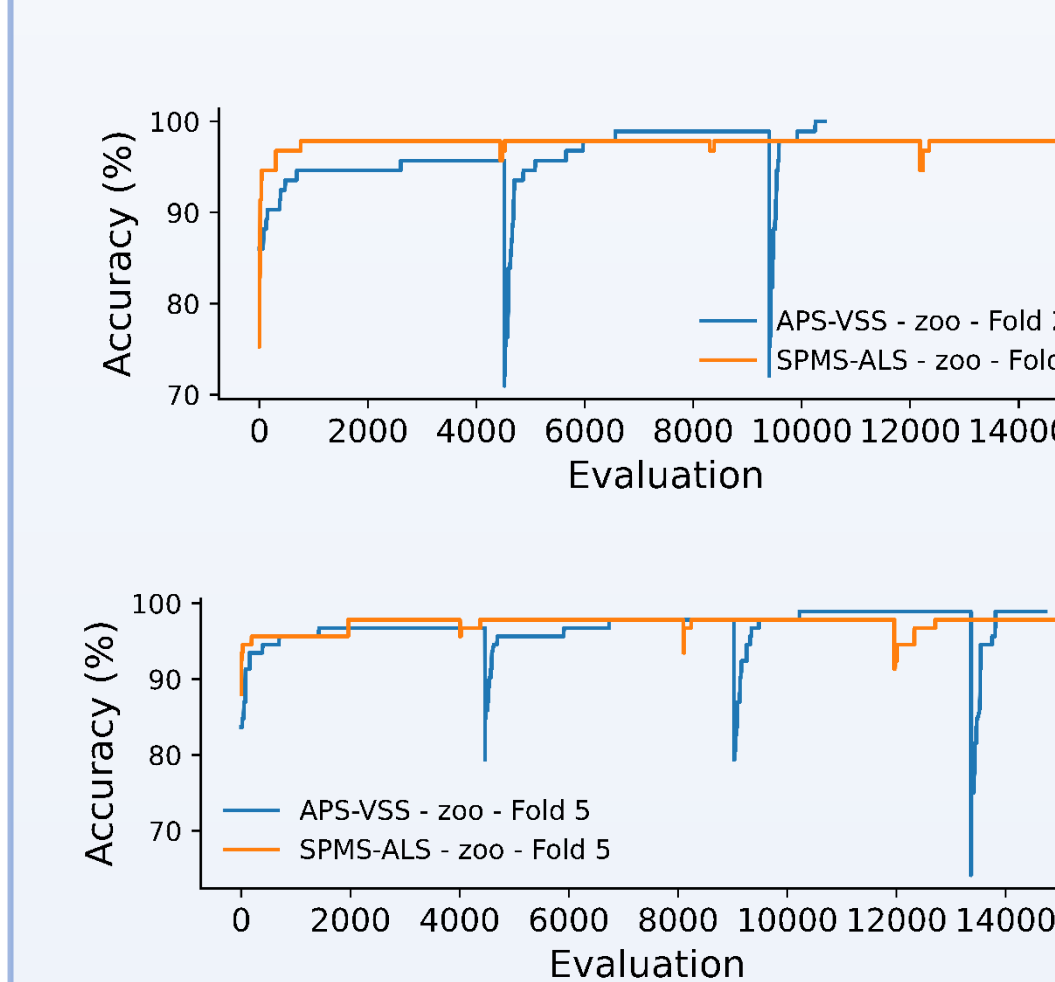
Parameters of APS-VSS

$P_{init} = 10\%$ size (TR)
 exploratory step $\rho = 0.4$

Compared Algorithms [2]

