Open access • Journal Article • DOI:10.1016/J.ENVSOFT.2014.08.015

An evaluation framework for input variable selection algorithms for environmental data-driven models — Source link

Stefano Galelli, G. Humphrey, Holger R. Maier, Andrea Castelletti ...+2 more authors

Institutions: Singapore University of Technology and Design, University of Adelaide, Polytechnic University of Milan

Published on: 01 Dec 2014 - Environmental Modelling and Software (Elsevier)

Topics: Feature selection and Environmental data

Elsevier Editorial System(tm) for Environmental Modelling & Software
Manuscript Draft

Manuscript Number: ENVSOFT-D-14-00445R1

Title: An evaluation framework for input variable selection algorithms for environmental data-driven models

Article Type: Research Paper

Keywords: Input variable selection; Data-driven modelling; Evaluation framework; Large environmental datasets; Artificial neural networks

Corresponding Author: Dr. Stefano Galelli, Ph.D., M.Sc., B.Sc.

Corresponding Author's Institution: Singapore University of Technology and Design

First Author: Stefano Galelli, Ph.D., M.Sc., B.Sc.

Order of Authors: Stefano Galelli, Ph.D., M.Sc., B.Sc.; Greer B Humphrey, PhD; Holger R Maier, PhD; Andrea Castelletti, PhD; Graeme C Dandy, PhD; Matthew S Gibbs, PhD

Abstract: Input Variable Selection (IVS) is an essential step in data-driven modelling and is particularly relevant in environmental applications, where potential input variables are often collinear and redundant. While methods for IVS continue to emerge, each has its own advantages and limitations and no single method is best suited to all datasets and modelling purposes. Rigorous evaluation of IVS methods would allow their effectiveness to be properly identified in various circumstances. However, such evaluations are largely neglected due to the lack of guidelines to facilitate consistent and standardised assessment. This work proposes a new evaluation framework, which consists of benchmark datasets with the typical properties of environmental data, a recommended set of evaluation criteria and a website for sharing data and code. The framework is demonstrated on four IVS algorithms commonly used in environmental modelling studies. The results indicate interesting differences in the algorithms' performance that have not been identified previously.

Response to Reviewers: Editor

I have now received reviews of the above paper and these lead me to recommend that revision according to all the reviewers' comments is necessary. I may not send it back to reviewers, trusting that you will cut it down, otherwise few people will not bother reading it.

Response to Editor comment No. 1. We significantly reduced the manuscript length by mostly focusing on Section 2 and 3, as also suggested by reviewer #2. Where possible, we also tried to reduce Section 5. Overall, we obtained a reduction of about 6 pages (from the introduction to the conclusion) with respect to the previous version of the manuscript. Furthermore, we removed Appendix A, since this material can be directly accessed from the framework website. This gives an overall reduction of 21 pages.

Another issue is that I'd like it to fit better with EMS being a generic journal and so link to our key outputs. Most citations to EMS papers are to the authors themselves! Just one way to do this is to link with/refer to other key modelling concepts and issues in the journal. For example see the next paragraph.

On model evaluation: that it is credible and addressed well. In this connection, I would like you to justify, and if pertinent expand or comment upon, your choice of evaluation metrics and methods among the ones, for example, in the recent EMS Position paper of Bennett et al (2013) on performance evaluation (they propose a 5-step procedure for evaluating the performance of models). You could also add/comment on visual methods and quantitative measures used to examine model quantities and residuals, including visual inspection. There are several other evaluation issues you could address/compare as well and the paper by Robson and cited below presents an excellent example in Section 13 of that paper. One of our aims for EMS is to strengthen the credibility and relevance of the modelling reported and do this whatever the environmental problem sector. That way your paper is more suited to our journal.

Bennett ND, Croke BFW, Guariso G, Guillaume JHA, Hamilton SH, Jakeman AJ, Marsili-Libelli S, Newham LTH, Norton JP, Perrin C, Pierce SA, Robson B, Seppelt R, Voinov AA, Fath BD and Andreassian V (2013) Characterising performance of environmental models. Environmental Modelling & Software 40: 1-20.

Robson, Barbara J (in press) State of the art in modelling of phosphorus in aquatic systems: Review, criticisms and commentary. Environmental Modelling & Software, In Press, Corrected Proof, Available online 6 February 2014.

Response to Editor comment No. 2. We linked our work with other key contributions in EMS by discussing about evaluation metrics and other issues related to the use of IVS algorithms in environmental modelling problems (as also suggested by reviewer #1). This discussion is contained in the newly added Section 6.3:

"Unlike the synthetic data here considered, a key aspect of real-world environmental modelling problems is that the true underlying function is unknown, and IVS is thus used to reduce the uncertainty in the model development process by selecting a subset of relevant and non-redundant input variables. This opens some relevant theoretical and practical issues that are highlighted below:

- Most of the IVS algorithms currently available select a unique subset of input variables, although the structural uncertainty in the inputs to be used often results in the possibility of choosing different, but equally informative, subsets. An attempt to account for this issue was recently made by Sharma and Chowdhury (2011), who proposed a PMI-based heuristic approach to select five different subsets of predictors in the context of medium-term hydro-climatic forecasting. The approach ensures that the cross-dependence between these subsets is limited, while the predictions of the resulting models are eventually combined with ensemble averaging.
- In many practical situations, input variables can be characterised by errors, due, for example, to the interpolation of data in space and time or to the conversion of point measurement into areal values. Whilst methods exist for assessing the impact of input errors on parameter estimation procedures (Chowdhury and Sharma, 2007; Woldemeskel et al., 2012), IVS algorithms cannot take into account the change in the uncertainty associated with the different inputs.
- A benefit of IVS is the improvement in the performance of the model being identified. Although the manner in which such performance is characterised depends on the specific domain of interest and the model objectives (Jakeman et al., 2006), two important aspects should always be considered when dealing with quantitative testing. First, the use of observational data for comparison must rely on appropriate data-division methods, such as cross-validation or bootstrapping, that allow for testing the ability of the model to generalise. Data division can account for both temporal and spatial dimensions, so it is suitable for spatial modelling as well (see Chowdhury and Sharma (2009) for an application to hydrological modelling problems). Second, an exhaustive quantitative evaluation should rely on a set of metrics focussing on different aspects in order to test the ability of the model in reproducing all the important features of the system. The reader is referred to Bennett et al. (2013) for a comprehensive

review of techniques available for both data-division and quantitative evaluation, and to Robson (2014) for a more general assessment of environmental models."

In preparing the review we used the following rules: references to line numbers, equations and figures are all to the original manuscript; authors' reply are in blue.

Reviewer #1

It was a joy reading this paper - very nicely put together. I just had four additions to include to what has been written here.

l72 - I think one other issue needs to be added here. Most of these algorithms assume a unique input variable set exists. In my experience, a natural system can be equally well described using alternate predictor sets. This represents the structural uncertainty in specifying any one predictive model. I have attempted to highlight this issue in an invited seasonal forecasting paper (Sharma, A., and S. Chowdhury (2011), Coping with model structural uncertainty in medium-term hydro-climatic forecasting, Hydrology Research, 42(2-3), 113, doi:10.2166/nh.2011.104.) where we select 5 plausible predictor sets, but ensure the cross-dependence between them is not too high (so they can be argued to represent independent predictive models). These when combined using some model averaging rationale, lead to significant improvements in the stability of the predictive model. My argument is - for a practical problem when you wouldn't like the model to issue unstable predictions, I would pursue this option any day over selecting a single unique model. The input variable selection problem never allows for reference datasets where multiple predictive models are plausible. One of these needs to be included in any evaluation framework that is proposed (maybe just as a mixture model having two different (psuedo-independent) predictor sets.

Response to Reviewer comment No. 1. As explained in our reply to the Editor's comments, we introduced a new section to discuss the most important issues related to the use of IVS algorithms in real-world environmental modelling problems (Section 6.3). In this case, the true underlying function is unknown and different, but equally informative, subsets could indeed exist. We highlighted this aspect in Section 6.3, where we also referred to the paper mentioned above.

"Most of the IVS algorithms currently available select a unique subset of input variables, although the structural uncertainty in the inputs to be used often results in the possibility of choosing different, but equally informative, subsets. An attempt to account for this issue was recently made by Sharma and Chowdhury (2011), who proposed a PMI-based heuristic approach to select five different subsets of predictors in the context of medium-term hydro-climatic forecasting. The approach ensures that the cross-dependence between these subsets is limited, while the predictions of the resulting models are eventually combined with ensemble averaging."

l458 - I would add another dataset to this list that we needed to create to highlight the importance of the predictive algorithm when coupled to input variable selection in Sharma and Mehrotra 2014 - I suggest this as the typical datasets listed would not be able to differenciate between situations where the partial weights associated with each predictor variable are dramatically different - something that was pointed out to us in the review process of the above mentioned paper. Please see equation 22 of the paper.

Response to Reviewer comment No. 2. Since we preferred not to highlight the importance of the predictive algorithm (and the corresponding predictive performance), we decided not to include the dataset within the framework. Furthermore, we notice that datasets 6-8 and 11-18 are characterized by similar properties.

l500 - I think the selection metrics being considered could be expanded. For instance, if I am developing a predictive model to make predictions in space, the assessment can be done by leaving data points out one at a time (the usual leave-one out cross-validation) or entire blocks (I think this is called block cross-validation but not sure). If model is making prediction over time, the same thing applies along with an independent sample, the blocks here representing longer periods of time to account for persistence that may create bias with L1CV measures. An example of this is in one of my seasonal forecasting papers - Chowdhury, S., and A. Sharma (2009), Multisite seasonal forecast of arid river flows using a dynamic model combination approach, Water Resources Research, 45(10), doi:10.1029/2008wr007510. What I like most about this paper is the very extensive cross-validation that was performed towards the end, which showed the differences when using one cross-validation measure versus another.

Response to Reviewer comment No. 3. As explained in Section 3.2.1, we believe that the predictive performance should not be used when dealing with synthetic data (such as those proposed in this framework), since the accuracy depends on different factors, e.g. choice of the model or calibration method. This said, we understand that the predictive accuracy becomes important in case of real-world applications, so we included a discussion about this aspect in Section 6.3.

"A benefit of IVS is the improvement in the performance of the model being identified. Although the manner in which such performance is characterised depends on the specific domain of interest and the model objectives (Jakeman et al., 2006), two important aspects should always be considered when dealing with quantitative testing. First, the use of observational data for comparison must rely on appropriate data-division methods, such as cross-validation or bootstrapping, that allow for testing the ability of the model to generalise. Data division can account for both temporal and spatial dimensions, so it is suitable for spatial modelling as well (see Chowdhury and Sharma (2009) for an application to hydrological modelling problems). Second, an exhaustive quantitative evaluation should rely on a set of metrics focussing on different aspects in order to test the ability of the model in reproducing all the important features of the system. The reader is referred to Bennett et al. (2013) for a comprehensive review of techniques available for both data-division and quantitative evaluation, and to Robson (2014) for a more general assessment of environmental models."

Last point - I have not published this yet - but my PMI code also takes into account the change in the uncertainty associated with the predictor variable over time. Again - this was included as typical seasonal forecasting problems have markedly different standard errors depending on when the data was collected. A good example of implications of this changing error on predictions is in Chowdhury, S., and A. Sharma (2007), Mitigating Parameter Bias in Hydrological Modelling due to Uncertainty in Covariates, Journal of Hydrology, 340(doi:10.1016/j.jhydrol.2007.04.010), 197-204. But a better example of how these standard errors can be ascertained (varying over space and time, in this case for GCM simulations) is in Woldemeskel, F. M., A. Sharma, B. Sivakumar, and R. Mehrotra (2012), An error estimation method for precipitation and temperature projections for future climates, Journal of Geophysical Research-Atmospheres, 117, doi:Artn D22104Doi 10.1029/2012jd018062. Strongly feel predictor identification needs to offer a sensible basis of including such variations in data quality over time. This should be stated somewhere in this paper.

Response to Reviewer comment No. 4. This aspect is discussed in Section 6.3.

"In many practical situations, input variables can be characterised by errors, due, for example, to the interpolation of data in space and time or to the conversion of point measurement into areal values. Whilst methods exist for assessing the impact of input errors on parameter estimation procedures (Chowdhury and Sharma, 2007; Woldemeskel et al., 2012), IVS algorithms cannot take into account the change in the uncertainty associated with the different inputs."

On the whole, this is a great paper, that can be quite useful to people who identify predictor variables for use in different prediction problems. Well done folks!

Response to Reviewer comment No. 5. We thank the reviewer for the comment.

Reviewer #2

The authors propose a framework in three points to evaluate input variable selection (IVS) algorithms:
1) 26 benchmark synthetic datasets
2) a set of evaluation criteria
3) website for sharing data and results

Four IVS algorithms are compared and evaluated according to the proposed framework and discussed thoroughly. The idea is really interesting and I think frameworks of this type are more and more developed and are necessary to help research scientist be more systematic in their evaluation and comparison of new and existing methodologies. The paper is generally well written but is very long (91 pages with appendices and 56 pages before the reference section). I think it could be shorten without diminishing its coherence. I will suggest some possible ways to shorten it below.

Response to Reviewer comment No. 1. We understand that the paper is a bit lengthy, so we shortened it by removing some marginal elements (while improving some specific aspects). The revised version is 21 pages shorter (including the appendices).

I have one major disappointment: I could not find the website at the address mentioned on p. 31 (www.ivs4em.deib.polimi.it) only www.deib.polimi.it works but from there, I cannot find the framework webpage. I also tried some Google searches unsuccessfully. I think this is a limitation of the paper, if one does not have access to the benchmark datasets and cannot have a look at the web page, the whole paper remains at the stage of a good idea. Also, I think the paper could include snapshot images of the website to illustrate its functionalities for sharing results for instance. I would also expect the authors to include a functionality in the website which would allow the computation of the recommended criteria automatically. If not directly in the website, some R code could be shared to compute the criteria easily and people could contribute to new criteria.

Response to Reviewer comment No. 2. We fixed this problem. The website is now accessible, and the updated url (http://ivs4em.deib.polimi.it) is included in the revised version of the manuscript. From the website it is possible to download the 26 datasets (with their corresponding description), the source code of each IVS algorithm and an R script to compute the evaluation criteria. We have also included a functionality to upload algorithms, datasets and evaluation criteria.

We agree with the reviewer that the paper could include snapshot images of the website; however, we decided not to include them in order to limit the manuscript length.

Other comments
Section 2 describes the background on IVS methods. I found this section both a little long and not so easy to understand. I had to read the Guyon and Elisseeff (2003) paper to understand more clearly the three IVS categories. In particular, filters method are basically ranking methods (as described in Guyon and Elisseeff (2003)) and I think it’s more intuitive to present them by mentioning ranks.

Response to Reviewer comment No. 3. Following the reviewer's suggestion, we shortened Section 2 and we clarified all the unclear aspects. Furthermore, we included at the beginning of Section 2.2.1 a brief explanation of filters that explicitly refers to ranking methods.

I would suggest to start the description of each class of IVS by a typical algorithm from this class. This would help to understand the definition of the class. For the filters, the method of ranking in terms of correlation between one input and the ouput, for instance. For wrapper, the GA-ANN method used in the application of the framework could be described rapidly here. And for embedded algorithms, I would think of the LASSO algorithm which is quite popular.

Response to Reviewer comment No. 4. We agree with the reviewer that a simple description of each class of IVS would simplify the understanding of Section 2.2. Hence, we included a brief explanation of filters, wrappers and embedded algorithms at the beginning of Section 2.2.1, 2.2.2 and 2.2.3, respectively. We think that this approach is more effective than describing a typical algorithm for each class.

P. 10, lines 202-204: I have some trouble to understand why the ACF and PACF are useful for input selection. As far as I know, these techniques are used for time series analyses to choose the proper coefficients in an ARMA model. I would suggest to mention here the partial correlaction and partial mutual information which are used in the application of the framework later.

Response to Reviewer comment No. 5. ACF and PACF can be used to measure the (linear) correlation between inputs and output, and then rank the former according to the pairwise correlation. As such, they can be seen as filters. Following the reviewer's suggestion, we also mentioned the Partial Correlation Input Selection algorithm.

P. 12 lines 245-250: the description of the Gamma near-neighbour test was not clear to me. I am wondering if it is useful since this method is not used in the comparison of IVS algorithms and the purpose of the paper is not to review the state-of-the-art on IVS algorithms.

Response to Reviewer comment No. 6. The description of the Gamma near-neighbour test was shortened as suggested.

Section 3 describes the evaluation framework. Basically, as far as I am concerned, two things are missing: some real datasets and a performance criterion based on predictive accuracy. I understand the point made by the authors for the synthetic datasets: it is the only way to know the "true" inputs and their performance criteria SA are based on this knowledge. However, from a practical point of view, I am, most of the time, mainly interested to evaluate if the model selection I performed yield the best model in terms of predictive power. I think that some real datasets along with a predictive accuracy criterion would be complementary to the framework. This could be similar in spirits with the Delve datasets mentioned by the authors: some datasets are used for development and other for assessment. The real datasets could serve the later goal.

Response to Reviewer comment No. 7. We understand the reviewer's suggestion, but we believe that including some real datasets and one, or more, performance of predictive accuracy may be counterproductive. This opinion is supported by the following reasons: 1) There exists a variety of filters that do not rely on any underlying model (induction or learning algorithm), so it is not possible to evaluate the accuracy of such algorithms in terms of predictive accuracy. This would be against the rationale of the IVS framework, which is aimed at supporting the quantitative (and qualitative) evaluation of any input selection algorithm; 2) The predictive accuracy is 'biased' by several factors, such as the choice of the underlying model and calibration (and validation) algorithm. Minimizing such bias would require introducing an exhaustive comparison of different models (e.g. neural networks, regression trees, linear models, support vector machines etc.) and calibration methods, but this would dramatically affect the length of the manuscript; 3) The same reasoning applies to the inclusion of some real datasets. Indeed, the comparison of different input selection algorithms on some real datasets could only be run by comparing the predictive accuracy of some underlying models; furthermore 4) The inclusion of a few real datasets prevents an exhaustive assessment of the IVS algorithms against the statistical properties described in Section 3.1.

P. 18 lines 399-402. The sentence "Finally, the use of synthetic data enable previously unalysed datasets ... would provide very little information about algorithm performance" is not clear to me.

Response to Reviewer comment No. 8. The sentence has been removed.

P. 19 line 417 : "a universal approximator", like an artificial neural network ? or the authors have something else in mind ?

Response to Reviewer comment No. 9. Yes, a feed-forward neural network (with a single hidden layer containing a finite number of neurons) could indeed serve as a universal approximator (Cybenko, 1989). We clarified this aspect in the revised version of the manuscript.

“The amount of noise in the output is defined as the fraction of the variance that would remain unexplained if a universal approximator, such as an artificial neural network (Cybenko, 1989), were used on an infinite training set.”

Section 3.2.1. Selection accuracy: do we really need SA in addition to SAe and SAc? I find the later two sufficient since SA is computed from them. Moreover, SA requires to set a parameter which controls the tradeoff between SAe and SAc and it seems not necessary to make such a choice.

Response to Reviewer comment No. 10. We believe that the three scores (i.e. SA, SAc and SAe) are important, since they serve two different purposes: 1) The Selection Accuracy (SA) makes the comparison between different algorithms quite fast and straightforward, since it quantifies the degree to which a model has been correctly or incorrectly specified. Furthermore, the presence of the parameter γ allows the user to weight the importance of missing a relevant input against choosing an extraneous one; 2) The SAc and SAe allow for a more in-depth analysis, since they quantify the proportion of correct and extraneous inputs that have been selected.
The single SA score also allows a simple and direct trade-off between selection accuracy and runtime.

Computational efficiency: I have a tendency to think that the total runtime is enough as a measure of computational efficiency. I understand it is not directly comparable across platforms and programming languages but I am not sure if that really matters that much. What basically matters is the order of magnitude: does it take a couple of seconds or a couple of days?

Response to Reviewer comment No. 11. The total runtime provides simple, ‘practical’ information that is certainly useful to most users and practitioners. For this reason, the results in terms of runtime are reported within the text, while the analysis of computational complexity is reported in Appendix C (now Appendix B). Although only few readers may be interested in it, we believe that such analysis can have both theoretical (e.g. determining the growth rate of the runtime) and practical (e.g. planning the execution of several IVS experiments) implications, particularly as enables platform independent comparisons of the computational efficiency of different IVS algorithms. This will become increasingly important as researchers will add the performance of different algorithms to the website, as these measures will enable computational efficiency to be compared in an objective manner.

P. 29 lines 657-658: how does the framework provides a theoretical measure of computational complexity? as far as I know, this has to be computed for each IVS algorithm by considering the computation steps involved. This would be a kind of O(NP) classification for instance, am I right ?

Response to Reviewer comment No. 12. Yes, the theoretical measure of computational complexity is determined for each algorithm by evaluating the computational steps involved at each iteration (see Appendix C). This concept has been further clarified in the revised version of the manuscript.

"In particular, the analysis of computational complexity is determined for each algorithm by evaluating the computational steps involved at each iteration, and it is aimed at producing a theoretical classification that estimates the increase in run-time as a function of the input dimensionality N and P."

Experimental setup
I found it difficult to follow the explanations on the IVS algorithms and on their performance without further explanations on their mechanisms which are given in the appendix. This is why I am suggesting to use the space in the section 2 to already introduce the IVS algorithms which will be compared.

Response to Reviewer comment No. 13. We understand that Section 4 may appear unclear without reading the appendix, but, at the same time, we think that Section 2 should contain a general description of IVS approaches and not a detailed description of the IVS algorithms adopted in this study. In order to solve this problem, we included a brief description of each algorithm in Section 4, and we tightened the connection between Section 4 and the appendix.

I am wondering if it is useful to include 4 IVS algorithms since this means that all of them should be described in details for the reader to understand what is going on. For instance, p. 32 line 743, I am wondering how the Gaussian reference bandwith is set and line 749, how do you compute the "correlation between inputs and output and a multiple linear regression".

Response to Reviewer comment No. 14. The presence of four algorithms is critical to demonstrate why the framework can be useful to identify the pros and cons of different types and classes of IVS algorithms. For example, the comparison between PCIS and PMIS shows the effect due to the presence of nonlinearities, while the one between PMIS and IIS is used to discuss the effects of non-Gaussian data. Furthermore, the comparison between filters (PCIS, PMIS and IIS) and wrappers (GA-ANN) allows discussing the computational demands of different methodologies. Limiting the comparison to two algorithms would not allow for this exhaustive analysis. This said, we understand the reviewer's concern, so we clarified all these technical aspects in Section 4 (please refer to the previous reply as well).

It would probably be possible to retain 2 distinct IVS algorithms and to compare them in order to illustrate the framework. The paper would be easier to read then since the goal is not so much to inform on IVS algorithms than to present to framework.

Response to Reviewer comment No. 15. Please refer to the previous reply.

Other questions on IVS algorithms: p. 33 line 755 what are SISO models? I found the explanation later in the appendix. In general, the explanation of the IIS algorithm was fairly obscure to me.

Response to Reviewer comment No. 16. The description of the IIS algorithm has been improved as suggested.

p. 33 lines 767: a 1 hidden unit neural network do not have much non-linear capability. I understand it takes time to tune the number of hidden units of a neural network but otherwise, they do not have much predictive power.

Response to Reviewer comment No. 17. Yes, we totally agree with this remark (which is indeed commented on in Section 6.1). The adoption of such architecture, however, can easily serve our purpose: we aim at practically demonstrating the pros and cons of wrappers (and filters), rather than providing a definitive answer as to which of the algorithms performs best.

p. 33 line 771: since the number of hidden units is fixed, what is the use of k-fold cross-validation?

Response to Reviewer comment No. 18. The k-fold cross-validation is used to quantify the accuracy of the ANN. We clarified this aspect in the revised version of the manuscript.

