Table A1. List of 25 most useful inputs identified using the PC IVS algorithm for the Bow and Don River watersheds, selected form the set of candidate inputs. Input variables are encoded in the following format "station ID"_"variable"_"statistic"_"lagged timesteps". Variable abbreviations "WL" and "Precip" refer to water level and precipitation.

rank	Bow	Don
1	BH004 WL Mean L4	HY022 WL Mean L4
2	BB001 WL Max L4	HY008 Precip Sum L4
3	BB001 WL Min L12	HY019 WL Mean L4
4	BH004 WL Mean L5	HY008 Precip Sum L5
5	Calgary Temp Max L4	HY027 Precip Sum L4
6	BB001 WL Max L6	HY017 WL Mean L4
7	BH004 WL Mean L15	HY022 WL Mean L5
8	Calgary Precip Sum L5	HY008 Precip Sum L8
9	Calgary Temp Min L10	HY027 Precip Sum L6
10	Calgary Precip Sum L11	HY017 WL Mean L5
11	BH004 WL Max L4	HY027 Precip Sum L5
12	BH004 WL Min L4	HY008 Precip Sum L10
13	BH004 WL Max L7	HY019 WL Mean L7
14	Calgary Precip Sum L7	HY080 WL Mean L4
15	BB001 WL Min L15	HY008 Precip Sum L11
16	BH004 WL Min L8	HY008 Precip Sum L6
17	Calgary Precip Sum L10	HY080 WL Mean L6
18	BH004 WL Max L12	HY027 Precip Sum L7
19	Calgary Precip Sum L6	HY022 WL Mean L6
20	BB001 WL Max L5	HY027 Precip Sum L8
21	Calgary Temp Min L15	HY022 WL Mean L7
22	BH004 WL Min L6	HY080 WL Mean L5
23	BH004 WL Mean L6	HY017 WL Mean L6
24	BH004 WL Max L5	HY080 WL Mean L7
25	BB001 WL Min L9	HY019 WL Mean L6

Appendix B: Pseudocode

Algorithm 1 Random undersampling

Require:

Set S containing X input features and Y observations, $(x_1, y_1), ..., (x_m, y_m)$

High stage threshold, θ

$S_{TS} = S$ where $Y < \phi_{TS}$

$S_{HS} = S$ where $Y \geq \phi_{HS}$

$S'_{TS} \leftarrow sample(S_{TS}, N_{HS})$

$S'_{HS} \leftarrow sample(S_{HS}, N_{HS})$

$S' = S'_{TS} \bigcup S'_{HS}$

Algorithm 2 Random oversampling

Require:

Set S containing X input features and Y observations, $(x_1, y_1), ..., (x_m, y_m)$

High stage threshold, θ

$S_{TS} = S$ where $Y < \phi_{TS}$

$S_{HS} = S$ where $Y \geq \phi_{HS}$

$S'_{TS} \leftarrow sample(S_{TS}, N_{TS})$

$S'_{HS} \leftarrow sample(S_{HS}, N_{TS})$

$S' = S'_{TS} \bigcup S'_{HS}$

Algorithm 3 SMOTER

Require:

Set S containing X input features and Y observations, $(x_1, y_1), ..., (x_m, y_m)$

High stage threshold, θ_{HS}

Ensure:

$\phi_{HS}/(1 - \phi_{HS}) \epsilon \mathbb{Z}$

$N_{synth} \leftarrow \phi_{HS}/(1 - \phi_{HS}) - 1$

$S_{TS} = S$ where $Y < \phi_{TS}$

$S_{HS} = S$ where $Y \geq \phi_{HS}$

for $s_i \epsilon S_{HS}$ **do**

  $nn_i = \mathrm{kNN}(S, k)$

  for $j = 1, 2, ...N_{synth}$ **do**

    $s_j = nn_i(\mathbf{randi}(1, k))$ {randomly select one nearest neighbour}

    $s_{diff}, = s_i - s_j$

    $gap = \mathbf{rand}(0, 1)$ {randomly select a point between sample and nearest neighbour}

    $s_{synth,i,j} = s_i + s_{diff} \times gap$

  end for

end for

$S' = S \bigcup S_{synth}$ {merge original and synthetic data}

Algorithm 4 Bagging with resampling

Require:

Set S containing X input features and Y observations, $(x_1, y_1), ..., (x_m, y_m)$