- LSIR
- SPMS-ALS
- APS-VSS
- SSMA-LSHADE
- SSMA-SFLSDE
- SSMA-SPMS-ALS

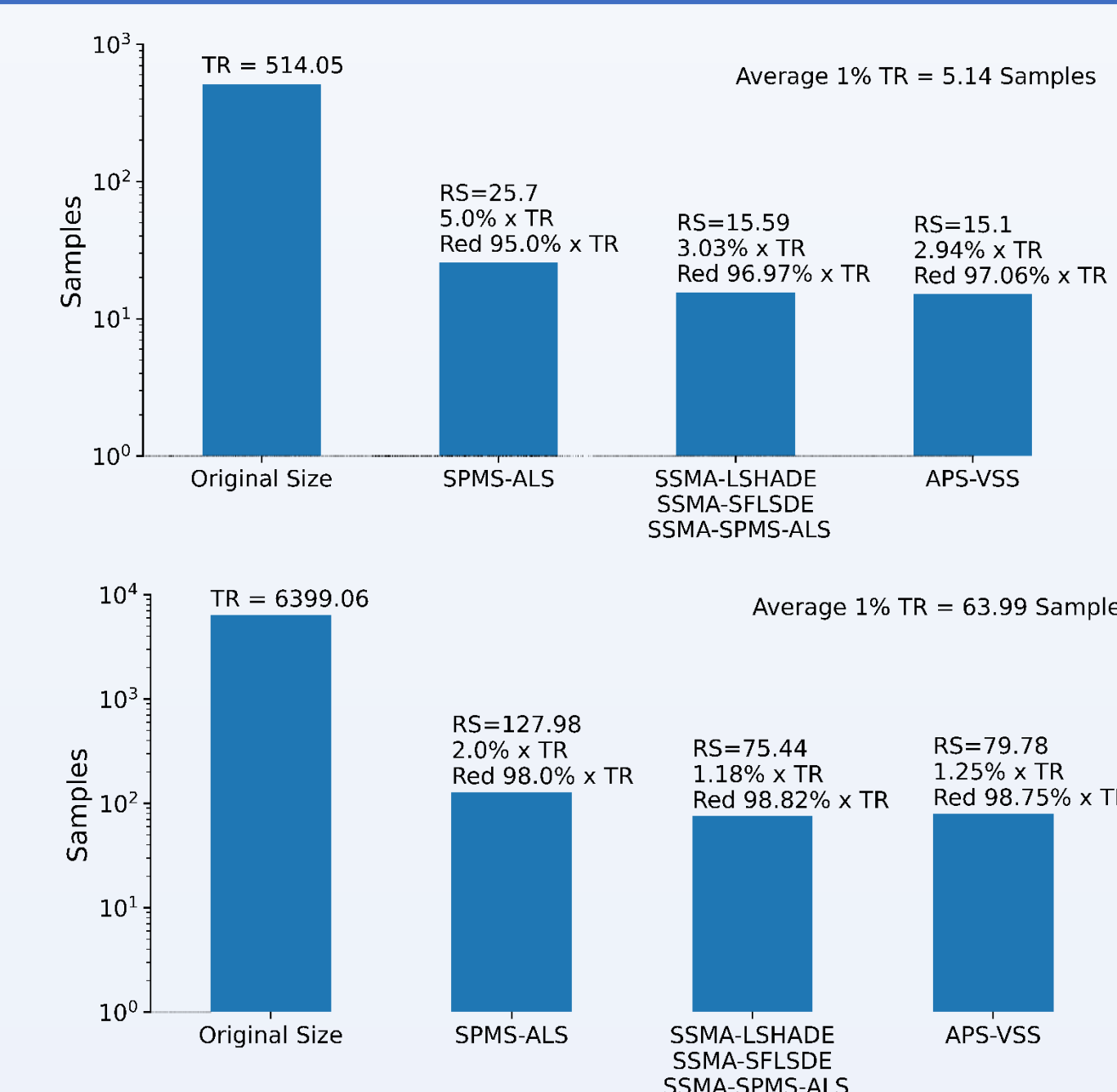
Search Behavior



- Restart mechanism is effective to prevent premature convergence
- APS-VSS gradually develops the accuracy whilst SPMS-ALS goes back to its previous peak. This can be attributed to the impact of LS^{eli} and LS^{asc}