“The accuracy of the ANN is measured in terms of out-of-sample AIC, computed using a k-fold cross-validation (with k = 5).”

p.36 lines 841-842 : " ... all four combinations of SAc and SAe were obtained for the combination of IVS algorithm and datasets..." this sentence needs to be rephrased.

Response to Reviewer comment No. 19. The sentence has been rephrased as suggested.

“Furthermore, Figure 4 shows that different values of SAc and SAe were obtained for the combination of IVS algorithms and datasets considered.”

Regarding Figs 4-5 and Figs 6-7, I think they could be re-organized; as it is, they are redundant. The authors could either choose to show the SA scores for datasets which yield contrasted results for the four IVS algorithms or to group datasets according to their properties (as it is done in the text in section 5.1.2).

Response to Reviewer comment No. 20. Following the reviewer’s comment, we removed Figure 4 and 6, since the most of the information in Figure 4 (or 6) is available from Figure 5 (or 7). The reason for maintaining Figure 5 and 7 is that they allow organizing the results by dataset (Figure 5) and by algorithm (Figure 7). The former highlights the performance of the four IVS algorithms on the same modelling conditions, while the latter provides insight into the way different dataset properties impact on the behavior of a specific algorithm.

The discussion on the results could be more condensed: p.44 lines 1021-1033: I found pretty evident that larger N helps model selection, I would suggest to shorten lines 1024-1033.

Response to Reviewer comment No. 21. Section 5.1.3 (‘Effect of N and P on algorithm performance’) has been revised and shortened. In general, the entire Section 5 has been thoroughly revised and condensed.

Computational efficiency
I found it difficult to follow the discussion on where the computations take more time for each IVS algorithm since I was not very familiar with them. I kept wondering: what is exactly Extra-trees, GRNN, PCIS, IIS... By retaining just 2 IVS algorithms and providing more detailed explanations would probably help to benefit from the kind of discussion in this section.

Response to Reviewer comment No. 22. Section 5.2 gives two different types of information about ‘Computational efficiency’. The first is based on the total runtime (Table 2) and it is built on the concept that filter algorithms (such as PMIS, PCIS and IIS) are computationally efficient, while wrappers (such as the GA-ANN algorithm) require more computing resources. This information is directly accessible by any reader, and does not require being familiar with the algorithms considered. The second information, which is based on the analysis of complexity (Table 3), requires an in-depth knowledge of the algorithms, so we believe that the improvements to Section 4 (plus the presence of a dedicated appendix) will allow the interested readers in understanding the technical aspects of such analysis.

I found the qualitive criteria section quite long. I understand the interest in these type of criteria but it could probably be shortened.

Response to Reviewer comment No. 23. The section was shortened as recommended.

Reviewer #3

The objective of this paper is to create a framework for evaluating and comparing input subset selection (IVS) algorithms for environmental modeling applications. IVS for environmental systems modeling is an extremely challenging task because of the vast number of possible explanatory variables given the space/time correlation of the processes being modeling. However, for the same reason there is also the possibility for significant colinearity of input variables. For this reason IVS is an important first step for any environmental modeling project. Unfortunately, as noted by the authors, there has been little research into what makes a "good" IVS algorithm, as most IVS algorithm research has been focused at a particular dataset or modeling task. The proposed framework would create a repository of data sets and algorithms that would permit comparison of the existing or newly proposed IVS algorithms to identify which perform well in general, thus providing guidance on which algorithm to select for new modeling projects.

This is a well written paper describing project of great interest to the environmental modeling community. The discussion of existing IVS methods is thorough given the scope and length of the paper, and the explanation of the evaluation criteria and the benchmark synthetic data sets is thorough. I recommend this paper be published as-is by Environmental Modelling and Software.

Response to Reviewer comment No. 1. We thank the reviewer for the comment.

References

Bennett, N. D., Croke, B. F., Guariso, G., Guillaume, J. H., Hamilton, S. H., Jakeman, A. J., ... & Andreassian, V. (2013). Characterising performance of environmental models. Environmental Modelling & Software, 40, 1-20.

Chowdhury, S., & Sharma, A. (2007). Mitigating parameter bias in hydrological modelling due to uncertainty in covariates. Journal of Hydrology, 340(3), 197-204.

Chowdhury, S., & Sharma, A. (2009). Multisite seasonal forecast of arid river flows using a dynamic model combination approach. Water resources research, 45(10).

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. Mathematics of control, signals and systems, 2(4), 303-314.

Jakeman, A. J., Letcher, R. A., & Norton, J. P. (2006). Ten iterative steps in development and evaluation of environmental models. Environmental Modelling & Software, 21(5), 602-614.

Robson, B. J. (2014). State of the art in modelling of phosphorus in aquatic systems: Review, criticisms and commentary. Environmental Modelling & Software.

Sharma, A., & Chowdhury, S. (2011). Coping with model structural uncertainty in medium-term hydro-climatic forecasting. Hydrology Research, 42(2-3), 113-127.

Sharma, A., & Mehrotra, R. (2014). An information theoretic alternative to model a natural system using observational information alone. Water Resources Research, 50(1), 650-660.

Woldemeskel, F. M., Sharma, A., Sivakumar, B., & Mehrotra, R. (2012). An error estimation method for precipitation and temperature projections for future climates. Journal of Geophysical Research: Atmospheres (1984–2012), 117(D22).

Cover Letter

SINGAPORE UNIVERSITY OF
TECHNOLOGY AND DESIGN

Established in collaboration with MIT

August 14, 2014

A.J. Jakeman (Editor in Chief)

Dear Tony,

We would like to thank you and the reviewers for the thorough and very helpful review. We took all the suggestions into consideration and revised the manuscript accordingly.

In particular, we improved the manuscript by focusing on three main aspects:

- We reduced the manuscript length by about 6 pages (from the introduction to the conclusion) by shortening Section 2, 3 and 5. Furthermore, we removed Appendix A. This gives an overall reduction of 21 pages with respect to the previous version of the manuscript;
- We linked our work with other key contributions in EMS, and included a discussion about the challenges related to the use of IVS algorithms in environmental modelling problems (see Section 6.3);
- We updated the link to the website url (http://ivs4em.deib.polimi.it). The website is now complete, and it also includes the scripts for calculating the evaluation criteria, as suggested by reviewer #2.

Further details and relevant information can be found in our response to the reviewers' comments.

Should you need further information, please do not hesitate to contact me.

Sincerely,

Stefano Galelli, PhD
(Corresponding author)

SINGAPORE UNIVERSITY OF
TECHNOLOGY AND DESIGN
Established in collaboration with MIT

Reply to reviewers about paper ENVSOFT-S-14-00494

An evaluation framework for input variable selection algorithms for environmental data-driven models

G.B. Humphrey, S. Galelli, H.R. Maier, A. Castelletti,
G.C. Dandy, M.S. Gibbs

Stefano Galelli, PhD
Pillar of Engineering Systems and Design
Singapore University of Technology and Design
20 Dover Drive, Singapore 138683
Tel: +65 6499 4786
Email: stefano_galelli@sutd.edu.sg

Editor

I have now received reviews of the above paper and these lead me to recommend that revision according to all the reviewers' comments is necessary. I may not send it back to reviewers, trusting that you will cut it down, otherwise few people will not bother reading it.

We significantly reduced the manuscript length by mostly focusing on Section 2 and 3, as also suggested by reviewer #2. Where possible, we also tried to reduce Section 5. Overall, we obtained a reduction of about 6 pages (from the introduction to the conclusion) with respect to the previous version of the manuscript. Furthermore, we removed Appendix A, since this material can be directly accessed from the framework website. This gives an overall reduction of 21 pages.

Another issue is that I'd like it to fit better with EMS being a generic journal and so link to our key outputs. Most citations to EMS papers are to the authors themselves! Just one way to do this is to link with/refer to other key modelling concepts and issues in the journal. For example see the next paragraph.

On model evaluation: that it is credible and addressed well. In this connection, I would like you to justify, and if pertinent expand or comment upon, your choice of evaluation metrics and methods among the ones, for example, in the recent EMS Position paper of Bennett et al (2013) on performance evaluation (they propose a 5-step procedure for evaluating the performance of models). You could also add/comment on visual methods and quantitative measures used to examine model quantities and residuals, including visual inspection. There are several other evaluation issues you could address/compare as well and the paper by Robson and cited below presents an excellent example in Section 13 of that paper. One of our aims for EMS is to strengthen the credibility and relevance of the modelling reported and do this whatever the environmental problem sector. That way your paper is more suited to our journal.

Bennett ND, Croke BFW, Guariso G, Guillaume JHA, Hamilton SH, Jakeman AJ, Marsili-Libelli S, Newham LTH, Norton JP, Perrin C, Pierce SA, Robson B, Seppelt R, Voinov AA, Fath BD and Andreassian V (2013) Characterising performance of environmental models. Environmental Modelling & Software 40: 1-20.

Robson, Barbara J (in press) State of the art in modelling of phosphorus in aquatic systems: Review, criticisms and commentary. Environmental Modelling & Software, In Press, Corrected Proof, Available online 6 February 2014.

We linked our work with other key contributions in EMS by discussing about evaluation metrics and other issues related to the use of IVS algorithms in environmental modelling problems (as also suggested by reviewer #1). This discussion is contained in the newly added Section 6.3:

"Unlike the synthetic data here considered, a key aspect of real-world environmental modelling problems is that the true underlying function is unknown, and IVS is thus used to reduce the uncertainty in the model development process by selecting a subset

of relevant and non-redundant input variables. This opens some relevant theoretical and practical issues that are highlighted below:

- *Most of the IVS algorithms currently available select a unique subset of input variables, although the structural uncertainty in the inputs to be used often results in the possibility of choosing different, but equally informative, subsets. An attempt to account for this issue was recently made by Sharma and Chowdhury (2011), who proposed a PMI-based heuristic approach to select five different subsets of predictors in the context of medium-term hydro-climatic forecasting. The approach ensures that the cross-dependence between these subsets is limited, while the predictions of the resulting models are eventually combined with ensemble averaging.*
- *In many practical situations, input variables can be characterised by errors, due, for example, to the interpolation of data in space and time or to the conversion of point measurement into areal values. Whilst methods exist for assessing the impact of input errors on parameter estimation procedures (Chowdhury and Sharma, 2007; Woldemeskel et al., 2012), IVS algorithms cannot take into account the change in the uncertainty associated with the different inputs.*
- *A benefit of IVS is the improvement in the performance of the model being identified. Although the manner in which such performance is characterised depends on the specific domain of interest and the model objectives (Jakeman et al., 2006), two important aspects should always be considered when dealing with quantitative testing. First, the use of observational data for comparison must rely on appropriate data-division methods, such as cross-validation or bootstrapping, that allow for testing the ability of the model to generalise. Data division can account for both temporal and spatial dimensions, so it is suitable for spatial modelling as well (see Chowdhury and Sharma (2009) for an application to hydrological modelling problems). Second, an exhaustive quantitative evaluation should rely on a set of metrics focussing on different aspects in order to test the ability of the model in reproducing all the important features of the system. The reader is referred to Bennett et al. (2013) for a comprehensive review of techniques available for both data-division and quantitative evaluation, and to Robson (2014) for a more general assessment of environmental models."*

In preparing the review we used the following rules: references to line numbers, equations and figures are all to the original manuscript; authors' reply are in blue.

Reviewer #1

It was a joy reading this paper - very nicely put together. I just had four additions to include to what has been written here.

l72 - I think one other issue needs to be added here. Most of these algorithms assume a unique input variable set exists. In my experience, a natural system can be equally well described using alternate predictor sets. This represents the structural uncertainty in specifying any one predictive model. I have attempted to highlight this issue in an invited seasonal forecasting paper (Sharma, A., and S. Chowdhury (2011), Coping with model structural uncertainty in medium-term hydro-climatic forecasting, Hydrology Research, 42(2-3), 113, doi:10.2166/nh.2011.104.) where we select 5 plausible predictor sets, but ensure the cross-dependence between them is not too high (so they can be argued to represent independent predictive models). These when combined using some model averaging rationale, lead to significant improvements in the stability of the predictive model. My argument is - for a practical problem when you wouldn't like the model to issue unstable predictions, I would pursue this option any day over selecting a single unique model. The input variable selection problem never allows for reference datasets where multiple predictive models are plausible. One of these needs to be included in any evaluation framework that is proposed (maybe just as a mixture model having two different (psuedo-independent) predictor sets.

As explained in our reply to the Editor's comments, we introduced a new section to discuss the most important issues related to the use of IVS algorithms in real-world environmental modelling problems (Section 6.3). In this case, the true underlying function is unknown and different, but equally informative, subsets could indeed exist. We highlighted this aspect in Section 6.3, where we also referred to the paper mentioned above.

"Most of the IVS algorithms currently available select a unique subset of input variables, although the structural uncertainty in the inputs to be used often results in the possibility of choosing different, but equally informative, subsets. An attempt to account for this issue was recently made by Sharma and Chowdhury (2011), who proposed a PMI-based heuristic approach to select five different subsets of predictors in the context of medium-term hydro-climatic forecasting. The approach ensures that the cross-dependence between these subsets is limited, while the predictions of the resulting models are eventually combined with ensemble averaging."

l458 - I would add another dataset to this list that we needed to create to highlight the importance of the predictive algorithm when coupled to input variable selection in Sharma and Mehrotra 2014 - I suggest this as the typical datasets listed would not be able to differenciate between situations where the partial weights associated with each predictor variable are dramatically different - something that was pointed out to us in the review process of the above mentioned paper. Please see equation 22 of the paper.

Since we preferred not to highlight the importance of the predictive algorithm (and the corresponding predictive performance), we decided not to include the dataset within the framework. Furthermore, we notice that datasets 6-8 and 11-18 are characterized by similar properties.

l500 - I think the selection metrics being considered could be expanded. For instance, if I am developing a predictive model to make predictions in space, the assessment can be done by leaving data points out one at a time (the usual leave-one out cross-validation) or entire blocks (I think this is called block cross-validation but not sure). If model is making prediction over time, the same thing applies along with an independent sample, the blocks here representing longer periods of time to account for persistence that may create bias with L1CV measures. An example of this is in one of my seasonal forecasting papers - Chowdhury, S., and A. Sharma (2009), Multisite seasonal forecast of arid river flows using a dynamic model combination approach, Water Resources Research, 45(10), doi:10.1029/2008wr007510. What I like most about this paper is the very extensive cross-validation that was performed towards the end, which showed the differences when using one cross-validation measure versus another.

As explained in Section 3.2.1, we believe that the predictive performance should not be used when dealing with synthetic data (such as those proposed in this framework), since the accuracy depends on different factors, e.g. choice of the model or calibration method. This said, we understand that the predictive accuracy becomes important in case of real-world applications, so we included a discussion about this aspect in Section 6.3.

"A benefit of IVS is the improvement in the performance of the model being identified. Although the manner in which such performance is characterised depends on the specific domain of interest and the model objectives (Jakeman et al., 2006), two important aspects should always be considered when dealing with quantitative testing. First, the use of observational data for comparison must rely on appropriate data-division methods, such as cross-validation or bootstrapping, that allow for testing the ability of the model to generalise. Data division can account for both temporal and spatial dimensions, so it is suitable for spatial modelling as well (see Chowdhury and Sharma (2009) for an application to hydrological modelling problems). Second, an exhaustive quantitative evaluation should rely on a set of metrics focussing on different aspects in order to test the ability of the model in reproducing all the important features of the system. The reader is referred to Bennett et al. (2013) for a comprehensive review of techniques available for both data-division and quantitative evaluation, and to Robson (2014) for a more general assessment of environmental models."

Last point - I have not published this yet - but my PMI code also takes into account the change in the uncertainty associated with the predictor variable over time. Again - this was included as typical seasonal forecasting problems have markedly different standard errors depending on when the data was collected. A good example of implications of this changing error on predictions is in Chowdhury, S., and A. Sharma (2007), Mitigating Parameter Bias in Hydrological Modelling due to Uncertainty in Covariates, Journal of Hydrology, 340(doi:10.1016/j.jhydrol.2007.04.010), 197-204. But a better example of how these standard errors can be ascertained (varying over space and time, in this case for GCM simulations) is in Woldemeskel, F. M., A. Sharma, B. Sivakumar, and R. Mehrotra (2012), An error estimation method for precipitation and temperature projections for future climates, Journal of Geophysical Research-Atmospheres, 117, doi:Artn D22104Doi 10.1029/2012jd018062. Strongly feel predictor identification needs to offer a sensible basis of including such variations in data quality over time. This should be stated somewhere in this paper.

This aspect is discussed in Section 6.3.

"In many practical situations, input variables can be characterised by errors, due, for example, to the interpolation of data in space and time or to the conversion of point measurement into areal values. Whilst methods exist for assessing the impact of input errors on parameter estimation procedures (Chowdhury and Sharma, 2007;

Woldemeskel et al., 2012), IVS algorithms cannot take into account the change in the uncertainty associated with the different inputs."

On the whole, this is a great paper, that can be quite useful to people who identify predictor variables for use in different prediction problems. Well done folks!

We thank the reviewer for the comment.

Reviewer #2

The authors propose a framework in three points to evaluate input variable selection (IVS) algorithms:
1) 26 benchmark synthetic datasets
2) a set of evaluation criteria
3) website for sharing data and results

Four IVS algorithms are compared and evaluated according to the proposed framework and discussed thoroughly. The idea is really interesting and I think frameworks of this type are more and more developed and are necessary to help research scientist be more systematic in their evaluation and comparison of new and existing methodologies. The paper is generally well written but is very long (91 pages with appendices and 56 pages before the reference section). I think it could be shorten without diminishing its coherence. I will suggest some possible ways to shorten it below.

We understand that the paper is a bit lengthy, so we shortened it by removing some marginal elements (while improving some specific aspects). The revised version is 21 pages shorter (including the appendices).

I have one major disappointment: I could not find the website at the address mentioned on p. 31 (www.ivs4em.deib.polimi.it) only www.deib.polimi.it works but from there, I cannot find the framework webpage. I also tried some Google searches unsuccessfully. I think this is a limitation of the paper, if one does not have access to the benchmark datasets and cannot have a look at the web page, the whole paper remains at the stage of a good idea. Also, I think the paper could include snapshot images of the website to illustrate its functionalities for sharing results for instance. I would also expect the authors to include a functionality in the website which would allow the computation of the recommended criteria automatically. If not directly in the website, some R code could be shared to compute the criteria easily and people could contribute to new criteria.

We fixed this problem. The website is now accessible, and the updated url (http://ivs4em.deib.polimi.it) is included in the revised version of the manuscript. From the website it is possible to download the 26 datasets (with their corresponding description), the source code of each IVS algorithm and an R script to compute the evaluation criteria. We have also included a functionality to upload algorithms, datasets and evaluation criteria.

We agree with the reviewer that the paper could include snapshot images of the website; however, we decided not to include them in order to limit the manuscript length.

Other comments
Section 2 describes the background on IVS methods. I found this section both a little long and not so easy to understand. I had to read the Guyon and Elisseeff (2003) paper to understand more clearly the three IVS categories. In particular, filters method are basically ranking methods (as described in Guyon and Elisseeff (2003)) and I think it's more intuitive to present them by mentioning ranks.

Following the reviewer's suggestion, we shortened Section 2 and we clarified all the unclear aspects. Furthermore, we included at the beginning of Section 2.2.1 a brief explanation of filters that explicitly refers to ranking methods.

I would suggest to start the description of each class of IVS by a typical algorithm from this class. This would help to understand the definition of the class. For the filters, the method of ranking in terms of correlation between one input and the ouput, for instance. For wrapper, the GA-ANN method used in the application of the framework could be described rapidly here. And for embedded algorithms, I would think of the LASSO algorithm which is quite popular.

We agree with the reviewer that a simple description of each class of IVS would simplify the understanding of Section 2.2. Hence, we included a brief explanation of filters, wrappers and embedded algorithms at the beginning of Section 2.2.1, 2.2.2 and 2.2.3, respectively. We think that this approach is more effective than describing a typical algorithm for each class.

P. 10, lines 202-204: I have some trouble to understand why the ACF and PACF are useful for input selection. As far as I know, these techniques are used for time series analyses to choose the proper coefficients in an ARMA model. I would suggest to mention here the partial correlaction and partial mutual information which are used in the application of the framework later.

ACF and PACF can be used to measure the (linear) correlation between inputs and output, and then rank the former according to the pairwise correlation. As such, they can be seen as filters. Following the reviewer's suggestion, we also mentioned the Partial Correlation Input Selection algorithm.

P. 12 lines 245-250: the description of the Gamma near-neighbour test was not clear to me. I am wondering if it is useful since this method is not used in the comparison of IVS algorithms and the purpose of the paper is not to review the state-of-the-art on IVS algorithms.

The description of the Gamma near-neighbour test was shortened as suggested.

Section 3 describes the evaluation framework. Basically, as far as I am concerned, two things are missing: some real datasets and a performance criterion based on predictive accuracy. I understand the point made by the authors for the synthetic datasets: it is the only way to know the "true" inputs and their performance criteria SA are based on this knowledge. However, from a practical point of view, I am, most of the time, mainly interested to evaluate if the model selection I performed yield the best model in terms of predictive power. I think that some real datasets along with a predictive accuracy criterion would be complementary to the framework. This could be similar in spirits with the Delve datasets mentioned by the authors: some datasets are used for development and other for assessment. The real datasets could serve the later goal.

We understand the reviewer's suggestion, but we believe that including some real datasets and one, or more, performance of predictive accuracy may be counterproductive. This opinion is supported by the following reasons: 1) There exists a variety of filters that do not rely on any underlying model (induction or learning algorithm), so it is not possible to evaluate the accuracy of such algorithms in terms of predictive accuracy. This would be against the rationale of the IVS framework, which is aimed at supporting the quantitative (and qualitative) evaluation of any input selection algorithm; 2) The predictive accuracy is 'biased' by several factors, such as the choice of the underlying model and calibration (and validation) algorithm. Minimizing such bias would require introducing an exhaustive comparison of different models (e.g. neural networks, regression trees, linear models, support vector machines etc.) and calibration methods, but this would dramatically affect the length of the manuscript; 3) The same reasoning applies to the inclusion of some real datasets. Indeed, the comparison of different input selection algorithms on some real datasets could only be run by comparing the predictive accuracy of some underlying models; furthermore 4) The inclusion of a few real datasets prevents an exhaustive assessment of the IVS algorithms against the statistical properties described in Section 3.1.

P. 18 lines 399-402. The sentence "Finally, the use of synthetic data enable previously unalysed datasets ... would provide very little information about algorithm performance" is not clear to me.

The sentence has been removed.

P. 19 line 417 : "a universal approximator", like an artificial neural network ? or the authors have something else in mind ?

Yes, a feed-forward neural network (with a single hidden layer containing a finite number of neurons) could indeed serve as a universal approximator (Cybenko, 1989). We clarified this aspect in the revised version of the manuscript.

"The amount of noise in the output is defined as the fraction of the variance that would remain unexplained if a universal approximator, such as an artificial neural network (Cybenko, 1989), were used on an infinite training set."

Section 3.2.1. Selection accuracy: do we really need SA in addition to SAe and SAc? I find the later two sufficient since SA is computed from them. Moreover, SA requires to set a parameter which controls the tradeoff between SAe and SAc and it seems not necessary to make such a choice.

We believe that the three scores (i.e. SA, SAc and SAe) are important, since they serve two different purposes: 1) The Selection Accuracy (SA) makes the comparison between different algorithms quite fast and straightforward, since it quantifies the degree to which a model has been correctly or incorrectly specified. Furthermore, the presence of the parameter γ allows the user to weight the importance of missing a relevant input against choosing an extraneous one; 2) The SAc and SAe allow for a more in-depth analysis, since they quantify the proportion of correct and extraneous inputs that have been selected.
The single SA score also allows a simple and direct trade-off between selection accuracy and runtime.

Computational efficiency: I have a tendency to think that the total runtime is enough as a measure of computational efficiency. I understand it is not directly comparable across platforms and programming languages but I am not sure if that really matters that much. What basically matters is the order of magnitude: does it take a couple of seconds or a couple of days?