Learner, $f()$

Number of iterations, T

Resampling function, $\mathbf{resample}()$

for $t = 1, 2, ...T$ **do**

$S'_t, D'_t \leftarrow \mathbf{resample}(S_t, D_t)$

$\mathbf{train}\ f(S'_t, D'_t)$ {train learner using resampled examples}

end for

Algorithm 5 AdaBoost.RT with resampling

Require:

Set S containing X input features and Y observations, $(x_1, y_1), ..., (x_m, y_m)$

Learner, $f()$

Number of iterations, T

Resampling function, $\mathbf{resample}()$

Relative error threshold ϕ

$D_1(i) \leftarrow \frac{1}{m}$ **for** $i = 1, ..., m$ {initialise weights array}

for $t = 1, 2, ...T$ **do**

$S', D'_t \leftarrow \mathbf{resample}(S, D_t)$

$\mathbf{train}\ f_t(S'_t, D'_t)$ {train learner using resampled examples and weights}

$\epsilon_t = \sum D_t(i), i = |\frac{(f_t(x_t) - y_t)}{y_t}| > \phi$ {calculate error rate}

$\beta_t = \epsilon_t^2$

$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \beta_t, & \text{if } |\frac{(f_t(x_t) - y_t)}{y_t}| \leq \phi \\ 1, & \text{otherwise.} \end{cases}$ {update weights for next boosting iteration}

$D_{t+1} = \mathbf{normalise}(D_t)$

end for

Algorithm 6 LSBoost with resampling

Require:

Set S containing X input features and Y observations, $(x_1, y_1), ..., (x_m, y_m)$

Learner, $f()$

Number of iterations, T

Resampling function, $\mathbf{resample}()$

Learning rate $\nu \{0 < \nu \leq 1$

$\hat{Y}_0 = \bar{Y}$

for $t = 1, 2, ...T$ **do**

$R_t = Y - \hat{Y}_{t-1}$

$S' \leftarrow \mathbf{resample}(S)$ {resample input features and residuals}

$R'_t = Y' - \hat{Y}_0 + \sum_{t=1}^{T} \rho_t f_t(X')$ {calculate the residuals corresponding the resampled data}

$\mathbf{train}\ f_t(X', R'_t)$ {train learner to latest residuals}

$\rho_t = \text{argmin} \sum [\hat{R}_t - \rho R_t]^2$

$\hat{Y}_t = \hat{Y}_{t-1} + \nu \rho_t f_t(X)$

end for

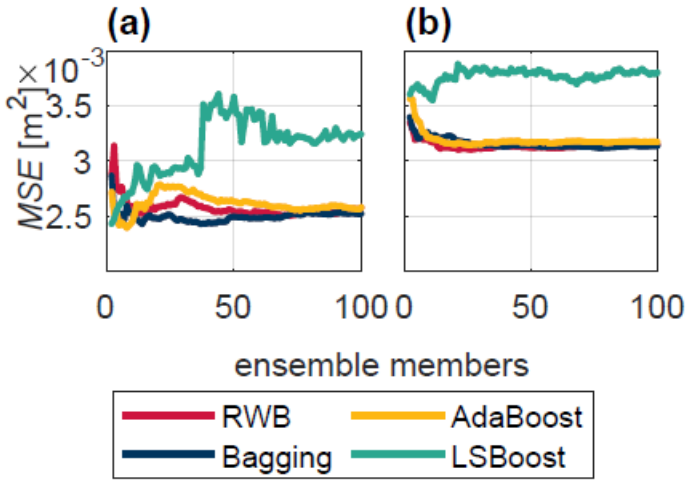

Figure 6. Test MSE across ensemble size for RWB (red), Bagging (blue), AdaBoost (yellow), and LSBoost (green) for the Don (left) and Bow River (right).

"Fig. 6 illustrates the change in test performance as the ensemble size increases from 2 to 100 for each river. This grid search is performed only for the base ensemble methods (RWB, Bagging, AdaBoost, and LSBoost) without any resampling. The Bow River plot indicates that AdaBoost and LSBoost tend to favour a small ensemble size (2-15 members), whereas the generalisation of RWB and Bagging improves with a larger size (>20 members). The performance of LSBoost rapidly deteriorates as the ensemble size grows, likely as the effects of overfitting become more pronounced. Similar results are obtained for the Don, except that RWB, Bagging, and AdaBoost all improve with larger ensemble size, while LSBoost does not offer competitive performance for any ensemble size. Again, a larger ensemble size (>20 members) produces favourable MSE."