Reduction Rate

Small datasets: Top
 Medium datasets: Bottom

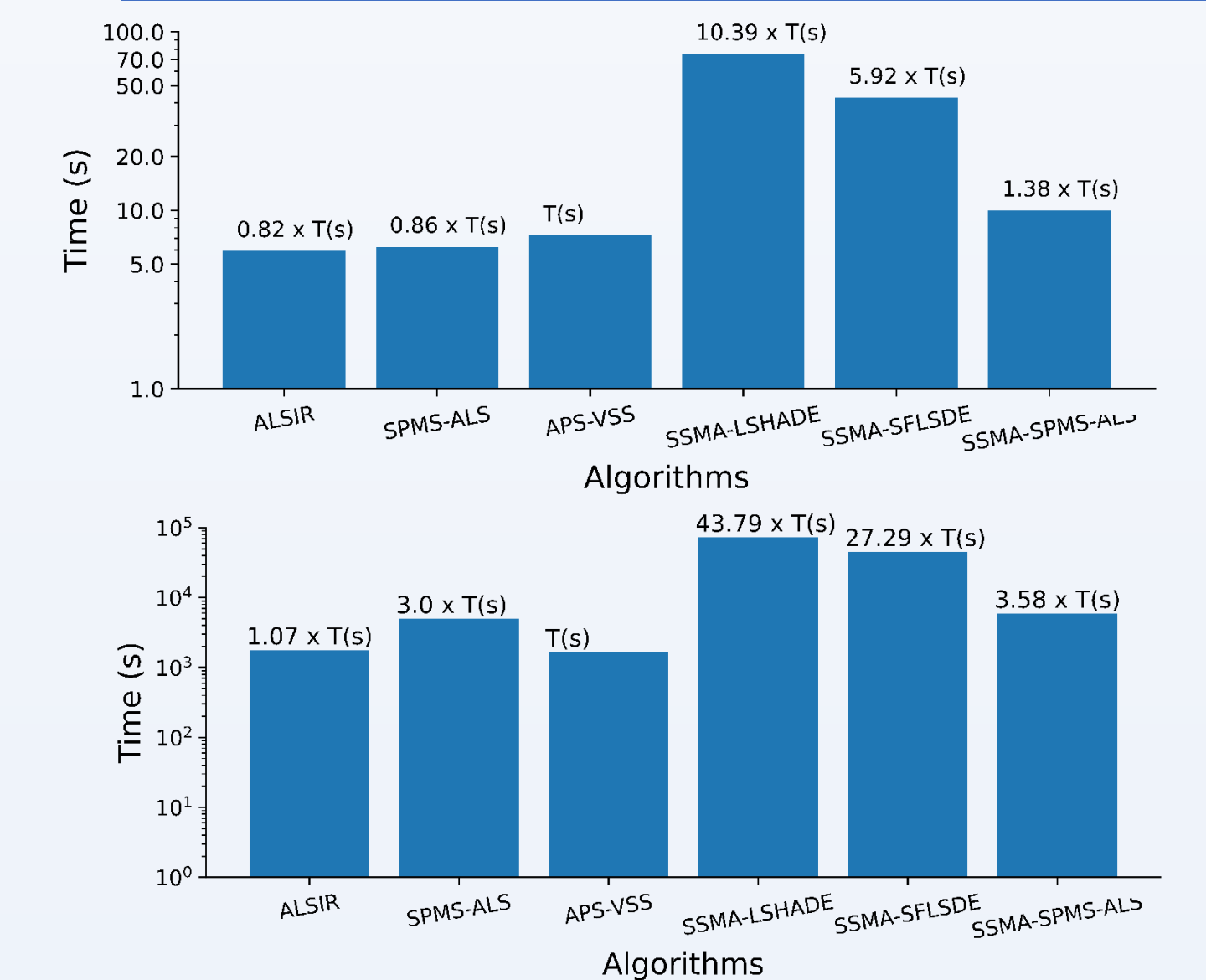


For small and medium, respectively:

- SPMS-ALS: 95% and 98%
- Hybrid approaches: 96.97% and 98.82%
- APS-VSS: 97.06% and 98.75%

Runtime

Small datasets: Top
 Medium datasets: Bottom



For small and medium, respectively:

- LSIR: 6s and 1784s
- SPMS-ALS: 6s and 5007s
- APS-VSS: 7s and 1669s
- SSMA-LSHADE: 75s and 73072s
- SSMA-SFLSDE: 43s and 45547s
- SSMA-SPMS-ALS: 10s and 5969s

References

- [1] Le, H. L., Landa-Silva D., Mikel G., Salvador G., Triguero I. 'EUSC: A Clustering-based Surrogate Model to Accelerate Evolutionary Undersampling in Imbalanced Classification.' *Applied Soft Computing* 101 (2021):107033.
 [2] Le, H. L., Neri F., Triguero I. 'SPMS-ALS: A Single-Point Memetic Structure with Accelerated Local Search for Instance Reduction.' *Swarm and Evolutionary Computation* 69 (2022): 100991.