The total runtime provides simple, ‘practical’ information that is certainly useful to most users and practitioners. For this reason, the results in terms of runtime are reported within the text, while the analysis of computational complexity is reported in Appendix C (now Appendix B). Although only few readers may be interested in it, we believe that such analysis can have both theoretical (e.g. determining the growth rate of the runtime) and practical (e.g. planning the execution of several IVS experiments) implications, particularly as enables platform independent comparisons of the computational efficiency of different IVS algorithms. This will become increasingly important as researchers will add the performance of different algorithms to the website, as these measures will enable computational efficiency to be compared in an objective manner.

P. 29 lines 657-658: how does the framework provides a theoretical measure of computational complexity? as far as I know, this has to be computed for each IVS algorithm by considering the computation steps involved. This would be a kind of O(NP) classification for instance, am I right ?

Yes, the theoretical measure of computational complexity is determined for each algorithm by evaluating the computational steps involved at each iteration (see Appendix C). This concept has been further clarified in the revised version of the manuscript.

“In particular, the analysis of computational complexity is determined for each algorithm by evaluating the computational steps involved at each iteration, and it is aimed at producing a theoretical classification that estimates the increase in run-time as a function of the input dimensionality N and P.”

Experimental setup
I found it difficult to follow the explanations on the IVS algorithms and on their performance without further explanations on their mechanisms which are given in the appendix. This is why I am suggesting to use the space in the section 2 to already introduce the IVS algorithms which will be compared.

We understand that Section 4 may appear unclear without reading the appendix, but, at the same time, we think that Section 2 should contain a general description of IVS approaches and not a detailed description of the IVS algorithms adopted in this study. In order to solve this problem, we included a brief description of each algorithm in Section 4, and we tightened the connection between Section 4 and the appendix.

I am wondering if it is useful to include 4 IVS algorithms since this means that all of them should be described in details for the reader to understand what is going on. For instance, p. 32 line 743, I am wondering how the Gaussian reference bandwith is set and line 749, how do you compute the "correlation between inputs and output and a multiple linear regression".

The presence of four algorithms is critical to demonstrate why the framework can be useful to identify the pros and cons of different types and classes of IVS algorithms. For example, the comparison between PCIS and PMIS shows the effect due to the presence of nonlinearities, while the one between PMIS and IIS is used to discuss the effects of non-Gaussian data. Furthermore, the comparison between filters (PCIS, PMIS and IIS) and wrappers (GA-ANN) allows discussing the computational demands of different methodologies. Limiting the comparison to two algorithms would not allow for this exhaustive analysis. This said, we understand the reviewer's concern, so we clarified all these technical aspects in Section 4 (please refer to the previous reply as well).

It would probably be possible to retain 2 distinct IVS algorithms and to compare them in order to illustrate the framework. The paper would be easier to read then since the goal is not so much to inform on IVS algorithms than to present to framework.

Please refer to the previous reply.

Other questions on IVS algorithms: p. 33 line 755 what are SISO models? I found the explanation later in the appendix. In general, the explanation of the IIS algorithm was fairly obscure to me.

The description of the IIS algorithm has been improved as suggested.

p. 33 lines 767: a 1 hidden unit neural network do not have much non-linear capability. I understand it takes time to tune the number of hidden units of a neural network but otherwise, they do not have much predictive power.

Yes, we totally agree with this remark (which is indeed commented on in Section 6.1). The adoption of such architecture, however, can easily serve our purpose: we aim at practically demonstrating the pros and cons of wrappers (and filters), rather than providing a definitive answer as to which of the algorithms performs best.

p. 33 line 771: since the number of hidden units is fixed, what is the use of k-fold cross-validation?

The *k*-fold cross-validation is used to quantify the accuracy of the ANN. We clarified this aspect in the revised version of the manuscript.

"The accuracy of the ANN is measured in terms of out-of-sample AIC, computed using a k-fold cross- validation (with k = 5)."

p.36 lines 841-842 : " ... all four combinations of SAc and SAe were obtained for the combination of IVS algorithm and datasets..." this sentence needs to be rephrased.

The sentence has been rephrased as suggested.

"Furthermore, Figure 4 shows that different values of SA_c and SA_e were obtained for the combination of IVS algorithms and datasets considered."

Regarding Figs 4-5 and Figs 6-7, I think they could be re-organized; as it is, they are redundant. The authors could either choose to show the SA scores for datasets which yield contrasted results for the four IVS algorithms or to group datasets according to their properties (as it is done in the text in section 5.1.2).

Following the reviewer's comment, we removed Figure 4 and 6, since the most of the information in Figure 4 (or 6) is available from Figure 5 (or 7). The reason for maintaining Figure 5 and 7 is that they allow organizing the results by dataset (Figure 5) and by algorithm (Figure 7). The former highlights the performance of the four IVS algorithms on the same modelling conditions, while the latter provides insight into the way different dataset properties impact on the behavior of a specific algorithm.

The discussion on the results could be more condensed: p.44 lines 1021-1033: I found pretty evident that larger N helps model selection, I would suggest to shorten lines 1024-1033.

Section 5.1.3 ('Effect of *N* and *P* on algorithm performance') has been revised and shortened. In general, the entire Section 5 has been thoroughly revised and condensed.

Computational efficiency
I found it difficult to follow the discussion on where the computations take more time for each IVS algorithm since I was not very familiar with them. I kept wondering: what is exactly Extra-trees, GRNN, PCIS, IIS... By retaining just 2 IVS algorithms and providing more detailed explanations would probably help to benefit from the kind of discussion in this section.

Section 5.2 gives two different types of information about 'Computational efficiency'. The first is based on the total runtime (Table 2) and it is built on the concept that filter algorithms (such as PMIS, PCIS and IIS) are computationally efficient, while wrappers (such as the GA-ANN algorithm) require more computing resources. This information is directly accessible by any reader, and does not require being familiar with the algorithms considered. The second information, which is based on the analysis of complexity (Table 3), requires an in-depth knowledge of the algorithms, so we believe that the improvements to Section 4 (plus the presence of a dedicated appendix) will allow the interested readers in understanding the technical aspects of such analysis.

I found the qualitive criteria section quite long. I understand the interest in these type of criteria but it could probably be shortened.

The section was shortened as recommended.

Reviewer #3

The objective of this paper is to create a framework for evaluating and comparing input subset selection (IVS) algorithms for environmental modeling applications. IVS for environmental systems modeling is an extremely challenging task because of the vast number of possible explanatory variables given the space/time correlation of the processes being modeling. However, for the same reason there is also the possibility for significant colinearity of input variables. For this reason IVS is an important first step for any environmental modeling project. Unfortunately, as noted by the authors, there has been little research into what makes a "good" IVS algorithm, as most IVS algorithm research has been focused at a particular dataset or modeling task. The proposed framework would create a repository of data sets and algorithms that would permit comparison of the existing or newly proposed IVS algorithms to identify which perform well in general, thus providing guidance on which algorithm to select for new modeling projects.

This is a well written paper describing project of great interest to the environmental modeling community. The discussion of existing IVS methods is thorough given the scope and length of the paper, and the explanation of the evaluation criteria and the benchmark synthetic data sets is thorough. I recommend this paper be published as-is by Environmental Modelling and Software.

We thank the reviewer for the comment.

References

Bennett, N. D., Croke, B. F., Guariso, G., Guillaume, J. H., Hamilton, S. H., Jakeman, A. J., ... & Andreassian, V. (2013). Characterising performance of environmental models. *Environmental Modelling & Software*, 40, 1-20.

Chowdhury, S., & Sharma, A. (2007). Mitigating parameter bias in hydrological modelling due to uncertainty in covariates. *Journal of Hydrology*, 340(3), 197-204.

Chowdhury, S., & Sharma, A. (2009). Multisite seasonal forecast of arid river flows using a dynamic model combination approach. *Water resources research*, 45(10).

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4), 303-314.

Jakeman, A. J., Letcher, R. A., & Norton, J. P. (2006). Ten iterative steps in development and evaluation of environmental models. *Environmental Modelling & Software*, 21(5), 602-614.

Robson, B. J. (2014). State of the art in modelling of phosphorus in aquatic systems: Review, criticisms and commentary. *Environmental Modelling & Software*.

Sharma, A., & Chowdhury, S. (2011). Coping with model structural uncertainty in medium-term hydro-climatic forecasting. *Hydrology Research*, 42(2-3), 113-127.

Sharma, A., & Mehrotra, R. (2014). An information theoretic alternative to model a natural system using observational information alone. *Water Resources Research*, 50(1), 650-660.

Woldemeskel, F. M., Sharma, A., Sivakumar, B., & Mehrotra, R. (2012). An error estimation method for precipitation and temperature projections for future climates. *Journal of Geophysical Research: Atmospheres* (1984–2012), 117(D22).

***Suggested Reviewer List (include up to 5 names and their contact details)**

Prof. Amin ELSHORBAGY
Department of Civil & Geological Engineering
University of Saskatchewan
E-mail: amin.elshorbagy@usask.ca
Tel: +1 (306) 966 5414
Fax: +1 (306) 966 5205
Web: http://www.hydropyramids.com

Prof. Ashu JAIN
Department of Civil Engineering
Indian Institute of Technology Kanpur
E-mail: ashujain@iitk.ac.in
Tel: +91 512 259 7411
Fax: +91 512 259 7395
Web: http://home.iitk.ac.in/~ashujain/

Prof. K.P. SUDHEER
Department of Civil Engineering
Indian Institute of Technology Madras
E-mail: sudheer@iitm.ac.in
Tel: +91 44 2257 4288
Fax: -
Web: http://www.civil.iitm.ac.in/?q=sudheer_edu

Prof. Dimitri SOLOMATINE
Hydroinformatics Chair group
UNESCO-IHE Institute for Water Education
E-mail: d.solomatine@unesco-ihe.org
Tel: +31 1521 51815
Fax: -
Web: http://www.unesco-ihe.org/dimitri-solomatine

Dr. Chris DAWSON
Loughborough University
Department of Computer Science
E-mail: c.w.dawson1@lboro.ac.uk
Tel: +44 (0) 1509 222684
Fax: -
Web: http://www.lboro.ac.uk/departments/compsci/staff/dr-christian-w-dawson.html

Prof. Nick MOUNT
School of Geography
The University of Nottingham
E-mail: nick.mount@nottingham.ac.uk
Tel: +44 (0) 115 95 15438
Fax: -
Web: http://www.nottingham.ac.uk/geography/people/nick.mount

***Highlights (for review)**

A framework for the evaluation of Input Variable Selection algorithms is proposed.

The framework consists of assessment criteria and twenty-six datasets.

The framework is supported by a dedicated website (http://ivs4em.deib.polimi.it).

Four popular IVS algorithms are considered for evaluation purposes.

An evaluation framework for input variable selection algorithms for environmental data-driven models

Stefano Galelli[a,*], Greer B. Humphrey[b], Holger R. Maier[b], Andrea Castelletti[c], Graeme C. Dandy[b], Matthew S. Gibbs[b,d]

[a] *Pillar of Engineering Systems and Design, Singapore University of Technology and Design, 20 Dover Drive, 138682, Singapore*
[b] *School of Civil, Environmental, and Mining Engineering, University of Adelaide, SA 5005, Australia*
[c] *Department of Electronics, Information, and Bioengineering, Politecnico di Milano, Piazza L. da Vinci, 32, 20133, Milan, Italy*
[d] *Department of Environment, Water and Natural Resources, GPO Box 2384, Adelaide, SA 5001, Australia*

Abstract

Input Variable Selection (IVS) is an essential step in the development of data–driven models and is particularly relevant in environmental modelling. While new methods for identifying important model inputs continue to emerge, each has its own advantages and limitations and no single method is best suited to all datasets and modelling purposes. Rigorous evaluation of new and existing input variable selection methods would allow the effectiveness of these algorithms to be properly identified in various circumstances. However, such evaluations are largely neglected due to the lack of guidelines or precedent to facilitate consistent and standardised assessment. In this paper, a new framework is proposed for the evaluation and inter–comparison of IVS methods which takes into account: (1) a wide range of dataset properties that are relevant to

[*] Corresponding author. Tel.: +65 6499 4786.
E-mail address: stefano galelli@sutd.edu.sg

Preprint submitted to Environmental Modelling & Software *August 14, 2014*

real world environmental data, (2) assessment criteria selected to highlight algorithm suitability in different situations of interest, and (3) a website for sharing data, algorithms and results (http://ivs4em.deib.polimi.it/). The framework is demonstrated on four IVS algorithms commonly used in environmental modelling studies and twenty-six datasets exhibiting different typical properties of environmental data. The main aim at this stage is to demonstrate the application of the proposed evaluation framework, rather than provide a definitive answer as to which of these algorithms has the best overall performance. Nevertheless, the results indicate interesting differences in the algorithms' performance that have not been identified previously.

Keywords: Input variable selection, Data-driven modelling, Evaluation framework, Large environmental datasets, Artificial neural networks

Software and data availability

Software

Name of software: PMIS_PCIS, IIS, GA_ANN.

Developers (PMIS_PCIS, GA_ANN): Greer B. Humphrey, Holger R. Maier, Graeme C. Dandy, Matthew S. Gibbs.

Developers (IIS): Stefano Galelli, Andrea Castelletti.

Year first available: 2014.

Hardware required: PC or MAC.

Software required: R (PMIS_PCIS and GA_ANN), MatLab (IIS).

Program language: R (PMIS_PCIS and GA_ANN), MatLab (IIS).

Program size: 41 KB (PMIS_PCIS), 135 KB (IIS), 172 KB (GA_ANN).

Data

Name of dataset: IVS Framework datasets.

Developers: Greer B. Humphrey.

Form of repository: zipped files.

Size of archive: 239.3 MB.

Access form: public Dropbox folder.

Contact address: Pillar of Engineering Systems and Design, Singapore University of Technology and Design, 20 Dover Drive, Singapore 138682.

Telephone: + 65 6499 4786.

E-mail: stefano_galelli@sutd.edu.sg.

Url: http://ivs4em.deib.polimi.it.

Availability: software and data are available on the IVS framework website.

Cost: free of charge.

1. Introduction

In data-driven modelling, such as the application of Artificial Neural Networks (ANNs), determining which inputs are most useful for predicting a variable of interest can be one of the most critical decisions in the model development process. The input variables (or predictors) contain the information necessary for defining, albeit, in a simplified manner, the underlying process that generated the data. However, the set of *candidate* inputs usually includes variables which might be either *irrelevant* to the problem or *redundant*. Irrelevant input variables are uninformative about the underlying process and only serve to add noise and complexity into the model, while the inclusion of redundant, but relevant, inputs increases the dimensionality of the model identification problem without providing any additional predictive benefit. The omission of relevant input variables, on the other hand, leads to an inaccurate model, where part of the output behaviour remains unexplained by the selected input variables. Thus, the appropriate selection of both relevant and non-redundant inputs can mean the difference between a reliable and parsimonious model, which generalises well to the underlying process, and a model that produces nonsensical outputs (garbage in, garbage out), is slower to run, and more difficult to interpret. The challenge of Input Variable Selection (IVS) is, therefore, to select the fewest input variables that best characterise the underlying input-output relationship while minimising variable redundancy (Guyon and Elisseeff, 2003).

While the task of IVS is not unique to environmental modelling, it can be a particularly difficult one when it comes to environmental systems, since many of the underlying processes are often partially understood. Furthermore, as environmental systems vary in space and time, potentially important inputs may include observations of causal variables at different locations and time lags, as well as lagged observations of the dependent variable of interest (Maier and Dandy, 2000). As a result, the number of potentially important inputs can be very large; a problem which has been exacerbated in recent years by the emergence of new types of data, including remotely sensed, GIS and reanalysis data. To further complicate matters, the correlated nature of such input variables induces redundancy and collinearity in the input pool (Galelli and Castelletti, 2013b), while the non-linearity and inherent complexity associated with environmental systems make it ineffective to apply well established analytical variable selection methods, such as correlation analysis (May et al., 2011). As such, the development and adaptation of IVS methods for environmental modelling applications is an important and active field of research, which has further been stimulated by reviews of environmental modelling procedures discussing the need for improved and more rigorous IVS (see, for example, Araújo and Guisan (2006); Elith and Leathwick (2009); Maier et al. (2010); Abrahart et al. (2012); Wu et al. (2014)).

However, despite recent efforts to improve IVS in environmental modelling, studies in this field tend to draw overly general conclusions about the performance of the IVS approaches used. They are usually conducted with a single focus (e.g. to select the inputs for a particular case study of interest) and the evaluation of IVS methods is summarised accordingly (e.g. based on the predictive performance of the resulting model). Such evaluations make it difficult to determine how the performance of one IVS method, either new or existing, compares with that of another, and, ultimately, are of limited value to users wishing to select an IVS method that is most appropriate for a particular problem. As noted by Elshorbagy et al. (2010) in relation to the development of data-driven modelling techniques in hydrology, "one of the fundamental means to assess a modelling technique is to evaluate it against other modelling techniques", yet "comparative studies are usually impaired due to the less-than-comprehensive approach adopted". The same can be said about the assessment and comparison of IVS methods, where current studies tend to:

- select a limited number of data sets which do not adequately encompass the range of properties typical of environmental data (e.g. nonlinear, non-Gaussian, high redundancy);

- select case studies for which the “true” inputs are unknown and thus do not enable selection accuracy to be properly assessed;

- consider limited assessment criteria, often based on the predictive performance of the constructed model, which is complicated by the chosen functional form of the model and calibration performance;

- lack rigorous implementation (e.g. no repeated experiments), thus preventing the statistical significance of any observed results to be evaluated; and

- only consider a single algorithm without comparison with other algorithms.

In order to address these shortcomings, a generic framework for the standardised and rigorous comparative analysis of IVS algorithms is introduced in this paper. The framework is comprised of three main components: (1) a set of benchmark data; (2) a recommended set of evaluation criteria; and (3) a website for sharing data and results. The datasets are synthetically generated to have, to different degrees, the typical properties of real environmental data, while the evaluation criteria are designed to quantitatively and comprehensively assess selection accuracy and computational complexity. To demonstrate the application of the framework, four IVS algorithms commonly adopted in environmental modelling exercises and representative of different IVS approaches are comparatively analysed. It is hoped that this framework will facilitate collaboration by researchers developing new IVS algorithms and modellers wishing to select an appropriate IVS method.

The remainder of this paper is structured as follows: Section 2 provides a background on IVS methods, with particular focus on those used to date in environmental modelling studies. In Section 3, the proposed IVS evaluation framework is presented, while Section 4 describes the application of the framework to four IVS algorithms. Results of these evaluations and comparisons are presented in Section 5, while discussion and conclusions are given in Sections 6 and 7.

2. Background on IVS Methods in Environmental Modelling

In recent years, the use of automatic and systematic IVS algorithms has been shown to improve prediction accuracy and produce more parsimonious models in numerous applications when compared with empirical IVS methods or the inclusion of all available input data (e.g. Bowden et al. (2005b); D'heygere et al. (2006); Yang and Ong (2011); Wan Jaafar et al. (2011); Tirelli and Pessani (2011)). Comprehensive discussions on the taxonomy of such IVS methods can be found in Blum and Langley (1997), Liu and Motoda (1998), Guyon and Elisseeff (2003) and May et al. (2011). A brief overview is provided here for the purpose of highlighting the relative differences and merits of the various IVS approaches. Figure 1 is adapted from Dash and Liu (1997), who outline the basic steps of any automatic IVS algorithm. As can be seen, such methods involve three main steps: (1) generating a subset of inputs from the candidate input pool; (2) evaluating the subset of inputs in terms of their ability to predict the output; and (3) assessing whether the selected set of inputs is optimal using a pre-specified stopping criterion.

2.1. Input subset generation

The generation of input subsets is determined by the method used to search the space of all possible input subsets. An exhaustive search of the space is generally infeasible, as there exist $2^P - 1$ possible subsets of input variables, where P is the dimension of the candidate input pool. Instead, search strategies applied to IVS algorithms seek to balance the trade-off between finding the optimal subset of input variables and computational efficiency. These strategies may be classified as *global*, where many combinations of input subsets are considered; or *local*, where the search begins at a defined starting point and moves through the search space sequentially (Maier et al., 2010). For example, local search strategies that begin with an empty input set and successively add individual variables are called *forward selection*, while those that start with all possible input variables and successively remove them are known as *backward elimination* (Blum and Langley, 1997). Both of these search strategies are *greedy*, in that they make locally optimal decisions with the hope that a globally optimum solution will be found; and once such a decision has been made, it cannot be undone (i.e. an input added (eliminated) in the early stages of the search can not later be eliminated (added)). *Stepwise selection* involves the successive addition or elimination of input variables, but allows an earlier decision to be retracted, potentially allowing more optimal subsets to be identified. However, decisions made at each step are still only locally optimal and are conditioned on the already selected inputs. *Random* or *probabilistic* search strategies are more adept at finding (near) globally optimum input subsets through their combined use of random subset generation with some mechanism to increase the focus of the search in regions of the search space that lead to good solutions. However, due to their random nature, these strategies search through many more solutions than their sequential counterparts and are, thus, less efficient than sequential search algorithms (Kohavi and John, 1997), yet still provide no guarantee that a globally optimal solution will be found.

2.2. Input subset evaluation

The evaluation step in Figure 1 involves determining which inputs should be added to the 'selected' input set and which should be discarded, based on their relevance. Automatic IVS algorithms can be broadly categorised as *filter*, *wrapper* or *embedded* approaches according to the way in which this input relevance is measured (Guyon and Elisseeff, 2003). Filter IVS methods are described as being *model-free*, as the entire IVS process is independent of the chosen induction or learning algorithm. Both embedded and wrapper IVS approaches, on the other hand, are *model-based*, relying on the performance of a predetermined underlying model to select the most appropriate inputs.

2.2.1. Filter algorithms

Filter techniques rely on the intrinsic properties of the data (e.g. distance, information, dependency, or consistency) to measure the relevance of the input variables, which are then ranked according to some a-priori defined criteria (Liu and Motoda, 1998). As such, filters tend to be computationally simple and scale easily to high-dimensional datasets. However, as filters are independent of the learning algorithm, they have the disadvantage of disregarding how the selected variable subset will affect the performance of the resulting model (Miller, 2002). In addition, they are typically univariate, which has the disadvantage that the relevance between each potential input and the output variable is considered separately (Saeys et al., 2007). Not only does this necessitate that each input-output relationship be evaluated, but it also means that input variable interactions are ignored. Consequently, an input may be found individually to be irrelevant when, in fact, it is very relevant when combined with other inputs.

Linear correlation-based filter algorithms are among the most commonly used IVS methods in environmental studies (see Maier et al. (2010) and Wu et al. (2014) for a review). An example of such an approach is used in the popular Box-Jenkins time-series analysis methodology (Box and Jenkins, 1976), where identification of the most important auto-regressive and moving-average parameters is based on the autocorrelation and partial autocorrelation function. Another popular linear correlation-based approach is the Partial Correlation Input Selection (PCIS) introduced by May et al. (2008a). Yet, despite their popularity and simplicity, such methods are likely to be inappropriate for nonlinear systems.

In recent years, information theoretic-based dependency measures, such as mutual information, have become more popular in IVS, since such measures make no assumptions regarding the structure of the dependence between two variables (i.e. they can estimate both linear and nonlinear dependence) (May et al., 2008a). For example, the Partial Mutual Information (PMI) based IVS method developed by Sharma (2000) and modified by Bowden et al. (2005a); May et al. (2008a); Fernando et al. (2009) has been applied in several studies for identifying the most relevant inputs for predicting rainfall (Sharma et al., 2000), streamflow (Wu et al., 2013), salinity (Bowden et al., 2005b; Fernando et al., 2009), water quality (Kingston et al., 2006; May et al., 2008b) and stormwater runoff (He et al., 2011). Other PMI-based metrics have been recently proposed by Chen et al. (2013) and Sharma and Mehrotra (2014). Another well established filter IVS technique based on mutual information is the minimum redundancy maximum relevance (mRMR) algorithm developed by Peng et al. (2005). A more computationally efficient version of this algorithm was proposed by Hejazi and Cai (2009), who applied it to selecting the most significant inputs to predict daily reservoir releases. Recently, Galelli and Castelletti (2013b) proposed the Iterative Input variable Selection (IIS) algorithm, where a tree-based ranking method is used in place of an information-theoretic measure to estimate the information gained from the data. This algorithm has been employed to select the most relevant input variables for daily streamflow prediction (Galelli and Castelletti, 2013b), prediction of phytoplankton biovolume (Fornarelli et al., 2013), prediction of spatially distributed hydro-ecological data (Surridge et al., 2014) and model reduction problems (Castelletti et al., 2012).