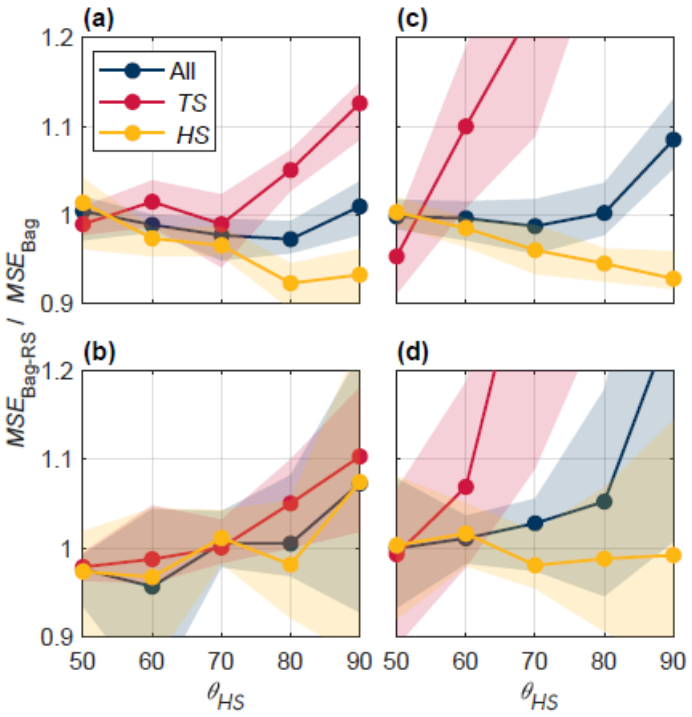

Figure 11. Calibration and test MSE ratio between Bagging and SMOTER-Bagging models for the Bow (a) and (c) and Don (b) and (d) Rivers across high stage threshold values ranging from 50% to 90%.

“As discussed in Sect. 2.1, a fixed threshold is used to distinguish between high and typical stages. Fig. 11 shows the effects of the fixed threshold increasing from the 50th to 90th percentile of the stage distribution. These plots show the relative effects of SMOTER-Bagging compared to simple Bagging. A performance ratio greater than 1 indicates that the SMOTER-Bagging model has greater error compared to the Bagging model, 1 indicates that they have the same performance, and less than 1, improved performance. The error (MSE) is presented for all stages as well as the TS and HS subsets. The calibration plots illustrate an asymmetric trade-off between HS and TS error. For a given θ_{HS} value, the error ratio of the TS subset increases more than the decline in HS error. More importantly, the improvements in HS performance obtained in calibration are considerably less pronounced in the test dataset, despite a loss in TS performance”.

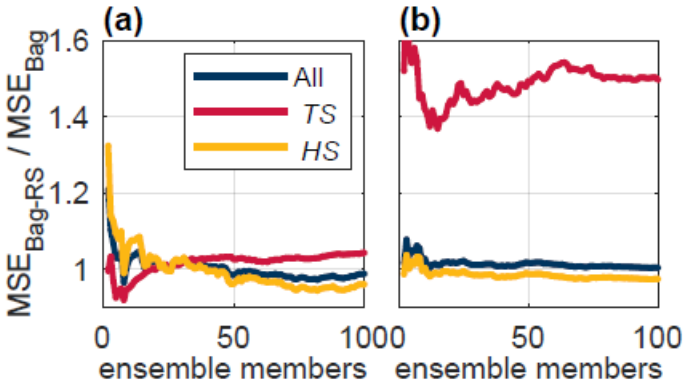

Figure 12. Test MSE ratio between Bagging and SMOTER-Bagging models for the Bow (a) and the Don (b) across ensemble size.

“Fig. 12 illustrates the effects of varying the ensemble size, thus, number of resampling repetitions, for the SMOTER-Bagging model, relative to the simple Bagging model. The plot shows the relative improvement in HS produced by the SMOTER resampling as the ensemble size increases, reaching a steady value at an ensemble size of approximately 70 for both models. This is larger than that required for the simple Bagging model to reach steady performance, indicating that SMOTER requires more resampling than simple resampling with replacement in order to reach stable performance. Consistent observations made from Fig. 11, an asymmetric trade-off between typical and high stage performance is noted.”