Another popular filter method is the Gamma (or near-neighbour) test (Končar, 1997; Stefánsson et al., 1997), which uses distance, rather than variable dependence or information gain, in the evaluation of input relevance. This method was first employed by Chuzhanova et al. (1998) and has recently become more popular for IVS in the field of environmental modelling. The

Gamma test has been used, for example, to select the best inputs for solar radiation prediction (Remesan et al., 2008; Ahmadi et al., 2009), runoff modelling (Remesan et al., 2009), flood regionalisation (Wan Jaafar et al., 2011; Wan Jaafar and Han, 2012) and downscaling climate variables for precipitation forecasting (Ahmadi and Han, 2013).

2.2.2. Wrapper algorithms

Wrapper methods use the learning algorithm itself as part of the IVS procedure, treating the model as a black box, while searching for the subset of inputs that yields the best model performance (Kohavi and John, 1997). Unlike filter methods, wrapper approaches take into account interactions and dependencies between input variables. However, since the learning algorithm must be called (and calibrated) for each input subset considered, wrapper methods can be very computationally intensive (Blum and Langley, 1997). They are also more susceptible to overfitting than filters, as most of these approaches focus on finding inputs that maximise predictive performance, rather than those that are both relevant and nonredundant. Consequently, it is particularly important when employing wrapper IVS algorithms to adopt an objective function or optimality criterion that penalises model complexity and, hence, overfitting. Wrapper IVS methods tend to be defined by the search strategy employed to generate input subsets. By far the most commonly used wrapper IVS methods in environmental modelling studies are those that involve the sequential (forward, backward or stepwise) selection of inputs (see Olden and Jackson (2000), Mac Nally (2000) and Ssegane et al. (2012) and references therein). In recognising the limitations of sequential search techniques, a number of relatively recent studies have employed random search strategies, such as evolutionary algorithms. For example, Abrahart et al. (1999); Schleiter et al. (2001); Bowden et al. (2005b); D'heygere et al. (2006) and Tirelli et al. (2009) used a genetic algorithm to select the best inputs for rainfall-runoff modelling, water quality modelling and the prediction of species presence/absence using ANNs and decision trees as the induction algorithms.

2.2.3. Embedded algorithms

In embedded IVS techniques, the search for an optimal subset of inputs and calibration of the underlying model occurs concurrently. Thus, the entire IVS process is part of the model training procedure. The basic principle behind embedded IVS algorithms is to specify an objective function for constructing a model consisting of both a goodness-of-fit term and a term that penalises model complexity (Guyon and Elisseeff, 2003). Similar to wrapper methods, embedded techniques account for interactions between inputs and are specific to the chosen learning algorithm, meaning that they can yield high prediction accuracy (the inputs selected will be those that optimise model performance), but at the cost of decreased generalisation on other learning algorithms (Guyon et al., 2006). Unlike wrapper methods, however, only a single model is trained, since the evaluation of input subsets occurs within the training algorithm. Thus, embedded methods are usually far less computationally intensive than wrapper methods (Guyon and Elisseeff, 2003). Furthermore, embedded algorithms consider the impact of each individual input on the performance of the model, and adjust the associated model parameters accordingly. A disadvantage of embedded IVS approaches is the lack of algorithms available for directly minimising the number of input variables for nonlinear predictors (Guyon and Elisseeff, 2003). Embedded algorithms often rely on regularisation methods, which balance model fit and model complexity during the calibration of a model. Using such methods, the penalty term in the objective function is replaced by a regularisation term, which shrinks parameters associated with irrelevant inputs toward zero or sets them equal to zero (Tikka, 2009). There are a number of available regularisation methods, which differ mainly in the way model complexity is measured and, hence, penalised. Ridge regression (Hoerl and Kennard, 1970) and the Lasso algorithm (least absolute shrinkage and selection operator, Tibshirani (1996)) are among the most popular. For an application of such methods to environmental modelling problems, see, for example, Reineking and Schröder (2006); Phatak et al. (2011); Dormann et al. (2013).

2.3. Stopping criterion

The definition of a suitable stopping criterion is another key consideration in IVS as it can significantly influence selection accuracy and computational efficiency. Stopping criteria may be related to either the search strategy or the evaluation method used in the IVS process. For example, stopping criteria related to the search strategy may include whether a predefined number of relevant variables has been selected or whether a predefined number of iterations has been reached (Dash and Liu, 1997). Stopping criteria based on the evaluation of input subsets may include whether the addition or elimination of any inputs produces a better (or worse) subset (using, for example, cross-validation error or parsimonious model selection criteria, such as Akaike's information criterion (AIC) or the Bayesian information criterion (BIC)), or whether selected inputs are relevant, as defined by some threshold value or significance level (using classical statistical tests, such as the t-test, F-test and chi-squared test or resampling methods such as bootstrapping, for example) (Dash and Liu, 1997).

3. IVS Evaluation Framework

The IVS evaluation framework proposed in this paper is designed such that it will be generally applicable to all IVS algorithms, producing easy to interpret and unbiased results and minimising any duplication of effort. As well as aiding comparative analyses of IVS approaches, it should also be useful for investigating parameter effects within individual IVS algorithms or selecting appropriate stopping criteria. As mentioned previously, the basic framework is comprised of three main components: (1) a set of benchmark data; (2) a recommended set of evaluation criteria; and (3) a website for sharing data and results. These are represented in Figure 2 and discussed in detail in the following sections.

3.1. Benchmark datasets

A total of 26 synthetic datasets, summarised in Table 1, were generated for benchmarking the performance of IVS algorithms. These datasets exhibit, to different degrees, the following properties, which are considered to reflect the features of real environmental data: nonlinearity in the underlying function, collinearity amongst input variables, non-Gaussian input/output variables, noise in the output, incomplete input information, and interdependence of input variables. The benchmark datasets were generated to have different sample sizes and dimensionalities to allow scalability and computational efficiency to be assessed on datasets of different sizes. This also enables an investigation of the sensitivities of IVS methods to the relative proportion of irrelevant candidate inputs and of the abilities of IVS methods to identify important input-output relationships within datasets of varying lengths. In Table 1, sample size is denoted by N, K is the number of relevant inputs (those that contain important and non-redundant information about the output) and P is the total number of candidate inputs (the total pool of potentially relevant inputs from which to select from). The $P-K$ candidate inputs that are included in the datasets but contain no (or only redundant) information about the outputs are primarily lagged values of the true inputs or inputs drawn from distributions resembling those of the true inputs. The ratio N/P is also given in Table 1, as this value is indicative of the risk of retaining irrelevant or redundant inputs (a small value of N/P suggests a greater likelihood of overfitting). This risk increases with increasing correlation between the candidate inputs.

While synthetic data may be considered somewhat unrealistic and lacking in substance, their use for IVS algorithm benchmarking is necessary since such data provide the only means for adequately assessing the performance of IVS algorithms using quantitative approaches. Firstly, and most importantly, the use of synthetic data enables selected inputs to be compared to the known set of “true” input variables. This allows ‘selection accuracy’ to be evaluated without relying on prediction accuracy, which can be complicated by a number of factors, including the choice of model, calibration method, error model and calibration criteria, among others. Secondly, with synthetic data it is relatively easy to systematically vary features such as those listed above in order to achieve a balanced design for the comprehensive evaluation of IVS techniques. With real data this would be far more difficult and would rely on methods for quantifying the above properties without knowledge of the true underlying function. While synthetic data may be somewhat simplistic, it would be reasonable to assume that an IVS method which fails to select the correct inputs from data generated from a rather simplistic model would be unlikely to have good selection accuracy when applied to real data. However, in order to ensure that the true characteristics of real environmental data were captured in the benchmark datasets at least to some extent, several of the benchmark sets are only partially synthetic, where the input data are real, while only the outputs are modelled. Whether a benchmark dataset is fully or partially synthetic is also indicated in Table 1.

3.1.1. Properties of the datasets

To define the degrees of noise and nonlinearity associated with the benchmark data, the following DELVE (Rasmussen et al., 1996) definitions were used:

- Noise: The amount of noise in the output is defined as the fraction of the variance that would remain unexplained if a universal approximator, such as an artificial neural network (Cybenko, 1989), were used on an infinite training set. If this residual variance is less than 0.25% the noise is "low". If it lies between 1% and 5% the noise is "moderate". If it exceeds 25% the noise is "high".

- Nonlinearity: A dataset is classified as "fairly linear" if a linear method would leave less than 5% residual variance on an infinite training set.

> It is "highly non-linear" if the linear method would leave more than 40% residual variance.

The degree of collinearity was simply defined according to the number of pairs of candidate inputs with correlation greater than 0.7. This is similar to the definition Amasyali and Ersoy (2009) used for defining the degrees of collinearity associated with their Friedman datasets, which they subsequently donated to the WEKA project. For the purposes of the proposed IVS framework, a dataset is considered to have high collinearity if the number of pairs of candidate inputs with correlation > 0.7 divided by the total number P of candidates is greater than or equal to one, i.e. there are at least as many pairs of highly correlated inputs as there are candidate inputs.

A number of common synthetic data generators have been used for evaluating IVS algorithms in previous environmental modelling studies (see, for example, Sharma (2000); Bowden et al. (2005a); May et al. (2008a); Hejazi and Cai (2009); Fernando et al. (2009) and Galelli and Castelletti (2013b)). These include two linear auto-regressive (AR) models, two nonlinear threshold autoregressive (TAR) models and a nonlinear (NL) model. For consistency, these test problems have been included within this framework. As can be seen in Table 1, the datasets corresponding to these test problems only cover a small subset of the possible combinations of the first four binary dataset properties (i.e. non-Gaussian outputs, nonlinearity, collinearity and noise). Furthermore, N, K and P are fairly uniform amongst these datasets. The extended set of benchmark datasets covers all possible combinations of these dataset properties and includes much greater variation in the size and dimensionality of the datasets. It also covers two special cases where there is incomplete information about the target data and where there is interdependence of inputs (i.e. there exist inputs that are only relevant when combined with other inputs). To account for any variability in algorithm performance that may result from variability in the data, 30 replicates of each benchmark dataset are provided. This enables the statistical significance of comparison results to be considered. These datasets are described in further detail on the framework website.

3.2. Evaluation criteria

There are a number of different factors to consider when evaluating and comparing IVS methods including, for example, accuracy; efficiency; scalability and ease of use. However, objective metrics that enable the general and standardised intercomparison of IVS methods are not necessarily straightforward to define. Firstly, not all of the criteria that are useful for evaluating IVS algorithms can be expressed as quantitative, objective metrics; some can be also qualitative. Secondly, it is important that the number of metrics used for intercomparisons be minimised, while the information gained from them is maximised. However, the metrics used for evaluation and comparison must also provide sufficient information such that any differences in algorithm performance are discernible. Thirdly, the metrics should be simple, easy to compute and interpret and have general applicability across a wide range of different IVS methods. Individual IVS methods may produce information specific to those techniques that can be useful for diagnosing algorithm performance. However, such information will have limited value when compared with algorithms that do not output the same information.

Finally, it is preferable that the metrics can be expressed probabilistically, such that the stability and robustness of algorithms can be assessed and the statistical significance of results computed.

As mentioned, the majority of IVS algorithms seek to balance the trade-off between finding the optimal subset of input variables and computational efficiency. As such, it is important to be able to evaluate algorithms against these criteria in a quantitative and objective way. Details of the proposed quantitative performance measures are given in Section 3.2.1. In contrast, criteria related to an algorithm's ease of use, flexibility or explanation capability, for example, are more difficult to define in such a manner and therefore it is recommended that these be treated as qualitative evaluation criteria. Details of some of these qualitative criteria are given in Section 3.2.2. When assessing and comparing the performance of IVS algorithms, it is recommended that all of the proposed quantitative metrics be used, while the proposed qualitative criteria should be considered and remarked upon where appropriate.

3.2.1. Quantitative metrics

Selection accuracy

A selection accuracy (SA) score which expresses the degree to which a selected input subset matches the true input subset is recommended for use in this framework. The proposed SA score is based on the similarity score proposed by Molina et al. (2002), but unlike the original version, it makes no distinction between irrelevant and redundant inputs and simply treats all unnecessary inputs as extraneous. The proposed SA score is given as follows:

$$SA = \gamma \frac{k}{K} + (1 - \gamma)\left(1 - \frac{p}{P - K}\right) \tag{1}$$

where K is the total number of relevant inputs; k is the number of relevant inputs selected; p is the number of extraneous (irrelevant or redundant) inputs selected; P is the total number of inputs in the candidate input pool (hence $P - K$ is the total number of extraneous inputs) and γ is a weight ranging from 0 to 1, which influences the penalty applied to the selection of extraneous inputs in relation to the gain achieved from each correctly selected input. This score can range from 0 to 1, where $SA = 1$ corresponds to a correctly specified model, while $SA = 0$ corresponds to a completely misspecified model, with no relevant inputs and all extraneous inputs selected. An advantage of this score is that information about the degree to which a model has been correctly or incorrectly specified is combined into a single metric, which makes for the straightforward comparison of IVS algorithm selection accuracy.

The SA score requires the choice of an appropriate value for γ. This choice is subjective and depends on how much one favours accuracy over parsimony, or vice versa. As suggested by Molina et al. (2002), a suitable value for γ should reflect the fact that choosing an extraneous input is usually better than missing a relevant one, which can be achieved by selecting γ such that $\gamma/K > (1 - \gamma)/(P - K)$. However, γ should not be so large that there is no appreciable penalty applied to unnecessary model complexity (for example, for $\gamma = 1$, the selection of extraneous inputs would not be penalised at all). Figure 3 illustrates the effect of γ on the SA score for a theoretical example with 10 inputs in the candidate input pool: 5 relevant and 5 irrelevant (or redundant). As can be seen in Figure 3 (a), when $\gamma = 0.5$ (i.e. $\gamma/K = (1-\gamma)/(P-K)$), the penalty incurred for the selection of extraneous inputs is weighted equally to any improvement in accuracy gained from the selection of relevant inputs (as evidenced by the lack of variation in the SA score in the diagonal direction). This would usually be undesirable given that, in terms of prediction accuracy, the consequences of under-specification (greater bias) are generally more severe than those of over-specification (greater variance). Conversely, when $\gamma = 0.9$ (i.e. $\gamma/K >> (1-\gamma)/(P-K)$), there is very little reduction in the SA score for increasing values of p, as can be seen in Figure 3 (c), indicating that unnecessary complexity is under penalised. As shown in Figure 3 (b), a value of $\gamma = 0.7$ results in the selection of extraneous inputs being appreciably penalised; however, missing a relevant input is assigned greater importance than the selection of an irrelevant or redundant input (as evidenced by the variability in the SA score in both the vertical and diagonal directions). For all of the benchmark datasets included in the proposed IVS evaluation framework, a value of $\gamma = 0.7$ satisfies $\gamma/K > (1-\gamma)/(P-K)$, while still being sufficiently less than 1 to appropriately penalise unnecessary complexity.

While values of $SA < 1$ denote over- or under-specification, a limitation of the SA score is that it does not indicate where the selected input subset is deficient; for example, whether too many or too few inputs have been selected. To overcome this limitation, the SA score given by eq. 1 can be broken into two sub-scores:

$$SA_c = \frac{k}{K} \tag{2}$$

$$SA_e = 1 - \frac{p}{P-K} \tag{3}$$

where SA_c indicates the proportion of correct inputs that have been selected and SA_e is based on the proportion of extraneous inputs that has been selected. Unlike the overall SA score given by eq. 1, these sub-scores do not trade off one measure of accuracy against another; therefore, they do not require the γ parameter. Both of these terms can range from 0 to 1, where a value closer to 1 denotes a better model. In particular, the following combinations of SA_c and SA_e are relevant to the analysis of algorithm performance:

- $SA_c = 1$ and $SA_e = 1$, i.e. perfect specification ($SA = 1$);
- $SA_c = 1$ and $SA_e < 1$, i.e. over-specification of some extraneous inputs ($SA < 1$);
- $SA_c < 1$ and $SA_e = 1$, i.e. under-specification of relevant inputs ($SA < 1$) (according to the definitions used by May et al. (2008a) and Galelli and Castelletti (2013b));
- $SA_c < 1$ and $SA_e < 1$, i.e. under-specification of relevant inputs and over-specification of some extraneous inputs ($SA < 1$).

The advantage of these scores is that they express the degree to which a model is over- or under-specified, which is important for differentiating between IVS algorithm results. Furthermore, the sub-scores can be useful for investigating parameter effects within individual IVS algorithms. For example, if a method consistently results in over- or under-specification over a range of different datasets, this may signify that the stopping criterion used to terminate the IVS method is inappropriately penalising model complexity or that the threshold or significance level used to determine the relevance of an input has been inappropriately set or computed.

Computational efficiency

Under the proposed framework, two quantitative measures of computational efficiency are recommended. The first is total run-time, namely the time required by an IVS algorithm to perform an input selection task. This metric provides a rough estimate of the time it may take to execute a particular algorithm on a given dataset, but depends on the software implementation and the adopted hardware. For this reason, the framework also includes a thorough analysis of computational complexity, which provides a theoretical, platform-independent estimate of the resources needed by an IVS algorithm. In particular, the analysis of computational complexity is determined for each algorithm by evaluating the computational steps involved at each iteration, and is aimed at producing a theoretical classification that estimates the increase in run-time as a function of the input dimensionality N and P. This classification allows calculation of the time that would be required by an IVS algorithm to perform a certain task, and is thus useful when planning the execution of several IVS experiments. Moreover, it allows calculation of the growth rate of the run-time for the worst case scenario (for example, when a forward selection algorithm is run over P iterations to evaluate all candidate inputs).

3.2.2. Qualitative criteria

Where appropriate, it is suggested that the following qualitative assessment criteria be commented on when evaluating IVS algorithms; however, it is not recommended that they be used in the intercomparison of algorithm performance.

1. Ease of use and robustness

   The ease of use of an IVS algorithm relates to how many parameters need to be tuned and how robust the algorithm's performance is, given a default set of parameter values. An IVS algorithm that can be applied without significant user expertise can be highly desirable, particularly for a potential user trying to select the most appropriate IVS algorithm for a problem at hand. Therefore, where possible, it is recommended that some information be provided about which parameters affect the performance of the algorithm and how readily robust values can be selected for these parameters.

2. Explanation capability

   Forward selection IVS methods and algorithms that utilise an input ranking approach provide information about the order of input relevance (i.e. inputs are sorted from most to least relevant) and possibly the relative magnitude of the influence these inputs have on the output. Such information can provide useful insight into the underlying mechanisms by which the data were generated and can be said to have some explanation capability. On the other hand, methods that evaluate the relevance of an input subset as a whole generally do not provide information about the relevance of individual inputs. Thus, while such algorithms may return the optimum input subset for a particular problem, it may be difficult to determine how the individual inputs relate to the output.

3. Flexibility

An IVS algorithm represents a single combination of the three main components shown in Figure 1. However, if it is found that the performance of an algorithm is limited by only one of these components, it would be advantageous if this component could be easily substituted with an alternative. The flexibility of an IVS algorithm relates to the ease with which components of the algorithm can be interchanged with other methods to suit user preferences or to overcome identified shortcomings.

3.3. Framework website

The website (http://ivs4em.deib.polimi.it) is an 'open platform' for sharing datasets, code and results. At the current stage, it contains all 26 benchmark datasets (30 replicates of each), the source code for the four IVS algorithms used in this study and the code for performance evaluation. Moreover, the website includes a functionality for uploading new datasets, algorithms and results to build up a comprehensive database for IVS in environmental modelling.

4. Experimental setup

The proposed framework was used for the evaluation and comparison of four IVS algorithms. The aim was not to provide a definitive answer as to which of the algorithms performed best, but rather to demonstrate the application of the proposed framework and how the results obtained may be used for evaluating and gaining greater insight into algorithm performance. The family of filter methods is represented by the PMI-based input Selection (PMIS) algorithm (in the form modified by May et al. (2008a)), the IIS algorithm (Galelli and Castelletti, 2013b) and the PCIS algorithm introduced by May et al. (2008a). The family of wrapper methods is represented by a GA-ANN algorithm, which adopts a Genetic Algorithm (GA) to select the subset of input variables that maximises the performance of an ANN. Each of the four IVS algorithms considered was implemented on 30 replicates of the 26 benchmark datasets, resulting in 780 runs for each algorithm. For all four algorithms, the same parameter sets were used for all case studies. Details of the parameters used and how their values were determined are given below (the reader is referred to Appendix A for a more detailed description of the algorithms):

- *PMIS and PCIS algorithms.* The PMIS algorithm adopts a forward selection strategy (i.e. one variable is selected at each iteration) based on the estimation of the PMI, which measures the partial dependence between each input variable and the output, conditional on the inputs that have already been selected. To estimate the PMI, the algorithm adopts a kernel density estimator, whose accuracy depends on the value of a smoothing parameter (or bandwidth) λ. Similarly to Sharma (2000) and Bowden et al. (2005a), the Gaussian reference bandwidth (Scott, 1992) is adopted, because of its simplicity and computational efficiency. The calculation of PMI also requires estimation of residual information in the input variables once the effect of the already selected inputs has been taken in consideration: this is done through the identification of a General Regression Neural Network (GRNN), which is a nonlinear and nonparametric regression method (Li et al., 2014). In addition to λ, the other parameter to be set is the stopping criterion, which is based on the coefficient of determination R^2: the PMIS algorithm is stopped when the selection of a further input variable leads to a decrease of R^2 in the underlying GRNN being identified.

The PCIS algorithm adopts the same structure as the PMIS algorithm, but it uses the Pearson correlation coefficient to estimate the strength of the relationship between inputs and output and a multiple linear regression based on least squares in place of PMI and GRNN, respectively. As such, the algorithm dos not require any tuning. PCIS is terminated when the selection of additional inputs no longer results in an improvement (increase) in the Bayesian information criterion (BIC), calculated based on the output variable residual, which provides a trade-off between goodness-of-fit and model complexity.

- *IIS algorithm.* Similarly to PMIS and PCIS, the IIS algorithm proceeds by selecting one input variable at each iteration, but the partial dependence between each input variable and the output relies on a tree-based ranking method, instead of an information-theoretic measure. Furthermore, the relative importance of the p ranked variables is refined through the identification of p Single-Input Single-Output (SISO) models, with the best performing input being added to the set of selected variables. The selection process continues until the accuracy of an underlying Multi-Input Single-Output (MISO) model, evaluated with a k-fold cross-validation, does not significantly improve. The number p and k of SISO models evaluated at each iteration and of the folds used in the k-fold cross-validation process is set to 5, while the algorithm tolerance ε is equal to 0.01, which was empirically found to provide an appropriate balance between accuracy and over fitting (Galelli and Castelletti, 2013b). This means that the IIS algorithm is stopped when the selection of a further variable leads to an increase of R^2 lower than 0.01. Extra-Trees, a regression based method introduced by Geurts et al. (2006), are used for both ranking and modelling. As for the Extra-Trees setting, default values for the parameters M, K and n_{min} are set according to Galelli and Castelletti (2013a). The number M of trees in an ensemble is 500, the number K of alternative cut-directions is equal to the number of candidate inputs and n_{min}, the minimum cardinality for splitting a node, is 5.

- *GA-ANN algorithm.* The ANN is a 1-hidden node multilayer perceptron, with the transfer functions used at the hidden and output nodes being the hyperbolic tangent and linear functions, respectively, which are commonly adopted in environmental modelling problems (Maier and Dandy, 2000). The accuracy of the ANN is measured in terms of out-of-sample AIC, computed using a k-fold cross-validation (with $k = 5$). As for the GA algorithm, the population size and maximum number of generations are equal to 50 and 100,000, respectively. The algorithm is terminated based either on the maximum number of evaluations or convergence of the fitness function (i.e. when the difference in the fitness between one generation and the next remains below a tolerance of 10^{-8} times the previous best value for 20 consecutive generations).

All experiments for PMIS, GA-ANN and PCIS are carried out in the R environment running on 12-core 2.6 GHz CPUs AMD with 2.7 GB RAM per core, while the experiments for the IIS algorithm are carried out using a compiled C++ package running on 8-core 2.2 GHz CPUs Intel Xeon with 8 GB RAM per core.

5. Results

The results obtained by evaluating the four IVS algorithms with the proposed framework are presented and discussed in this section, and organised in terms of selection accuracy, computational efficiency and qualitative criteria in accordance with the framework presented in Section 3.

5.1. Selection accuracy

Each of the proposed selection accuracy scores (i.e. SA, SA_c and SA_e) was computed as the average over the 30 replicates for each of the four IVS algorithm and 26 benchmark datasets.

5.1.1. Overall accuracy

The results for the SA metric reported in Figure 4 show that the PMIS, IIS and GA-ANN algorithms share a similar range of variation for the SA score, which varies from 1 (corresponding to correctly specified models) to about 0.4. On the other hand, the SA values for the PCIS algorithm vary from 1 to 0, where a value of 0 corresponds to a completely mis-specified model. The cases at the extreme ends of these ranges correspond to the AR1 and Miller datasets: all algorithms are capable of selecting the only relevant variable in the AR1 dataset without choosing any other extraneous input, while all algorithms have difficulties in selecting the correct inputs for the Miller dataset, with the PCIS algorithm unable to select any of the correct inputs. Unsurprisingly, the performance of the four algorithms varies depending on dataset properties. For instance, even though the AR1 and AR9_500 datasets are characterised by high noise and high collinearity, the fact that there are only a few relevant inputs (one and three, respectively, see Table 1) and that the N/P ratio is high, makes the input selection task relatively simple for all algorithms, as indicated by the high SA scores (see Figure 4 (AR1)-(AR9_500)). However, a variation in just one of the properties of the data, such as the N/P ratio in the AR9_70 dataset, which differs from the AR9_500 dataset in the number of observations (70 instead of 500), affects the performance of all algorithms.

Furthermore, Figure 4 shows that different values of SA_c and SA_e were obtained for the combination of IVS algorithms and datasets considered. The case of perfect specification (i.e. $SA = 1$) was obtained for a number of different datasets, such as AR1, AR9_500, TAR1 and TAR2, for which all algorithms were capable of only selecting the relevant variables. As commented above, these datasets have high noise and high collinearity, which are somehow 'compensated' for by the N/P ratio, equal to 33.3 (see Table 1). This high ratio between the number of observations N and candidate inputs P allows all algorithms to limit the bias in the estimation of the strength of dependence between inputs and output due to the presence of noise in the observational dataset (*regression dilution*, Frost and Thompson (2000)). A decrease in the number of observations, as in the AR9_70 dataset, had a negative impact on algorithm performance, causing under-specification of relevant inputs (i.e. $SA_c < 1$ and $SA_e = 1$) or both under-specification of relevant and over-specification of extraneous inputs (i.e. $SA_c < 1$ and $SA_e < 1$). For example, use of the PMIS and GA-ANN algorithms did not result in the selection of extraneous inputs ($SA_e = 1$), but over the 30 replicates of the dataset, they show an average SA_c score of about 0.65, indicating that the proportion of correct inputs that has been selected is 65%. This results in a SA score of about 0.75, as shown in Figure 4 (AR9_70). For the same dataset, the IIS algorithm results in a SA_c score of about 0.65, but the overall performance is affected negatively by the over-specification of some extraneous inputs, with SA_e equal to 0.80, which means that the proportion of extraneous inputs that has been selected is 20%. On the other hand, the PCIS algorithm shows perfect specification, with both SA_c and SA_e equal to 1. The over-specification of extraneous inputs (i.e. $SA_c = 1$ and $SA_e < 1$) was observed for the Miller dataset, where all relevant variables were selected when using the IIS and GA-ANN algorithms ($SA_c = 1$), but the only extraneous input was also included, resulting in a value of SA_e equal to 0. Worse results were obtained for the PMIS and PCIS algorithms: the former resulted in a SA_c score of 0.5 and a SA_e score of 0, while the latter performed poorly with respect to both indicators (SA_c and SA_e equal to 0).

5.1.2. Effect of dataset properties on algorithm performance

The impact of dataset properties on IVS algorithm performance in terms of SA, SA_c and SA_e is shown in Figure 5 and described below for each of the datasets in turn. A discussion of the findings is provided in Section 6.

- *AR and TAR datasets.* As discussed in the previous section, these datasets have high noise and collinearity, but a high number N of observations and a reduced number P of total candidate inputs. The ratio N/P is therefore high, allowing all of the algorithms to identify the K relevant inputs ($SA_c = 1$), without including any extraneous inputs ($SA_e = 1$). The only exception is the AR9_70 dataset: in this case the number of observations decreases from 500 to 70, and the N/P ratio from 33.3 to 4.7, with an associated greater likelihood of overfitting. This is empirically demonstrated by the IIS algorithm, which indeed shows a value of SA_e lower than 1.

- *NL datasets.* The NL_500 dataset is not characterised by high noise or high collinearity, but is highly nonlinear and has a non-Gaussian output. This combination seems to have a negative impact on the performance of the PMIS, ANN-GA and PCIS algorithms, as indicated by a significant decrease in SA_c (i.e. greater under-specification of the relevant inputs). The reason for this may be due to the specific characteristics of each algorithm. PMIS, for example, can accurately model nonlinear relationships, but the Gaussian reference bandwidth used in the estimation of the PMI is known to result in reduced performance in cases where the data follow a non-Gaussian distribution (May et al., 2008a). As expected, the performance of the PCIS algorithm is worse than that of the PMIS algorithm, since it is based on partial linear correlation, and is therefore unable to account for the nonlinearity in the data. Finally, the low performance shown by the GA-ANN algorithm may be due to the simple ANN architecture adopted (i.e. 1-hidden node multi-layer perceptron), which might not be fully capable of characterizing the highly nonlinear behaviour of NL_500 dataset.

The IIS algorithm, based on Extra-Trees, is capable of selecting all relevant inputs ($SA_c = 1$) without including any extraneous inputs ($SA_e = 1$), probably because Extra-Trees are capable of accounting for nonlinear relationships and do not require any assumption about the statistical properties of the dataset. However, as seen for the AR9_70 dataset, IIS is sensitive to a decrease in the N/P ratio: while PMIS, ANN-GA and PCIS show similar performance for both the NL_500 and NL_70 datasets, the performance of IIS decreases for the second dataset. Finally the high noise and collinearity (in addition to non-Gaussian output and nonlinearity) characterising the NL2 dataset do not seem to affect the performance of the PMIS, ANN-GA and PCIS algorithms, when compared with the performance of these algorithms on the NL_500 dataset. This seems to be in line with the results for the AR and TAR datasets, and empirically demonstrates that the overall performance of these algorithms is more sensitive to non-Gaussian outputs and/or nonlinearity. This is not the case for the IIS algorithm, the performance of which decreases in terms of both SA_c and SA_e when including high noise and collinearity.

- *Bank datasets.* Unlike any other dataset, these four datasets are characterised by incomplete information about the output data, which seems to have a negative effect on all algorithms, with all values of SA lower than 1. It is important to note that the sub-optimal values in SA are due to sub-optimal values in SA_c only (SA_e is always equal to 1), indicating that the algorithms do not have sufficient information to single out the relevant inputs. The degree of nonlinearity of the underlying function does not seem to significantly affect SA_c (and therefore SA), while the presence of noise (datasets Bank_fh and Bank_nh) appears to have a greater negative impact on the ANN-GA and PCIS algorithms.

- *Friedman datasets.* These datasets are characterised by a combination of nonlinearity, noise, collinearity and different numbers P of candidate inputs. For Friedman_c0_10_m and Friedman_c0_50_m, the PMIS and IIS algorithms result in a value of SA equal to 1, as they are both capable of dealing with nonlinear datasets. On the other hand, the GA-ANN and PCIS algorithms have slightly lower performances, most likely due to their lower efficiency in characterising highly nonlinear functions, as discussed previously. Other potential sources of failure include the parameterization of the GA (e.g. insufficient exploration of the search space or insufficient numbers of generations). The addition of high noise (Friedman_c0_10_h and Friedman_c0_50_h) reduces the selection accuracy of PMIS and IIS, while it does not affect that of GA-ANN and PCIS algorithms. It thus seems that the presence of noise flattens the modelling conditions, with all algorithms showing similar performance. The addition of high collinearity in the Friedman_c25_10_m and Friedman_c25_50_h datasets causes a decrease in SA for all algorithms, with the combination of high noise and high collinearity being particularly critical. It is interesting to note that all algorithms show a value of SA_e equal to 1, with SA_c and SA lower than 1. This means that in the presence of nonlinearity, high noise and collinearity, the algorithms under-specify the relevant inputs, but do not select extraneous inputs.

- *Salinity datasets.* The salinity datasets have a relatively large number of candidate inputs (80 or 160), including time lagged values (5 or 10) of 16 variables, resulting in high collinearity in the input data. The presence of high collinearity and 80 candidate inputs (Salinity_5_l dataset) slightly affects the performance of the PCIS and GA-ANN algorithms ($SA_e < 1$), which is further reduced by the addition of 80 extra inputs (Salinity_10_l dataset). This may be due to the difficulty the GA has in finding the correct combination of input variables among a set of 160 highly correlated inputs. On the contrary, PMIS and IIS are capable of determining all relevant inputs for both datasets. When moderate noise is added to these data (Salinity_5_m and Salinity_10_m datasets), IIS and PCIS maintain the same performance (i.e. perfect specification and a slight over-specification of some extraneous inputs, respectively), while GA-ANN shows a further decrease. The PMIS algorithm shows a pronounced under-specification of relevant inputs ($SA_c < 1$), which may be due to some difficulties in estimating the correct values of PMI in the presence of noise. Finally, the addition of high noise (Salinity_5_h and Salinity_10_h datasets) to both high collinearity and a large number of input variables has a negative effect on all algorithms, which show a decrease in SA_c and hence in SA.

- *Kentucky dataset.* Similar to the Salinity datasets, this dataset is also characterised by a large number P of candidate inputs defined as time lagged values of flow and rainfall observations, causing high collinearity in the data. The presence of random noise is limited, but the output is non-Gaussian. As was found for the NL and Bank datasets, the presence of a non-Gaussian output particularly affects the performance of PMIS and GA-ANN: the former has a SA_c score equal to about 0.50 (meaning that the proportion of correct inputs that has been selected is only 50%), while the latter is capable of selecting the relevant variables, but tends to include extraneous inputs (SA_e equal to 0.80). On the other hand, IIS and PCIS seem to be less affected by the non-Gaussian output.

- *Miller dataset.* This dataset has a non-Gaussian output and three candidate inputs only: x_1 and x_2 have a strong inter-dependency (i.e. they jointly influence the output, while having little influence on the output individually), while the extraneous input x_3 has the highest (spurious) correlation with the output. This last characteristic makes the input selection exercise particularly challenging for forward selection methods (i.e. PMIS, IIS and PCIS) that select only one input at each iteration, as evidenced by SA scores of less than 1. PMIS and PCIS are particularly affected by the inter-dependency. The SA_e score is equal to 0 for both algorithms, meaning that they always select the extraneous input x_3, and the SA_c score is respectively equal to 0.5 and 0, resulting in very low values for the SA score. Surprisingly, the IIS algorithm achieves the same performance as the GA-ANN algorithm (the only wrapper method adopted in this study), with SA_e equal to 0, but SA_c equal to 1. That is both IIS and GA-ANN select all candidate inputs. The unexpectedly good performance of the IIS algorithm may be due to the algorithm tolerance (i.e. $\varepsilon = 0.01$), which could cause a slight tendency to over-specify models and allow the algorithm to select additional variables beyond the first ‘most relevant’ input.

5.1.3. Effect of N and P on algorithm performance

As discussed in the previous section, an increase in the number N of observations increases the information available for the IVS algorithm, thus positively impacting selection accuracy (and vice versa for a decrease in N). As far as the number P of candidates is concerned, an increase in P should increase the overall complexity of the IVS problem, but, in practice, the results show that the correspondence between P and SA is neither univocal nor monotonic. Indeed, different values of SA are found for the same value of P (see, for instance, the AR datasets), and SA does not decrease with P. For example, the average performance of the four algorithms on the Miller dataset ($P = 3$) is lower than the average performance on the Salinity datasets, where P is either 80 or 160. The ratio N/P is probably a better indicator of IVS problem complexity, since its value is indicative of the risk of retaining extraneous inputs (i.e. likelihood of overfitting). High values of this ratio are generally associated with high values of SA, and vice versa. The ratio N/P should be evaluated together with the other properties of a dataset, and collinearity in particular, as the likelihood of overfitting increases with the correlation between the candidate inputs. Indeed, it can be observed that when the ratio N/P falls below a critical threshold (about 5), most of the considered IVS algorithms have a SA score of less than 1. In other words, the ratio N/P appears to be a ‘limiting factor’, where low values limit the capability of any IVS algorithm to select the relevant input variables.

5.2. Computational efficiency

From Table 2, it is evident that when the IVS algorithms are applied to the benchmark datasets, two different behaviours, in terms of average run-time, are observed. First, the PMIS and IIS algorithms show similar run-times, especially for the first eighteen datasets. These datasets are characterised by a limited number of observations and candidate inputs N and P (ranging from 70 to 500 and 10 to 50, respectively), so both algorithms perform the input selection task in a time ranging from a few seconds to about one minute. Such computational efficiency is due to their forward selection nature (i.e. one variable is selected at each iteration), which only requires a small number of iterations and therefore few calibrations of GRNNs and Extra-Trees. On the other hand, the application of these two algorithms to the Salinity and Kentucky datasets, which are characterised by much larger numbers of samples and candidate inputs, increases the run-time up to about 2.5 hours, but with IIS faster than PMIS. Apart from the adopted server and the specific implementation (a C++ executable may be faster than the R environment), the reason behind this difference stands in the good scalability of Extra-Trees to large datasets. Indeed, as further discussed in Appendix B, the Extra-Trees run-time increases linearly with P and superlinearly with N, while the time required to calibrate a GRNN does not scale well to large datasets. Moreover, Extra-Trees are used in the IIS algorithm a smaller number of times than GRNNs in the PCIS algorithm, resulting in a run-time order that is quadratic in P and superlinear in N, whilst the run-time order of PMIS is $O(P^4 \cdot N^2 + P^5)$ (Table 3). The PCIS algorithm has the smallest run-time, which varies from a few tenths of a second to about 3 minutes for the most complex datasets. Similarly to PMIS and IIS, this algorithm also has a forward selection nature, requiring only few estimates of the Pearson correlation coefficient and calibrations of a linear regression model. The latter have a high computational efficiency, thus reducing PCIS run-time in comparison with that for PMIS and IIS. Apart from this specific difference, these three filters are characterised by similar growth rates of the run-time: PMIS and PCIS find a solution in an expected number of $O(P^4 \cdot N^2 + P^5)$ and $O(P^4 \cdot N + P^5)$ steps, respectively, while IIS requires $O(T \cdot P^2)$ steps (where T is the time required for cross-validating and ensemble of Extra-Trees). That is, the growth rate for the three filters is polynomial in P and N (see Appendix B for further details).

Second, the GA-ANN algorithm has a run-time that is almost two orders of magnitude higher than that of PMIS and IIS. This is due to its wrapper nature, which requires several ANN calibration runs at each iteration of the GA. As a consequence, the application of this algorithm to the first eighteen datasets takes a time ranging from a few minutes to about 30 minutes, while the time required to analyse the largest datasets (e.g. Salinity_10_l) takes almost 80 hours. Notice that such run-times are obtained by adopting a relatively-simple ANN, i.e. a 1-hidden node multilayer perceptron, so higher run-times would be required if a more complex network architecture were adopted. Unlike PMIS, IIS and PCIS, it is much harder to determine the growth rate of the run-time of the GA-ANN algorithm, since the GA is a stochastic optimisation algorithm, the computational complexity of which depends on different factors, such as the (randomly generated) initial population. In general, it can be assumed that when the GA-ANN algorithm is run over the whole number of generations, the run-time is proportional to $G \cdot I$, where G and I represent the number of generations and the population size, respectively.

5.3. Qualitative criteria

In addition to the quantitative criteria, the four IVS algorithms are also assessed in terms of ease of use, explanation capability and flexibility.

- *Ease of use.* The three filter algorithms adopted in this study (i.e. PMIS, IIS and PCIS) are easy to use, in terms of both the number of parameters to be tuned and the robustness of the results with respect to the default set of parameter values. The PMIS and PCIS algorithms require the tuning of two and one parameter, respectively (see Section 4). The IIS algorithm has a larger number of parameters, but these can be easily tuned via a trial-and-error procedure (Galelli and Castelletti, 2013b). Furthermore, the accuracy of the PMIS, IIS and PCIS algorithms is very robust with respect to the adopted (default) parameterization. On the other hand, the accuracy of the GA-ANN algorithm is sensitive to the parameterization of both GA and ANN. In particular, the results described in Section 5 show that the selection of the ANN architecture appears to be critical. This is a common feature of wrapper algorithms, which generally require an accurate, dataset-dependent tuning of the underlying model (Guyon and Elisseeff, 2003).

- *Explanation capability.* Another advantage of the PMIS, IIS and PCIS algorithms, and of filter methods in general, is that they provide information about the relative importance of each selected input. The same information may be obtained from a wrapper algorithm, but this requires an ex-post interpretation of the data-driven model structure.

- *Flexibility.* The PCIS and PMIS algorithms are very flexible, as their structure is identical, but with (1) the PMI criterion used in place of the partial linear correlation coefficient, and (2) a GRNN in place of a linear model. On the other hand, the IIS algorithm relies on Extra-Trees for both ranking the candidate input variables and assessing the significance of the selected ones. This reduces the overall flexibility of the algorithm, although it may increase the accuracy of the underlying model. Finally, it could be argued that the GA-ANN algorithm exhibits a high level of flexibility, as both the optimization algorithm (GA) and the underlying data-driven model (ANN) can be interchanged with other methods. However, such high flexibility comes at a price, since the adoption of complex, highly parameterized components may negatively impact the algorithm's easy of use, explanation capability and computational efficiency.

6. Discussion, recommendations and issues in environmental modelling

6.1. Discussion

The effect of the properties of the benchmark datasets on IVS algorithm performance can be summarised as follows:

- *Non-Gaussian Output.* Unsurprisingly, it is found that a non-Gaussian output inhibits a high level of performance for the PMIS algorithm, which assumes Gaussian data when estimating the PMI. This tendency is accentuated when this property is combined with other limiting properties (e.g. incomplete information or inter-dependency as in the Bank or Miller datasets). The IIS and PCIS algorithms, which do not assume Gaussian data, appear to be unaffected by a non-Gaussian output, and can indeed achieve a perfect specification, with $SA = 1$, when this property is not combined with other limiting properties (e.g. NL_500 for IIS and Kentucky dataset for PCIS).

- *High Nonlinearity.* The presence of a highly nonlinear relationship between inputs and output can be effectively handled by IIS and PMIS, which rely on regression methods capable of characterising such relationships (Extra-Trees and GRNN, respectively). This is demonstrated on the Friedman_c0_10m and Friedman_c0_50m datasets, which are characterised by this property only. As mentioned above, IIS can simultaneously deal with non-Gaussian outputs and highly nonlinear input-output relationships, if enough observations are available (see NL_500 and NL_70 datasets). On the other hand, both GA-ANN and PCIS are affected by highly nonlinear datasets. Indeed, the former relies on a simple 1-hidden node ANN, which is effective with weakly nonlinear functions only, while the latter is based on linear partial correlation.

- *High Noise.* The presence of high noise affects the performance of all IVS algorithms, but only when combined with certain other properties of the data. For example, the combination of high noise with high nonlinearity decreases the signal-to-noise ratio, which deteriorates the performance of both the PMIS and IIS algorithms on the Friedman datasets. Similarly, the presence of high noise is critical when evaluated in relation to the number N of observations. Indeed, a decrease in N affects the signal-to-noise ratio, as illustrated by the deterioration in performance when reducing the number of observations from 500 to 70 in the AR_70 dataset.

- *High Collinearity.* This property is normally due to the presence of time lagged values of some input variables, as in the AR, TAR, Salinity and Kentucky datasets. Similar to the presence of high noise, collinearity can also be effectively handled by all algorithms, even in the presence of many inputs, such as in the Salinity_5_l and Salinity_10_l datasets, where P is equal to 80 and 160, respectively. However, when high noise is introduced into the dataset, good performance can only be achieved if the number P of candidate inputs is limited, as is the case for the AR and TAR datasets.

- *Inter-dependency.* As explained in Section 5, inter-dependency between input variables (Miller dataset) is generally problematic for filter, forward selection methods that evaluate one input variable at each iteration. Indeed, PMIS and PCIS exhibit low accuracy on the Miller dataset (SA equal to about 0.40 and 0, respectively), while IIS is capable of achieving greater accuracy, probably because of the pre-selected exit condition. Unsurprisingly, the GA-ANN algorithm, the only wrapper method considered in this study, achieves the best performance.

- *Incomplete information.* This property has a significant impact on all IVS algorithms, as evidenced by the inability of any of the algorithms to select the relevant input variables on the Bank dataset.
- N, P and N/P. The ratio N/P is a further limiting factor that reduces the accuracy of all algorithms when it drops to values below 5. In particular, IIS seems to be more sensitive to drops in the ratio N/P, as found for the AR_70 and the NL_70 datasets. In addition, both N and P have a strong impact on the computational performance of IVS algorithms. While filter methods, such as PMIS and IIS, exhibit good scalability with respect to input dataset dimensionality, the run-time of wrapper methods is particularly sensitive to an increase in N and P, up to the point where their adoption becomes impractical for large datasets. For example, the GA-ANN algorithm requires more than 3 days of computation for solving an input selection problem with 4115 observations and 160 candidate input variables, while PMIS and IIS require 2.5 and 1.5 hours, respectively.

Finally, it is interesting to highlight that the presence of two properties of the data, namely inter-dependency and incomplete information, have a strong impact on the accuracy of all algorithms, irrespective of the presence/absence of other properties. Non-Gaussian output and a highly nonlinear input-output relationship can only be fully handled by some algorithms (i.e. IIS and PCIS, and PMIS and IIS respectively), and they require the adoption of specific metrics and regression methods that can deal with such properties. Finally, high noise and high collinearity are not a problem per se, but their presence adds a further level of complication when they are combined with other properties, such as non-Gaussian outputs or nonlinear datasets.

6.2. Recommendations

Although the results reported here are primarily discussed for the purposes of the demonstration of the framework, they can be used to develop some preliminary guidelines in relation to the relative importance of different properties of the data and the corresponding performance of the four IVS algorithms.

- The presence of a *non-Gaussian* output is a potential limiting factor, which requires the adoption of IVS algorithms that do not assume Gaussian data when estimating the relative importance of each input or when building a regression model. A similar recommendation is valid in case of a *highly nonlinear* relationship between input and output. The only algorithm capable of selecting the correct inputs in the presence of both properties is IIS.

- As discussed in Section 6.1, the presence of *high noise* is problematic only when combined with certain other properties of the data (e.g. non-Gaussian output or inter-dependency), so the choice of the most appropriate IVS algorithm should be based on its capability of dealing with such properties. This guideline still holds in the presence of *high collinearity.*

- In the presence of *inter-dependency* between input variables, it is advisable to adopt a wrapper method, which can handle this property through the selection of multiple inputs at each iteration. This cannot be done by filters, unless the candidate input set is enlarged to include features that are combinations of the original input variables.

- *Incomplete information* within the dataset affects the performance of all IVS algorithms, so the most suitable algorithm should be chosen according to its capability of dealing with the other properties characterising the dataset in hand.

- The *ratio* N/P is a limiting factor when it drops to values approximately below 5: in this case it is recommendable to use IVS algorithms that rely on simple metrics and regression techniques, such as PCIS, which is based on the Pearson correlation coefficient and linear regression. Indeed, more advanced algorithms (e.g. PMIS and IIS) require the identification of complex data-driven models, whose performance increases in the presence of more observations.

- While the properties and recommendations above should be considered before any IVS experiment, the size of the dataset and the computational performance of an IVS algorithm matter only in the presence of limited computational resources (or limited time available to conduct the experiments). In general, if the maximum computing time that is available for each experiment is in the order of a few hours, it is advisable to adopt a filter method. On there other hand, if more or unlimited time is available, a wrapper can be a viable solution. However, it must be remembered that the tuning of a wrapper is a time consuming task, which requires an accurate parameterisation of both the optimisation algorithm and the architecture of the data-driven model.

6.3. Issues in environmental modelling

Unlike for the synthetic data considered here, a key aspect of real-world environmental modelling problems is that the true underlying function is unknown, and IVS is thus used to reduce the uncertainty in the model development process by selecting a subset of relevant and non-redundant input variables. This opens some relevant theoretical and practical issues that are highlighted below:

- Most of the IVS algorithms currently available select a unique subset of input variables, although the structural uncertainty in the inputs to be used often results in the possibility of choosing different, but equally informative, subsets. An attempt to account for this issue was recently made by Sharma and Chowdhury (2011), who proposed a PMI-based heuristic approach to select five different subsets of predictors in the context of medium-term hydro-climatic forecasting. The approach ensures that the cross-dependence between these subsets is limited, while the predictions of the resulting models are eventually combined with ensemble averaging.

- In many practical situations, input variables can be characterised by errors, due to, for example, the interpolation of data in space and time or the conversion of point measurement into areal values. Whilst methods exist for assessing the impact of input errors on parameter estimation procedures (Chowdhury and Sharma, 2007; Woldemeskel et al., 2012), IVS algorithms cannot take into account the change in the uncertainty associated with the different inputs.

- A benefit of IVS is the improvement in the performance of the model being identified. Although the manner in which such performance is characterised depends on the specific domain of interest and the model objectives (Jakeman et al., 2006), two important aspects should always be considered when dealing with quantitative testing. First, the use of observational data for comparison must rely on appropriate data-division methods, such as cross-validation or bootstrapping, that allow for testing the ability of the model to generalise. Data division can account for both temporal and spatial dimensions, so it is suitable for spatial modelling as well (see Chowdhury and Sharma (2009) for an application to hydrological modelling problems). Second, an exhaustive quantitative evaluation should rely on a set of metrics focussing on different aspects in order to test the ability of the model in reproducing all the important features of the system. The reader is referred to Bennett et al. (2013) for a comprehensive review of techniques available for both data-division and quantitative evaluation, and to Robson (2014) for a more general assessment of environmental models.

7. Closure

In this work we present a framework for the comparative analysis of IVS algorithms in environmental modelling problems. The framework consists of a set of benchmark datasets with the typical properties of environmental data, a recommended set of evaluation criteria and a website for sharing data, code and results. Since the data and criteria proposed here cannot exhaustively represent all modelling contexts encountered by developers and users, it is hoped that the presence of a dedicated website will increase the flexibility of this framework and facilitate collaboration between researchers. For example, the benchmark datasets are currently limited to those that have both continuous input and output variables; however, it is intended that this set will be extended to include datasets comprised of nominal and categorical variables. In addition, as this framework is applied to an increasing number of IVS algorithms and datasets, it is hoped that guidelines for the adoption of the most appropriate IVS algorithms for datasets with particular properties can be developed.

Acknowledgements

This work was supported by the Goyder Institute for Water Research, Project E.2.4. The first author is currently supported by the SRG ESD 2013 061 Start-up Research project. The authors are grateful to Marcello Restelli for his support in the analysis of the computational efficiency of the four IVS algorithms.

References

Abrahart, R.J., Anctil, F., Coulibaly, P., Dawson, C.W., Mount, N.J., See, L.M., Shamseldin, A.Y., Solomatine, D.P., Toth, E., Wilby, R.L., 2012. Two decades of anarchy? Emerging themes and outstanding challenges for neural network river forecasting. Progress in Physical Geography 36, 480–513.

Abrahart, R.J., See, L., Kneale, P.E., 1999. Using pruning algorithms and genetic algorithms to optimise network architectures and forecasting inputs in a neural network rainfall-runoff model. Journal of Hydroinformatics 1, 103–114.

Ahmadi, A., Han, D., 2013. Identification of dominant sources of sea level pressure for precipitation forecasting over wales. Journal of Hydroinformatics 15, 1002–1021.

Ahmadi, A., Han, D., Karamouz, M., Remesan, R., 2009. Input data selection for solar radiation estimation. Hydrological Processes 23, 2754–2764.

Allen, D., 1974. The relationship between variable selection and data augmentation and a method for prediction. Technometrics 16, 125–127.

Amasyali, M.F., Ersoy, O.K., 2009. A Study of Meta Learning for Regression. ECE Technical Reports.

Araújo, M.B., Guisan, A., 2006. Five (or so) challenges for species distribution modelling. Journal of Biogeography 33, 1677–1688.

Belisle, C.J.P., 1992. Convergence theorems for a class of simulated annealing algorithms on Rd. Journal of Applied Probability 29, 885–895.

Bennett, N.D., Croke, B.F., Guariso, G., Guillaume, J.H., Hamilton, S.H., Jakeman, A.J., Marsili-Libelli, S., Newham, L.T., Norton, J.P., Perrin, C., et al., 2013. Characterising performance of environmental models. Environmental Modelling & Software 40, 1–20.

Blum, A.L., Langley, P., 1997. Selection of relevant features and examples in machine learning. Artificial Intelligence 97, 245–271.

Bowden, G.J., Maier, H.R., Dandy, G.C., 2005a. Input determination for neural network models in water resources applications. Part 1. Background and methodology. Journal of Hydrology 301, 75–92.

Bowden, G.J., Maier, H.R., Dandy, G.C., 2005b. Input determination for neural network models in water resources applications. Part 2. Case study: forecasting salinity in a river. Journal of Hydrology 301, 93–107.

Box, G.E.P., Jenkins, G.M., 1976. Time Series Analysis, Forecasting and Control. Holden-Day Inc., San Francisco, CA.

Castelletti, A., Galelli, S., Restelli, S., Soncini-Sessa, R., 2012. Data-driven dynamic emulation modelling for the optimal management of environmental systems. Environmental Modelling & Software 34, 30–43.

Chen, L., Ye, L., Singh, V., Zhou, J., Guo, S., 2013. Determination of input for artificial neural networks for flood forecasting using the copula entropy method. Journal of Hydrologic Engineering -, –. doi:10.1061/(ASCE)HE.1943-5584.0000932.

Chowdhury, S., Sharma, A., 2007. Mitigating parameter bias in hydrological modelling due to uncertainty in covariates. Journal of Hydrology 340, 197–204.

Chowdhury, S., Sharma, A., 2009. Multisite seasonal forecast of arid river flows using a dynamic model combination approach. Water resources research 45.

Chuzhanova, N.A., Jones, A.J., Margetts, S., 1998. Feature selection for genetic sequence classification. Bioinformatics 14, 139–143.

Cybenko, G., 1989. Approximation by superpositions of a sigmoidal function. Mathematics of control, signals and systems 2, 303–314.

Dash, M., Liu, H., 1997. Feature selection for classification. Intelligent Data Analysis 1, 131–156.

D'heygere, T., Goethals, P.L.M., De Pauw, N., 2006. Genetic algorithms for optimisation of predictive ecosystems models based on decision trees and neural networks. Ecological Modelling 195, 20–29.

Dormann, C.F., Elith, J., Bacher, S., Buchmann, C., Carl, G., Carré, G., Marquéz, J.R.G., Gruber, B., Lafourcade, B., Leitão, P.J., Münkemüller, T., McClean, C., Osborne, P.E., Reineking, B., Schröder, B., Skidmore, A.K., Zurell, D., Lautenbach, S., 2013. Collinearity: a review of methods to deal with it and a simulation study evaluating their performance. Ecography 36, 27–46.

Elith, J., Leathwick, J.R., 2009. Species distribution models: Ecological explanation and prediction across space and time. Annual Review of Ecology, Evolution, and Systematics 40, 677–697.

Elshorbagy, A., Corzo, G., Srinivasulu, S., Solomatine, D.P., 2010. Experimental investigation of the predictive capabilities of data driven modeling techniques in hydrology - part 1: Concepts and methodology. Hydrology and Earth System Sciences 14, 1931–1941.

Fernando, T.M.K.G., Maier, H.R., Dandy, G.C., 2009. Selection of input variables for data driven models: An average shifted histogram partial mutual information estimator approach. Journal of Hydrology 367, 165–176.

Fornarelli, R., Galelli, S., Castelletti, A., Antenucci, J.P., Marti, C.L., 2013. An empirical modeling approach to predict and understand phytoplankton dynamics in a reservoir affected by interbasin water transfers. Water Resources Research 49, 3626–3641.

Frost, C., Thompson, S., 2000. Correcting for regression dilution bias: comparison of methods for a single predictor variable. Journal of the Royal Statistical Society Series A 163, 173–190.

Galelli, S., 2010. Dealing with complexity and dimensionality in water resources management. Ph.D. thesis. Politecnico di Milano, Italy.

Galelli, S., Castelletti, A., 2013a. Assessing the predictive capability of randomized tree-based ensembles in streamflow modelling. Hydrology and Earth System Sciences 17, 2669–2684.

Galelli, S., Castelletti, A., 2013b. Tree-based iterative input variable selection for hydrological modelling. Water Resources Research 49, 4295–4310.

Geurts, P., Ernst, D., Wehenkel, L., 2006. Extremely randomized trees. Machine Learning 63, 3 – 42.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Pub. Co., Reading, MA.

Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. The Journal of Machine Learning Research 3, 1157–1182.

Guyon, I., Gunn., S., Nikravesh, M., Zadeh, L., 2006. Feature Extraction: Foundations and Applications. Studies in Fuzziness and Soft Computing, Springer, Berlin, D.

He, J., Valeo, C., Chu, A., Neumann, N.F., 2011. Prediction of event-based stormwater runoff quantity and quality by ANNs developed using pmi-based input selection. Journal of Hydrology 400, 10–23.

Hejazi, M.I., Cai, X., 2009. Input variable selection for water resources systems using a modified minimum redundancy maximum relevance (mmrmr) algorithm. Advances in Water Resources 32, 582–593.

Hoerl, A.E., Kennard, R.W., 1970. Ridge regression: biased estimation for nonorthogonal problems. Technometrics 12, 55–67.

Jakeman, A.J., Letcher, R.A., Norton, J.P., 2006. Ten iterative steps in development and evaluation of environmental models. Environmental Modelling & Software 21, 602–614.

Kingston, G., Maier, H., Lambert, M., 2006. Forecasting cyanobacteria with bayesian and deterministic artificial neural networks, in: IEEE World Congress of Computational Intelligence, AAAI Press. pp. 129–134.

Kohavi, R., John, G.H., 1997. Wrappers for feature subset selection. Artificial Intelligence 97, 273–324.

Končar, N., 1997. Optimisation methodologies for direct inverse neurocontrol. Ph.D. thesis. Imperial College, London.

Li, X., Zecchin, A.C., Maier, H.R., 2014. Selection of smoothing parameter estimators for general regression neural networks applications to hydrological and water resources modelling. Environmental Modelling & Software 59, 162 – 186.

Liu, H., Motoda, H., 1998. Feature Selection for Knowledge Discovery and Data Mining. volume 454 of *The Springer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Boston, MA.

Mac Nally, R., 2000. Regression and model-building in conservation biology, biogeography and ecology: The distinction between – and reconciliation of – 'predictive' and 'explanatory' models. Biodiversity & Conservation 9, 655–671.

Maier, H.R., Dandy, G.C., 2000. Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. Environmental Modelling and Software 15, 101–124.

Maier, H.R., Jain, A., Dandy, G.C., Sudheer, K.P., 2010. Methods used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions. Environmental Modelling & Software 25, 891–909.

May, R., Dandy, G., Maier, H., 2011. Review of input variable selection methods for artificial neural networks. Artificial neural networks–Methodological advances and biomedical applications , 19–44.

May, R., Dandy, G., Maier, H., Nixon, J., 2008b. Application of partial mutual information variable selection to ANN forecasting of water quality in water distribution systems. Environmental Modelling & Software 23, 1289–1299.

May, R.J., Maier, H.R., Dandy, G.C., Fernando, T.M.K.G., 2008a. Nonlinear variable selection for artificial neural networks using partial mutual information. Environmental Modelling & Software 23, 1312–1326.

Miller, A.J., 2002. Subset Selection in Regression. Monographs on Statistics and Applied Probability. 2nd ed., Chapman & Hall / CRC.

Molina, L.C., Belanche, L., Nebot, A., 2002. Feature selection algorithms: a survey and experimental evaluation, in: The 2002 IEEE International Conference on Data Mining, pp. 306–313.

Olden, J.D., Jackson, D.A., 2000. Torturing data for the sake of generality: How valid are our regression models? Ecoscience 7, 501–510.

Peng, H., Fulmi, L., Ding, C., 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. Pattern Analysis and Machine Intelligence, IEEE Transactions on 27, 1226–1238.

Phatak, A., Bates, B.C., Charles, S.P., 2011. Statistical downscaling of rainfall data using sparse variable selection methods. Environmental Modelling & Software 26, 1363–1371.

Rasmussen, C.E., Neal, R.M., Hinton, G.E., van Camp, D., Revow, M., Ghahramani, Z., Kustra, R., Tibshirani, R., 1996. Data for evaluating learning in valid experiments (delve). URL: http://www.cs.toronto.edu/ delve/.

Reineking, B., Schröder, B., 2006. Constrain to perform: regularization of habitat models. Ecological Modelling 193, 675–690.

Remesan, R., Shamim, M.A., Han, D., 2008. Model data selection using gamma test for daily solar radiation estimation. Hydrological Processes 22, 4301–4309.

Remesan, R., Shamim, M.A., Han, D., Mathew, J., 2009. Runoff prediction using an integrated hybrid modelling scheme. Journal of Hydrology 372, 48–60.

Robson, B.J., 2014. State of the art in modelling of phosphorus in aquatic systems: Review, criticisms and commentary. Environmental Modelling & Software doi:10.1016/j.envsoft.2014.01.012.

Saeys, Y., Inza, I., Larrañaga, P., 2007. A review of feature selection techniques in bioinformatics. Bioinformatics 23, 2507–2517.

Schleiter, I.M., Obach, M., Borchardt, D., Werner, H., 2001. Bioindication of chemical and hydromorphological habitat characteristics with benthic macro-invertebrates based on artificial neural networks. Aquatic Ecology 35, 147–158.

Scott, D.W., 1992. Multivariate Density Estimation: Theory, Practice, and Visualization. Wiley Series in Probability and Statistics, John Wiley and Sons, Inc., New York.

Sharma, A., 2000. Seasonal to interannual rainfall probabilistic forecasts for improved water supply management: Part 1 - a strategy for system predictor identification. Journal of Hydrology 239, 232–239.

Sharma, A., Chowdhury, S., 2011. Coping with model structural uncertainty in medium-term hydro-climatic forecasting. Hydrology Research 42, 113–127.

Sharma, A., Luk, K.C., Cordery, I., Lall, U., 2000. Seasonal to interannual rainfall probabilistic forecasts for improved water supply management: Part 2 — predictor identification of quarterly rainfall using ocean-atmosphere information. Journal of Hydrology 239, 240–248.

Sharma, A., Mehrotra, R., 2014. An information theoretic alternative to model a natural system using observational information alone. Water Resources Research 50, 650–660.

Silverman, B.W., 1986. Density Estimation for Statistics and Data Analysis. Mongraphs on Statistics and Applied Probability, Chapman and Hall, London.

Ssegane, H., Tollner, E.W., Mohamoud, Y.M., Rasmussen, T.C., Dowd, J.F., 2012. Advances in variable selection methods I: Causal selection methods versus stepwise regression and principal component analysis on data of known and unknown functional relationships. Journal of Hydrology 438–439, 16–25.

Stefánsson, A., Končar, N., Jones, A.J., 1997. A note on the gamma test. Neural Computing & Applications 5, 131–133.

Surridge, B., Bizzi, S., Castelletti, A., 2014. Coupling explanation and prediction in the modelling of hydroecological data. Environmental Modelling & Software -, –. doi:10.1016/j.envsoft.2014.02.012.

Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological) 58, 267–288.

Tikka, J., 2009. Simultaneous input variable and basis function selection for rbf networks. Neurocomputing 72, 2649–2658.

Tirelli, T., Pessani, D., 2011. Importance of feature selection in decision-tree and artificial-neural-network ecological applications. alburnus alburnus alborella: A practical example. Ecological Informatics 6, 309–315.

Tirelli, T., Pozzi, L., Pessani, D., 2009. Use of different approaches to model presence/absence of salmo marmoratus in piedmont (northwestern italy). Ecological Informatics 4, 234–242.

Wan Jaafar, W.Z., Han, D., 2012. Variable selection using the gamma test forward and backward selections. Journal of Hydrologic Engineering 17, 182–190.

Wan Jaafar, W.Z., Liu, J., Han, D., 2011. Input variable selection for median flood regionalization. Water Resources Research 47, W07503.

Woldemeskel, F., Sharma, A., Sivakumar, B., Mehrotra, R., 2012. An error estimation method for precipitation and temperature projections for future climates. Journal of Geophysical Research: Atmospheres (1984–2012) 117.

Wu, W., Dandy, G., Maier, H., 2014. Protocol for developing ann models and its application to the assessment of the quality of the ann model development process in drinking water quality modeling. Environmental Modelling & Software 54, 108–127.

Wu, W., May, R., Maier, H., Dandy, G., 2013. A benchmarking approach for comparing data splitting methods for modeling water resources parameters using artificial neural networks. Water Resources Research 49, 7598–7614.

Yang, J.B., Ong, C.J., 2011. Feature selection using probabilistic prediction of support vector regression. Neural Networks, IEEE Transactions on 22, 954–962.

Appendix A

A.1 PMIS algorithm

The PMIS algorithm is a filter IVS method developed by Sharma (2000) and later modified by Bowden et al. (2005a) and May et al. (2008a), where the relevance of potential inputs is evaluated based on the mutual information (MI) between each input variable and the output. While MI is a useful measure of dependence between a potential input variable $\mathbf{x}$ and a dependent variable $\mathbf{y}$, it cannot account for redundancy in the candidate input pool, $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_P\}$. To account for such redundancy, the PMI criterion, which measures the *partial* dependence between a potential input variable and the output, conditional on any inputs that have already been selected, is instead used in this algorithm. This criterion is analogous to the partial correlation coefficient and can be formulated as:

$$\text{PMI} = \frac{1}{N}\sum_{i=1}^{N}\log\left[\frac{f\left(x'_i, y'_i\right)}{f\left(x'_i\right) f\left(y'_i\right)}\right] \tag{4}$$

where

$$\mathbf{x}' = \mathbf{x} - E[\mathbf{x}|Z]; \text{and } \mathbf{y}' = \mathbf{y} - E[\mathbf{y}|Z] \tag{5}$$

represent the residual information in variables $\mathbf{x}$ and $\mathbf{y}$ once the effect of the already selected inputs, Z, has been taken into consideration. In eq. (4), x'_i and y'_i are the i-th residuals in a sample dataset of size N and $f\left(x'_i\right)$, $f\left(y'_i\right)$ and $f\left(x'_i, y'_i\right)$ are the respective marginal (univariate) and joint (bivariate) probability density functions (pdfs).

Calculation of the PMI criterion in eq. (4) requires estimation of the marginal and joint pdfs of $\mathbf{x}$ and $\mathbf{y}$. For the PMIS algorithm, Sharma (2000) proposed the use of a non-parametric kernel density estimation based on the Gaussian kernel function (Silverman, 1986). The accuracy of this kernel estimator is largely dependent on the choice of the smoothing parameter (or bandwidth) λ, with its optimal value depending on the distribution of the available data sample (May et al., 2008a). A value of λ that is too large will result in an over-smoothed probability density, while a value that is too small can lead to density estimates overly influenced by individual data points (under-smooth). Sharma (2000) adopted the Gaussian reference bandwidth (Scott, 1992) due to its simplicity and computational efficiency. Calculation of the PMI criterion also requires the appropriate estimation of the conditional expectation $E[\cdot]$ of $\mathbf{x}$ and $\mathbf{y}$ on Z. Bowden et al. (2005a) proposed the use of a General Regression Neural Network (GRNN) to compute these conditional expectations. GRNNs are very similar in their underlying philosophy to kernel regression, where a non-parametric estimate of the pdf of the observed data, similar to that given by eq. (4), is utilised in the estimation of $E[\cdot]$, rather than assuming any particular form for the regression function. At each iteration, the PMIS algorithm seeks to find the variable $\mathbf{x}_s$ which maximises the PMI with respect to $\mathbf{y}$, conditional on the inputs that have been selected in previous iterations, Z. If $\mathbf{x}_s$ is found to be relevant (based on some stopping criterion), it is added to the selected subset Z and the selection continues; otherwise, the algorithm is terminated since there are no more relevant candidate inputs remaining. For the purposes of this study, the stopping criterion utilised was the coefficient of determination, R^2, of the output variable residual, $\mathbf{y}'$ (see May et al. (2008a) for an analysis of different stopping criteria).

The advantages of using a GRNN in the PMIS algorithm include: accuracy in modelling the nonlinear relationships between the inputs and output, computationally efficient model calibration, and fixed model structure that does not have to be tuned on each specific dataset (Bowden et al., 2005a). On the other hand, a limitation of the PMIS algorithm is that, although it is a filter method, it can still be relatively computationally expensive due to the use of kernel based approaches for estimating the PMI criterion and the conditional expectations $E[\cdot]$. While such approaches give efficient and reliable density estimates for smaller data sets, their computational efficiency decreases dramatically with increasing sample size (Fernando et al., 2009). Furthermore, the Gaussian reference bandwidth, which is utilised in the calculation of the marginal and joint pdfs of x and y, as well as in the GRNN estimates of $E[x|Z]$ and $E[y|Z]$, can tend to over-smooth and its optimality might be questionable if the data are not Gaussian (May et al., 2008a). For further details of this algorithm, see Sharma (2000); Bowden et al. (2005a) and May et al. (2008a).

A.2 IIS algorithm

The IIS algorithm is a hybrid filter-wrapper IVS method introduced by Galelli and Castelletti (2013b). Similar to the PMIS algorithm, IIS adopts a forward selection approach to iteratively select the most significant inputs, but uses a tree-based ranking method instead of an information-theoretic measure to estimate the relative contribution of each candidate input. At each iteration, all the input variables are ranked according to their relative contribution to the building of an underlying model of the output. The relative significance of the first p ranked variables is then assessed against the output by identifying p Single Input-Single Output (SISO) models. Eventually, the best performing input among the p considered (according to a preselected measure of accuracy) is added to the set of the selected variables. At the first iteration of the IIS algorithm, both ranking and SISO models are run on a data set composed of time series of the candidate input variables and the associated output values. At the subsequent iterations, the original output values are replaced by the residuals of the underlying model built at the previous iteration. The re-evaluation of ranking and SISO models every time an input is selected (i.e., at each iteration) ensures that all the candidate inputs that are highly correlated with the selected input are discarded, thus minimizing the redundancy of the final set of selected inputs.

The IIS algorithm terminates when the accuracy of the model built upon the selected variables, as evaluated with a k-fold cross validation (Allen, 1974), starts decreasing (or when it does not significantly improve). As discussed in Wan Jaafar et al. (2011), this process is aimed at minimizing the risk of overfitting the data, since it estimates the ability of the model to capture the behavior of unseen or future observations.

In the present study the underlying model performance is computed with the coefficient of determination R^2, while both the ranking and model building algorithm are based on Extremely Randomized Trees (Extra-Trees, Geurts et al. (2006); Galelli and Castelletti (2013a)). Similar to the PMIS algorithm, the idea of exploiting the underlying model residuals provides robustness against redundant inputs, while the adoption of Extra-Trees allows accounting for non-linear interactions and computational efficiency (with respect to sample size N and the number P of candidate inputs). Furthermore, the tree-based ranking method does not require any specific assumption regarding the structure of the dependence between input and output variables. However, as any other forward selection method, the IIS algorithm does not account for the inter-dependency between candidate input variables. For further technical details the reader is referred to Galelli (2010) and Galelli and Castelletti (2013b).

A.3 GA-ANN algorithm

The algorithm described herein is one particular implementation of a combination of a Genetic Algorithm (GA) search procedure with an Artificial Neural Network (ANN) model. In particular, a simple 1-hidden node multilayer perceptron was utilised in this algorithm[1]. The model training process is performed by means of a simulated annealing algorithm (Belisle, 1992), which is used each time a new combination of inputs is evaluated.

The GA here adopted is a relatively simple variant which is outlined in Goldberg (1989). In this implementation, solutions representing different subsets of inputs, are encoded as binary strings, called 'chromosomes'. Each bit, or 'gene', in these chromosomes represents a candidate input variable, where a '1' denotes that the input will be included in the model, while a '0' denotes its omission from the model. The objective function used to determine whether one subset of inputs is better (fitter) than another was the out-of-sample AIC, computed using a k-fold cross-validation. This objective function was also used as a stopping criterion to terminate the GA-ANN algorithm.

The main drawback of this implementation of the GA-ANN algorithm is that the complexity of the learning algorithm, and hence its ability to accurately model complex functions, is limited by the choice of an ANN with a single hidden node. However, this model should still provide an improvement over a simple linear mapping when applied to nonlinear datasets.

[1] Ideally, the structure and complexity of the ANN model would be optimised to suit the problem at hand; however, when evaluating a general algorithm across a number of different datasets this can become impractical.

A.4 PCIS algorithm

The partial correlation input selection (PCIS) algorithm (May et al., 2008a) is based on partial correlation analysis, which aims to find the linear correlation between two variables after removing the effects of other variables. The PCIS algorithm is structured the same as the PMIS algorithm, but with the partial linear correlation coefficient used in place of the PMI criterion for measuring the relevance of inputs. This coefficient is calculated as Pearson's correlation between the residuals $\mathbf{x}'$ and $\mathbf{y}'$, given by eq. (5), once the effect of the already selected inputs, Z, has been taken into consideration. In this case, the conditional expectation $E[\cdot]$ is a linear regression of $\mathbf{x}$ and $\mathbf{y}$ with Z. The regression is based on a least-squares approach, which implies a Gaussian distribution of the residuals. The PCIS algorithm is terminated when the selection of additional inputs no longer results in an improvement (increase) in the BIC, calculated based on the output variable residual $\mathbf{y}'$, which provides a trade-off between goodness-of-fit and model complexity.

Appendix B

B.1 PMIS algorithm

The computing time $t_{PMIS,i}$ associated with the i-th iteration of the PMIS algorithm is the combination of the time $t_{PMIS,T1}$ required to calibrate a GRNN to estimate the output based on the selected inputs, the time $t_{PMIS,T2}$ required to calibrate a GRNN to estimate each (non-selected) input based on the selected inputs, and the time $t_{PMIS,T3}$ for computing the PMI between the residual of each model. Knowing that the run-time order to calibrate a GRNN is $O(K^2 \cdot N^2 + K^3)$ (where K and N are the number of inputs and observations, respectively), and that the run-time order to estimate the PMI is $O(N^2)$, the time $t_{PMIS,T1}$, $t_{PMIS,T2}$ and $t_{PMIS,T3}$ can be estimated as follows:

$$t_{PMIS,T1} = c \cdot \left((i-1)^2 \cdot N^2 + (i-1)^3\right) \tag{6a}$$

where c is a constant, machine-dependent parameter and i the iteration number.

$$t_{PMIS,T2} = c \cdot (P-(i-1)) \cdot \left((i-1)^2 \cdot N^2 + (i-1)^3\right) \tag{6b}$$

where P is the number of candidate input variables.

$$t_{PMIS,T3} = c \cdot (P-(i-1)) \cdot N^2 \tag{6c}$$

Therefore, the time $t_{PMIS,i}$ associated with the i-th iteration is equal to

$$t_{PMIS,i} = t_{PMIS,T1} + t_{PMIS,T2} + t_{PMIS,T3} \tag{7a}$$

while the time $t_{PMIS,n}$ required to perform n iterations is

$$\begin{aligned} t_{PMIS,n} = {} & c \cdot \sum_{i=1}^{n} \left((i-1)^2 \cdot N^2 + (i-1)^3\right) + \\ & + c \cdot \left[\sum_{i=1}^{n} (P-(i-1)) \cdot \left((i-1)^2 \cdot N^2 + (i-1)^3\right) + \sum_{i=1}^{n} (P-(i-1)) \cdot N^2\right] = \\ & = c \cdot \sum_{i=1}^{n} \left((i-1)^2 \cdot N^2 + (i-1)^3\right) + \\ & + c \cdot \sum_{i=1}^{n} (P-(i-1)) \cdot \left((i-1)^2 \cdot N^2 + (i-1)^3 + N^2\right) \end{aligned} \tag{7b}$$

In the worst case scenario, the PMIS algorithm is run over P iterations to evaluate all candidate inputs. In this case, the total computing time is

$$\begin{aligned} t_{PMIS}(P) = c \cdot \sum_{i=1}^{P} \left((i-1)^2 \cdot N^2 + (i-1)^3\right) + \\ + c \cdot \sum_{i=1}^{P} (P-(i-1)) \cdot \left((i-1)^2 \cdot N^2 + (i-1)^3 + N^2\right) \end{aligned} \tag{8}$$

so the run-time order is $O(P^4 \cdot N^2 + P^5)$.

B.2 IIS algorithm

The computing time $t_{IIS,i}$ associated with the i-th iteration of the IIS algorithm is the combination of the time $t_{IIS,T1}$ required to run the ranking method, the time $t_{IIS,T2}$ for evaluating the accuracy of p SISO models and the time $t_{IIS,T3}$ for evaluating the underlying MISO model. Knowing that the computing time of Extra-Trees grows superlinearly in the number N of observations, and linearly in the number K and M of inputs and trees, the time $t_{IIS,T1}$, $t_{IIS,T2}$ and $t_{IIS,T3}$ can be estimated as follows:

$$t_{IIS,T1} = c \cdot (N \cdot \log(N)) \cdot M \cdot P \tag{9a}$$

where c is a constant, machine-dependent parameter, and P the number of candidate input variables.

$$\begin{aligned} t_{IIS,T2} &= c \cdot p \cdot k \cdot \left(\left(\frac{N}{k} \cdot (k-1)\right) \cdot \log\left(\frac{N}{k} \cdot (k-1)\right)\right) \cdot M \cdot 1 = \\ &= c \cdot p \cdot T \end{aligned} \tag{9b}$$

where k is the number of folds in the k-fold cross-validation process and T is equal to $k \cdot \left(\left(\frac{N}{k} \cdot (k-1)\right) \cdot \log\left(\frac{N}{k} \cdot (k-1)\right)\right) \cdot M$.

$$\begin{aligned} t_{IIS,T3} &= c \cdot k \cdot \left(\left(\frac{N}{k} \cdot (k-1)\right) \cdot \log\left(\frac{N}{k} \cdot (k-1)\right)\right) \cdot M \cdot i = \\ &= c \cdot T \cdot i \end{aligned} \tag{9c}$$

where i is the iteration number, which can range from 1 to P.

Therefore, the time $t_{IIS,i}$ associated with the i-th iteration is equal to

$$t_{IIS,i} = t_{IIS,T1} + t_{IIS,T2} + t_{IIS,T3} \tag{10a}$$

while the time $t_{IIS,n}$ required to perform n iterations is

$$t_{IIS,n} = n \cdot t_{IIS,T1} + n \cdot t_{IIS,T2} + \sum_{i=1}^{n} c \cdot T \cdot i \tag{10b}$$

In the worst case scenario, the IIS algorithm is run over P iterations to evaluate all candidate inputs. In this case, the total computing time is

$$\begin{aligned} t_{IIS}(P) &= P \cdot t_{IIS,T1} + P \cdot t_{IIS,T2} + c \cdot T \cdot [1 + 2 + \ldots + (P-1) + P] = \\ &= P \cdot t_{IIS,T1} + P \cdot t_{IIS,T2} + c \cdot T \cdot [\frac{1}{2} \cdot (P^2 + P)] \end{aligned} \tag{11}$$

so the run-time order is $O(T \cdot P^2)$, that is $O(k \cdot \left(\left(\frac{N}{k} \cdot (k-1)\right) \cdot \log\left(\frac{N}{k} \cdot (k-1)\right)\right) \cdot M \cdot P^2)$.

B.3 PCIS algorithm

The computing time $t_{PCIS,i}$ associated with the i-th iteration of the PCIS algorithm is the combination of the time $t_{PCIS,T1}$ required to build a linear model to estimate the output based on the selected inputs, the time $t_{PCIS,T2}$ required to build a linear model to estimate each (non-selected) input based on the selected inputs, and the time $t_{PCIS,T3}$ for computing the Pearson correlation between the residual of each model. Knowing that the run-time order to build a linear model is $O(K^2 \cdot N + K^3)$, and that the run-time order to estimate the Pearson correlation is $O(N)$, the time $t_{PCIS,T1}$, $t_{PCIS,T2}$ and $t_{PCIS,T3}$ can be estimated as follows:

$$t_{PCIS,T1} = c \cdot \left((i-1)^2 \cdot N + (i-1)^3\right) \tag{12a}$$

where c is a constant, machine-dependent parameter and i the iteration number.

$$t_{PCIS,T2} = c \cdot (P-(i-1)) \cdot \left((i-1)^2 \cdot N + (i-1)^3\right) \tag{12b}$$

where P is the number of candidate input variables.

$$t_{PCIS,T3} = c \cdot (P-(i-1)) \cdot N \tag{12c}$$

Therefore, the time $t_{PCIS,i}$ associated with the i-th iteration is equal to

$$t_{PCIS,i} = t_{PCIS,T1} + t_{PCIS,T2} + t_{PCIS,T3} \tag{13a}$$

while the time $t_{PCIS,n}$ required to perform n iterations is

$$\begin{aligned} t_{PCIS,n} &= c \cdot \sum_{i=1}^{n} \left((i-1)^2 \cdot N + (i-1)^3\right) + \\ &+ c \cdot \left[\sum_{i=1}^{n} (P-(i-1)) \cdot \left((i-1)^2 \cdot N + (i-1)^3\right) + \sum_{i=1}^{n} (P-(i-1)) \cdot N\right] = \\ &= c \cdot \sum_{i=1}^{n} \left((i-1)^2 \cdot N + (i-1)^3\right) + \\ &+ c \cdot \sum_{i=1}^{n} (P-(i-1)) \cdot \left((i-1)^2 \cdot N + (i-1)^3 + N\right) \end{aligned} \tag{13b}$$

In the worst case scenario, the PCIS algorithm is run over P iterations to evaluate all candidate inputs. In this case, the total computing time is

$$\begin{aligned} t_{PCIS}(P) = c \cdot \sum_{i=1}^{P} \left((i-1)^2 \cdot N + (i-1)^3\right) + \\ + c \cdot \sum_{i=1}^{P} (P - (i-1)) \cdot \left((i-1)^2 \cdot N + (i-1)^3 + N\right) \end{aligned} \tag{14}$$

so the run-time order is $O(P^4 \cdot N + P^5)$.

Table 1: Benchmark dataset properties

	Dataset	N	K	P	N/P	Fully/Partially Synthetic	Non-Gaussian Output	Highly Nonlinear	High Noise	High Collinearity	Inter-dependency	Incomplete Information
1	AR1	500	1	15	33.3	Fully			X	X		
2	AR9_500	500	3	15	33.3	Fully			X	X		
3	AR9_70	70	3	15	4.7	Fully			X	X		
4	TAR1	500	1	15	33.3	Fully			X	X		
5	TAR2	500	2	15	33.3	Fully			X	X		
6	NL_500	500	3	15	33.3	Fully	X	X				
7	NL_70	70	3	15	4.7	Fully	X	X				
8	NL2	500	3	15	33.3	Fully	X	X	X	X		
9	Bank_fm	400	8	32	12.5	Fully	X					X
10	Bank_fh	400	8	32	12.5	Fully	X		X			X
11	Bank_nm	400	8	32	12.5	Fully	X	X				X
12	Bank_nh	400	8	32	12.5	Fully	X	X	X			X
13	Friedman_c0_10_m	250	5	10	25	Fully		X				
14	Friedman_c0_10_h	250	5	10	25	Fully		X	X			
15	Friedman_c0_50_m	250	5	50	5	Fully		X				
16	Friedman_c0_50_h	250	5	50	5	Fully		X	X			
17	Friedman_c25_10_m	250	5	10	25	Fully		X		X		
18	Friedman_c25_10_h	250	5	10	25	Fully		X	X	X		
19	Salinity_5_l	4120	3	80	51.5	Partially				X		
20	Salinity_5_m	4120	3	80	51.5	Partially				X		
21	Salinity_5_h	4120	3	80	51.5	Partially			X	X		
22	Salinity_10_l	4115	3	160	25.7	Partially				X		
23	Salinity_10_m	4115	3	160	25.7	Partially				X		
24	Salinity_10_h	4115	3	160	25.7	Partially			X	X		
25	Kentucky	4739	4	21	225.7	Partially	X			X		
26	Miller	200	2	3	66.7	Fully	X				X	

Table 2: Average run-time [sec] for the PMIS, IIS, GA-ANN and PCIS algorithms over the 26 benchmark datasets.

	Dataset	N	K	P	PMIS	IIS	GA-ANN	PCIS
1	AR1	500	1	15	16.80 ± 2.87	9.49 ± 2.70	1491.22 ± 560.28	0.16 ± 0.05
2	AR9_500	500	3	15	38.84 ± 3.29	26.73 ± 6.64	1973.36 ± 864.35	0.38 ± 0.13
3	AR9_70	70	3	15	2.22 ± 0.43	3.39 ± 0.91	378.49 ± 199.71	0.38 ± 0.13
4	TAR1	500	1	15	14.02 ± 1.22	11.66 ± 3.92	841.69 ± 330.93	0.23 ± 0.09
5	TAR2	500	2	15	26.13 ± 3.46	8.08 ± 0.16	1630.37 ± 860.63	0.40 ± 0.27
6	NL_500	500	3	15	23.41 ± 3.72	24.51 ± 3.39	878.56 ± 272.30	0.20 ± 0.08
7	NL_70	70	3	15	1.82 ± 0.66	4.93 ± 1.79	185.83 ± 117.37	0.25 ± 0.25
8	NL2	500	3	15	24.26 ± 4.94	15.66 ± 5.02	850.15 ± 468.98	0.33 ± 0.15
9	Bank_fm	400	8	32	31.03 ± 6.18	44.41 ± 6.22	1544.37 ± 341.93	1.23 ± 0.36
10	Bank_fh	400	8	32	35.07 ± 7.45	25.52 ± 4.08	1754.10 ± 775.55	0.99 ± 0.27
11	Bank_nm	400	8	32	48.33 ± 17.61	41.94 ± 2.12	1732.64 ± 288.63	1.55 ± 0.50
12	Bank_nh	400	8	32	34.50 ± 12.82	30.59 ± 3.65	1667.77 ± 634.60	1.29 ± 0.37
13	Friedman_c0_10_m	250	5	10	13.36 ± 1.10	9.64 ± 0.45	609.90 ± 215.67	0.31 ± 0.07
14	Friedman_c0_10_h	250	5	10	10.79 ± 1.41	6.68 ± 0.44	710.52 ± 331.96	0.56 ± 0.58
15	Friedman_c0_50_m	250	5	50	46.61 ± 3.42	60.59 ± 4.85	2074.70 ± 564.22	1.26 ± 0.35
16	Friedman_c0_50_h	250	5	50	38.72 ± 6.58	57.39 ± 6.45	1832.76 ± 593.04	1.33 ± 0.55
17	Friedman_c25_10_m	250	5	10	10.31 ± 2.62	10.40 ± 2.47	325.50 ± 87.30	0.26 ± 0.18
18	Friedman_c25_10_h	250	5	10	7.78 ± 1.87	3.17 ± 0.56	303.35 ± 146.60	0.22 ± 0.05
19	Salinity_5_l	4120	3	80	5,017.90 ± 477.32	1,872.05 ± 24.43	100,393.26 ± 24,601.97	73.99 ± 26.89
20	Salinity_5_m	4120	3	80	3,672.10 ± 521.62	1,595.38 ± 66.56	83,150.26 ± 19,792.51	53.32 ± 21.84
21	Salinity_5_h	4120	3	80	3,574.37 ± 394.79	1,451.59 ± 130.43	52,066.80 ± 10,753.36	11.16 ± 4.72
22	Salinity_10_l	4115	3	160	9,024.13 ± 515.44	5,427.50 ± 81.49	287,687.47 ± 58,411.79	143.65 ± 45.33
23	Salinity_10_m	4115	3	160	7,995.80 ± 1,541.41	5,457.86 ± 149.86	226,791.40 ± 47,583.28	143.83 ± 52.24
24	Salinity_10_h	4115	3	160	7,877.20 ± 2,832.16	6,005.77 ± 840.08	147,507.80 ± 25,653.14	18.09 ± 9.16
25	Kentucky	4739	4	21	1,860.37 ± 107.22	800.58 ± 92.86	106,725.55 ± 27,214.61	7.65 ± 1.88
26	Miller	200	2	3	2.65 ± 0.27	0.98 ± 0.55	5664.92 ± 1561.83	0.14 ± 0.08

Table 3: Run-time order of PMIS, IIS, GA-ANN and PCIS algorithms. P and N represent the number of candidate inputs and observations, respectively, while T is equal to $k \cdot \left(\left(\frac{N}{k} \cdot (k-1)\right) \cdot \log\left(\frac{N}{k} \cdot (k-1)\right)\right) \cdot M$ (where k is the number of folds in the k-fold cross-validation process and M is the number of trees in an ensemble). See Appendix B for further details.

IVS algorithm	PMIS	IIS	GA-ANN	PCIS
Run-time order	$O(P^4 \cdot N^2 + P^5)$	$O(T \cdot P^2)$	-	$O(P^4 \cdot N + P^5)$

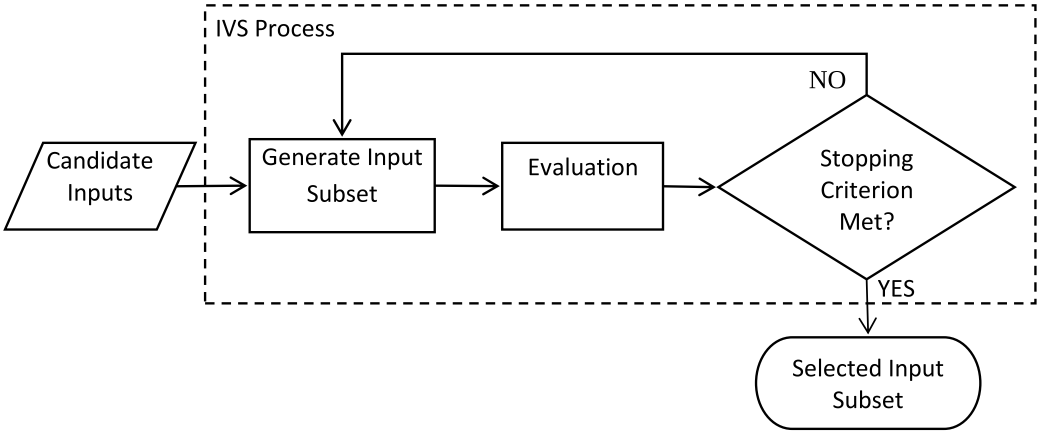


Figure 1: The generic IVS process (adapted from Dash and Liu (1997)).

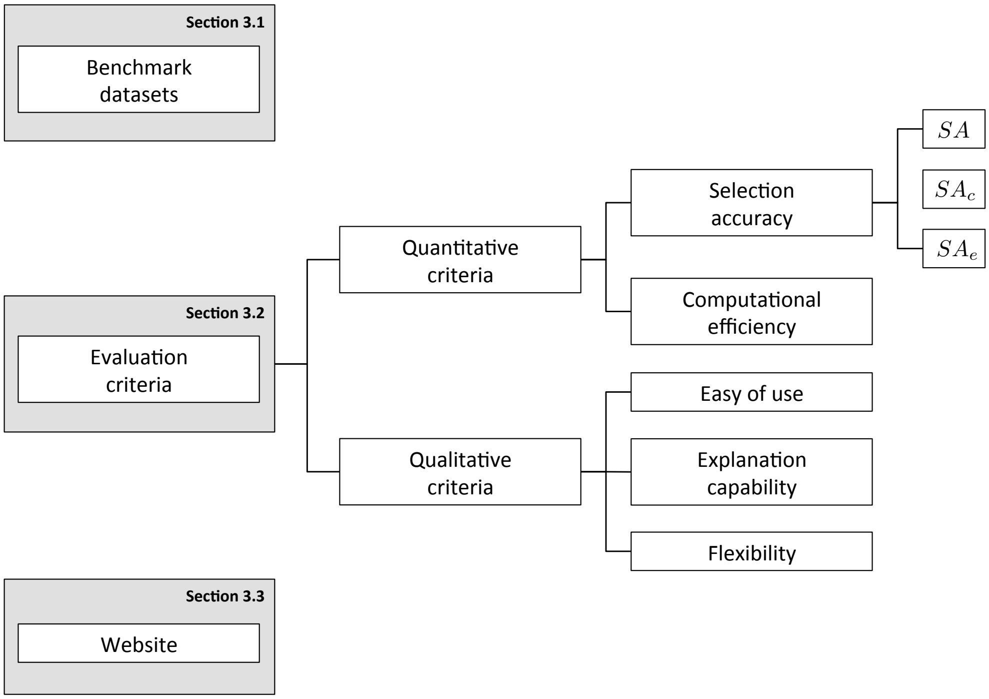


Figure 2: Schematic representation of the IVS framework components.

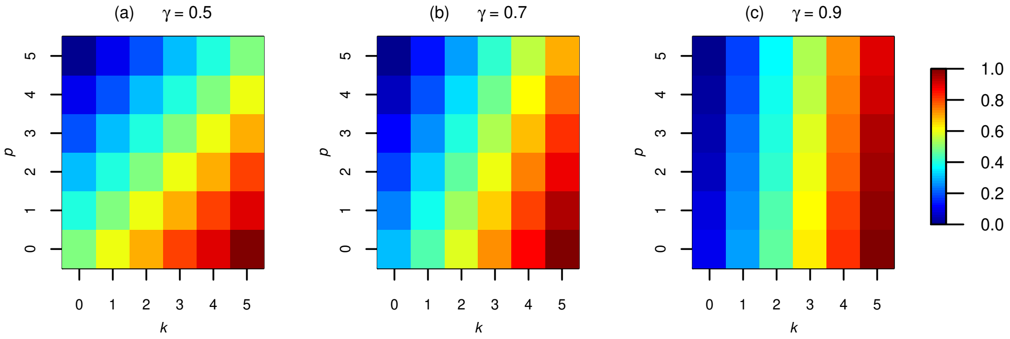


Figure 3: The effect of γ on SA score.

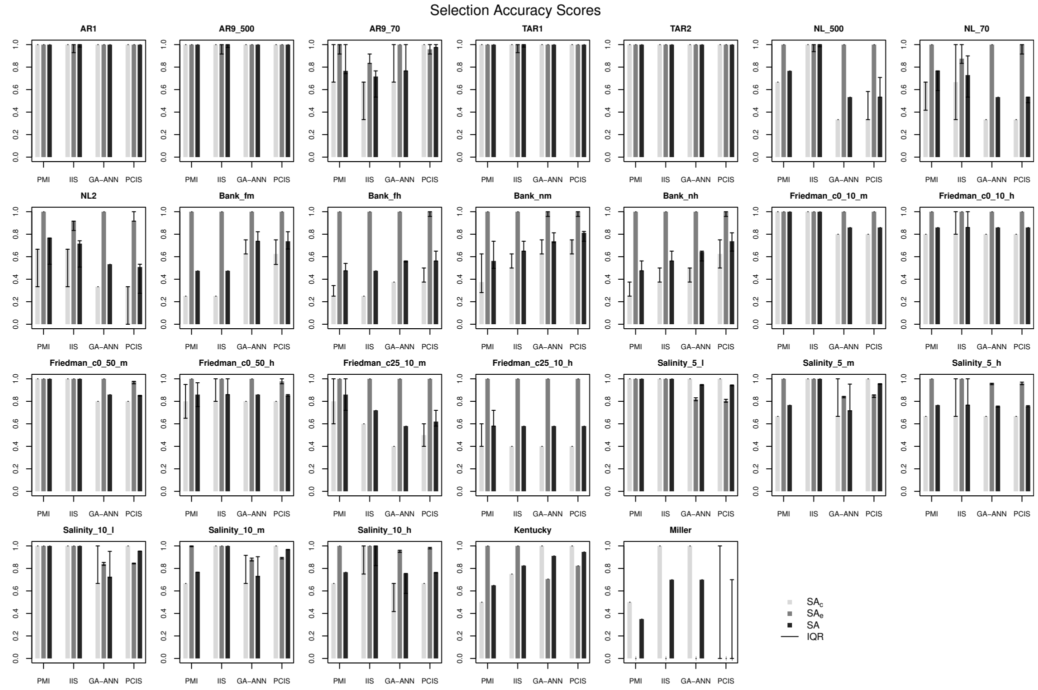


Figure 4: Bar charts representing the values of the scores SA, SA_c and SA_e obtained by running the PMIS, IIS, GA-ANN and PCIS algorithms on the 26 benchmark datasets.

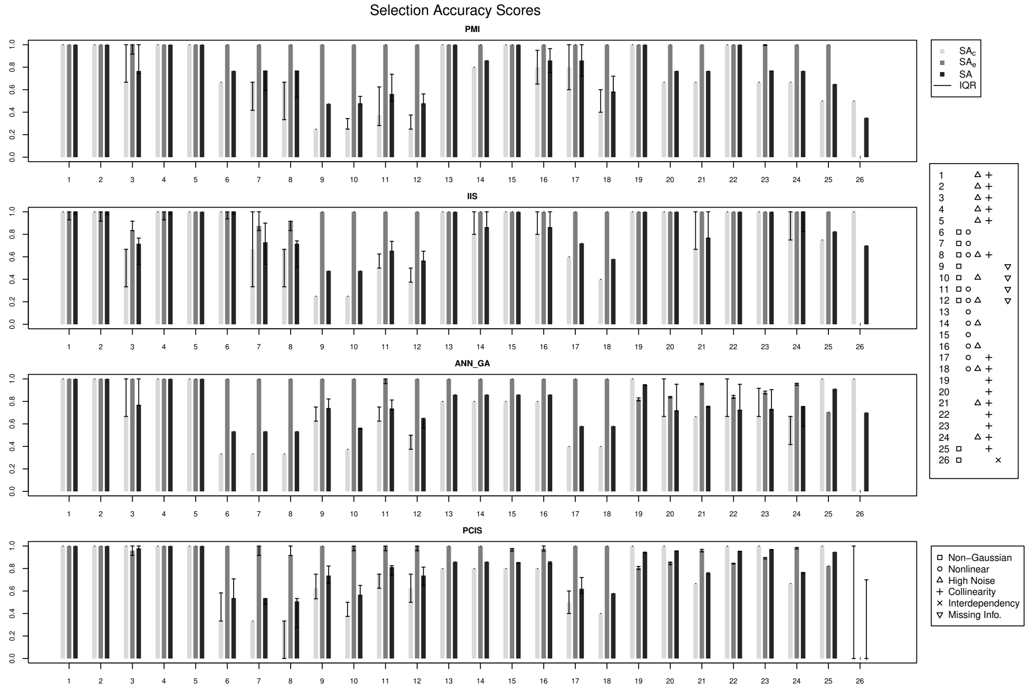


Figure 5: Bar charts representing the values of the scores SA, SA_c and SA_e obtained by running the PMIS, IIS, GA-ANN and PCIS algorithms on the 26 benchmark datasets. The datasets properties are described on the right-hand side.

Figure 1

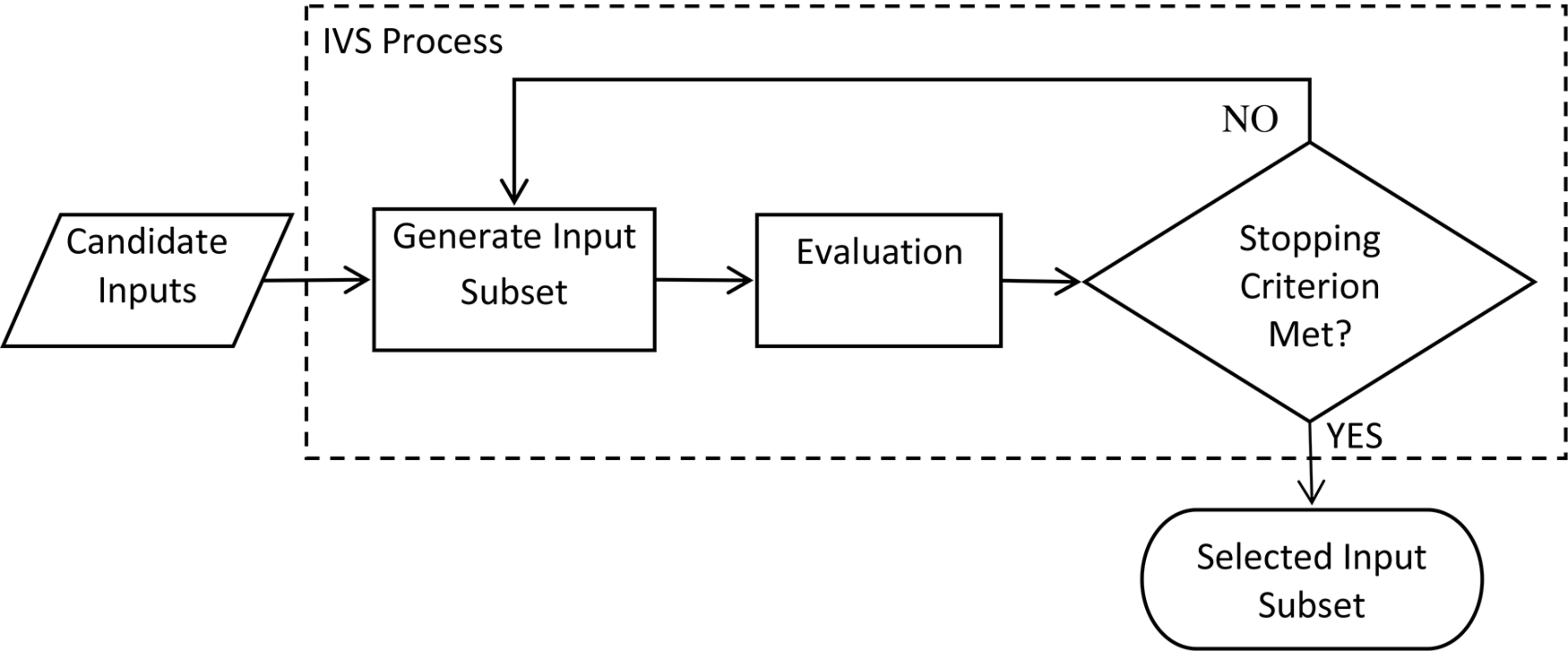

IVS Process
Candidate Inputs
Generate Input Subset
Evaluation
Stopping Criterion Met?
NO
YES
Selected Input Subset

Figure 2

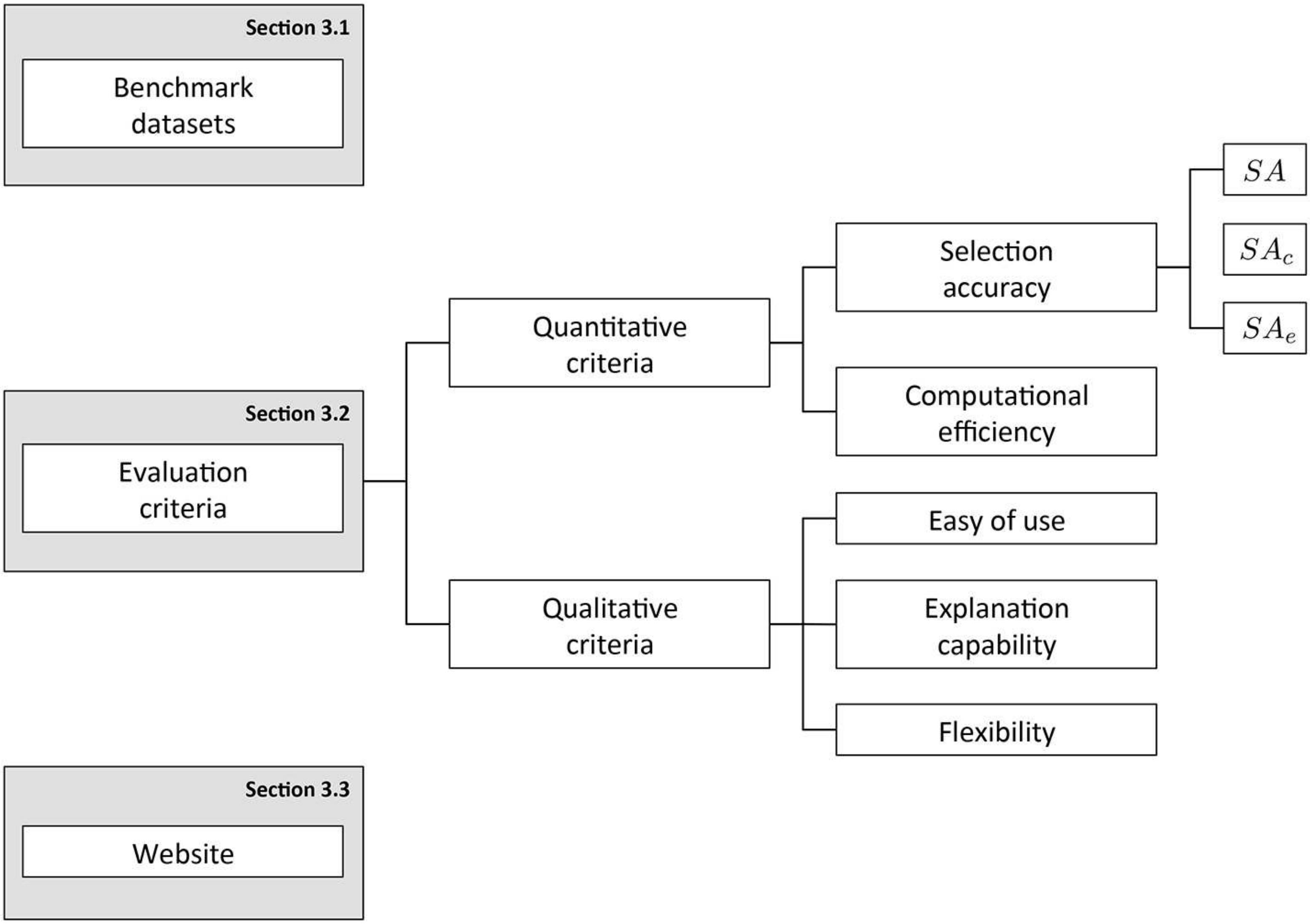

Section 3.1
Benchmark datasets
Section 3.2
Evaluation criteria
Quantitative criteria
Selection accuracy
SA
SA_c
SA_e
Computational efficiency
Qualitative criteria
Easy of use
Explanation capability
Flexibility
Section 3.3
Website

Figure 3

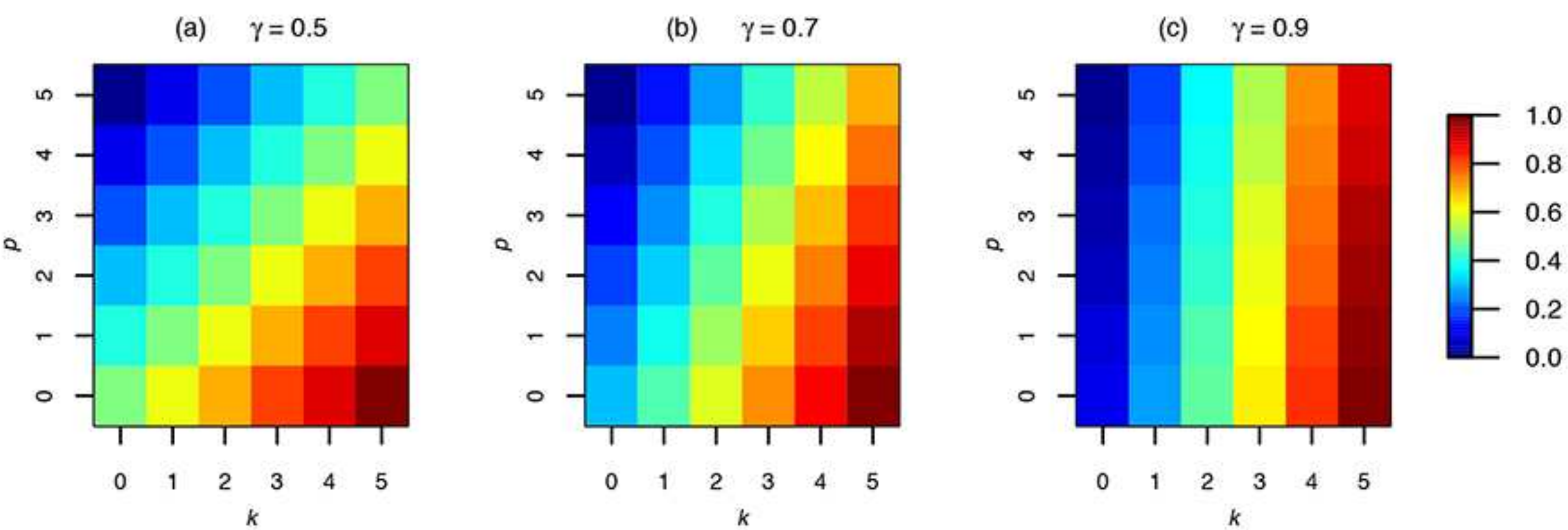


Figure 4

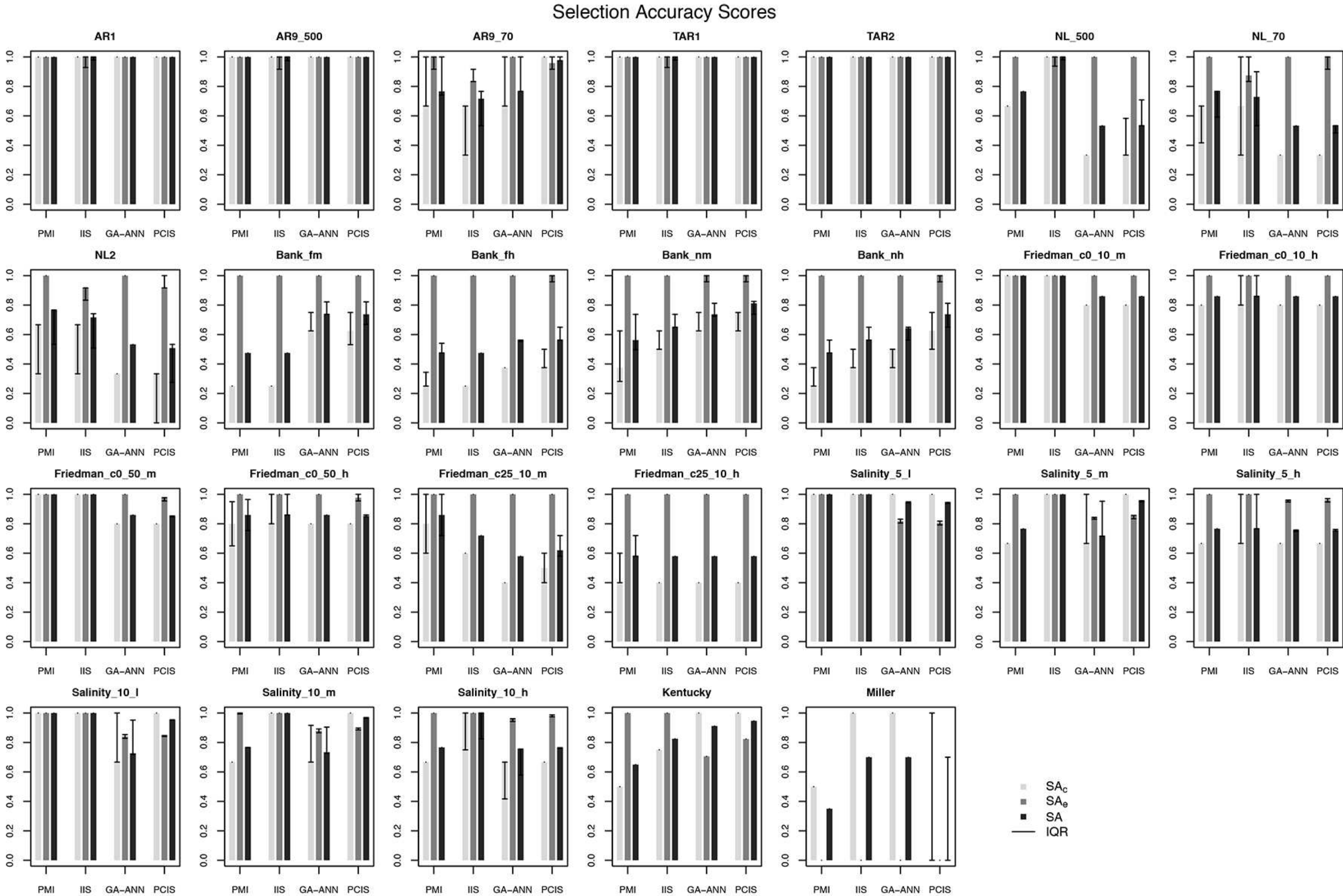
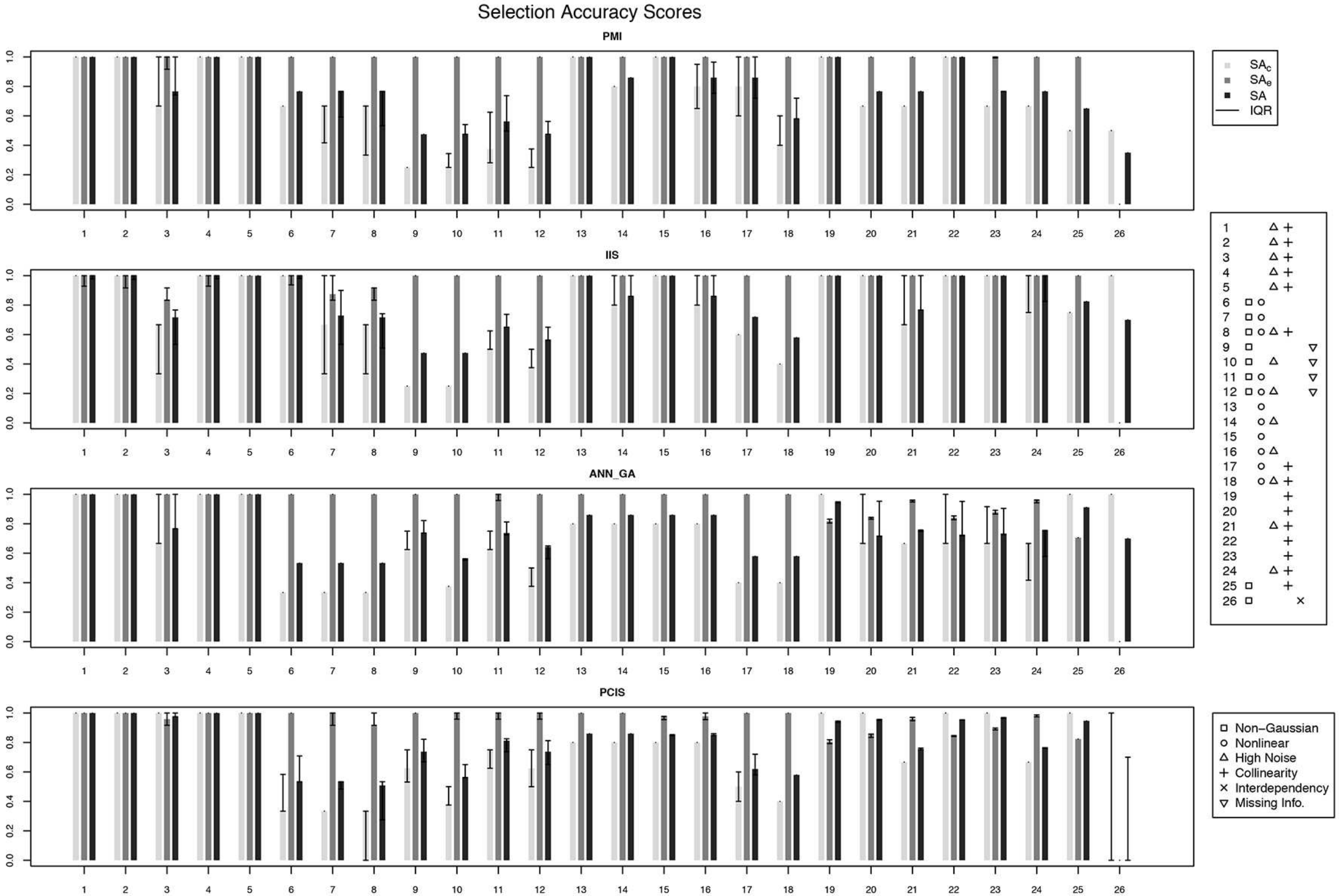

Selection Accuracy Scores
AR1
AR9_500
AR9_70
TAR1
TAR2
NL_500
NL_70
NL2
Bank_fm
Bank_fh
Bank_nm
Bank_nh
Friedman_c0_10_m
Friedman_c0_10_h
Friedman_c0_50_m
Friedman_c0_50_h
Friedman_c25_10_m
Friedman_c25_10_h
Salinity_5_l
Salinity_5_m
Salinity_5_h
Salinity_10_l
Salinity_10_m
Salinity_10_h
Kentucky
Miller
0.0
0.2
0.4
0.6
0.8
1.0
PMI
IIS
GA–ANN
PCIS
SA_c
SA_e
SA
IQR

Figure 5


Selection Accuracy Scores
PMI
IIS
ANN_GA
PCIS
0.0
0.2
0.4
0.6
0.8
1.0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
SA_c
SA_e
SA
IQR
1 △ +
2 △ +
3 △ +
4 △ +
5 △ +
6 □ ○
7 □ ○
8 □ ○ △ +
9 □ ▽
10 □ △ ▽
11 □ ○ ▽
12 □ ○ △ ▽
13 ○
14 ○ △
15 ○
16 ○ △
17 ○ +
18 ○ △ +
19 +
20 +
21 △ +
22 +
23 +
24 △ +
25 □ +
26 □ ×
□ Non−Gaussian
○ Nonlinear
△ High Noise
+ Collinearity
× Interdependency
▽ Missing Info.