Application of all relevant feature selection for the failure analysis of parameter-induced simulation crashes in climate models

W. Paja[1], M. Wrzesien[2] , R.Niemiec[2] and W. R. Rudnicki[3,4]

[1]{Department of Computer Science, Faculty of Mathematics and Natural Sciences, University of Rzeszow, Rzeszow, Poland}

[2]{Department of Artificial Intelligence and Expert Systems, Faculty of Applied Informatics, University of Information Technology and Management, Rzeszów, Poland}

[3]{Interdisciplinary Centre for Mathematical and Computational Modelling, University of Warsaw, Warsaw, Poland}

[3]{Department of Bioinformatics, University of Białystok, Białystok, Poland}

Correspondence to: W. R. Rudnicki (w.rudnicki@icm.edu.pl)

Abstract

The climate models are extremely complex pieces of software. They reflect the best knowledge on the physical components of the climate, nevertheless, they contain several parameters, which are too weakly constrained by observations, and can potentially lead to a crash of simulation. Recently a study by Lucas et al. has shown that machine learning methods can be used for predicting which combinations of parameters can lead to the crash of simulation, and hence which processes described by these parameters need refined analyses. In the current study we re-analyse the dataset used in this research using different methodology. We confirm the main conclusion of the original study concerning the suitability of machine learning for the prediction of crashes. We show that only three of the eight parameters indicated in the original study as relevant for prediction of the crash are indeed strongly relevant, three other are relevant but redundant, and two are not relevant at all. We also show that the variance due to the split of data between training and validation sets has

a large influence both on the accuracy of predictions and relative importance of variables, hence only cross-validated approach can deliver a robust prediction of performance and relevance of variables.

1 Introduction

The development of realistic models of climate is one of the most important areas of research due to the dangers posed by global warming. It is by no means a trivial task since it involves the parameterisation of many processes that are not directly solved within the model. It has been shown by (Lucas et al., 2013) that certain combinations of these parameters, lead to failure of a model, despite each individual parameter having a reasonable value. Authors of this study performed 540 simulations with randomly varied combinations of 18 parameters of the Parallel Ocean Program (POP2) (Smith et al., 2010) module in the Community Climate System Model Version 4 (CCSM4) (UCAR, 2010). About 10 percent of these simulations crashed due to numerical instabilities. Then they have applied machine learning methods to attribute failures to the parameters of the model. To this end they had used the support vector machine (SVM) (Vapnik, 1995) classification to quantify and predict the probability of failure as a function of the values of 18 from POP2 parameters. The causes of the simulation failures were determined through a global sensitivity analysis. Combinations of 8 parameters related to ocean mixing and viscosity from three different POP2 parameterizations were then determined as the major sources of the failures. These 8 parameters were indicated as targets for more detailed research.

These results are somewhat disappointing, since the number of parameters is still rather high. Hence we decided to check whether more elaborate method for analysis could decrease this number further. We have observed potential weak points of the analysis performed by Lucas and co-workers, namely, they have not fully taken into account that the apparent importance of a variable for classification may be in fact the result of a spurious fluctuation. The problem is most acute when a sample used for machine learning algorithm is small. In such a case random fluctuation may introduce spurious correlations within data, which can be utilized by the classification algorithm for model building. The appropriate procedure should be applied to minimize the influence of such random correlations on the final results.

Lucas and co-workers have also analyzed the impact of the decision variable that is used for the classification on the quality of results. While the models were built as an ensemble of learners built on the bootstrap samples of the training set, the evaluation of the classification performance was based on a single split of data between training set and test set. This setup was due to the construction of the study – simulations for the validation set were performed after the predictions have been made. While this is a very honest method for the verification of the predictions, however, it precludes the estimation of the statistical uncertainty of the result. In particular, it is impossible to say whether the observed differences between classification accuracy observed for different decision functions are significant or do they arise due to statistical fluctuations.

The current study is devoted to the reanalysis of the data. It aims at minimizing the influence of random fluctuations on the final results. Our aim was to establish all variables that truly contribute to the final result of the simulations, i.e. whether the simulation was finished successfully or it crashed. To this end we use contrast variables that carry no information on the decision variable, apply Boruta algorithm for all-relevant feature selection and extensive Monte Carlo sampling. We also compare the quality of classification for several subsets of variables used for prediction of simulation result, to perform a parallel check of relevance of variables.

2 Methods

Similarly to the original work, we rely on machine learning algorithms to identify parameters that critically influence the fate of the simulation. The fundamental idea is that when the classification algorithm can predict result of the simulation, i.e. the successful completion of simulation or the crash, using only the information on the values of certain combinations of selected parameters, then these parameters are indeed responsible for the result. In the original paper the authors performed true prediction and achieved a high degree of accuracy, therefore showing the true predictive power of this approach. On the other hand, this setup precludes estimation of statistical uncertainty for some of their findings. In particular, the discussion of the prediction accuracy in sections 4.4 and 4.5 is based on a single split of data between training and test sets and ignores possibility that effects may depend on the particular split.

In the current study we know all results beforehand, thus we are limited to virtual predictions only. In this approach we split the entire dataset into training and validation sets. We then build a model using the training set and check its quality by performing virtual prediction on the validation set and comparing the predicted results with the true ones. One can take advantage of virtualisation to obtain information about the probability distribution of results.

To this end one can perform multiple virtual experiments, with different splits between training and validation sets, and perform classification experiment on each of these splits. The results of individual trials will differ in most cases, allowing one to draw conclusions not only about mean values but also about variance and even shape of probability distribution. Lucas et al. have used this approach for the sensitivity analysis, utilising ensembles of SVM (Vapnik, 1995) learners for classification. Each member of the ensemble was obtained using different subsample of the training set. The classifier was then used for prediction of the simulation result for the validation set.

We have used a different classification algorithm, namely the Random Forest (Breiman, 2001) and instead of the sensitivity analysis we have applied the all-relevant feature selection algorithm Boruta (Kursa et al., 2010). All computations were performed in R environment for statistical modelling (R Development Core Team, 2008), using the *randomForest* package for classification (Liaw and Wiener, 2002) and the *Boruta* package for feature selection. (Kursa and Rudnicki, 2010). Interestingly, some of the authors of Lucas et al. have recently used Random Forest in their analysis of the results of the CAM5 model applied for study of Madden Julian Oscillation. It was applied to analyse the influence of the model parameters on selected diagnostic variables.

Random Forest is an ensemble algorithm based on decision trees. To ensure the low correlation between elementary learners, each tree is grown using a different random subsample of the original data set. Moreover, each split in the tree is built using only a random subset of the predictor variables. The number of variables in this subset influences the balance between bias and variance for the training set. The default value for classification tasks is a square root of the total number of variables and it is usually a very robust selection. Random Forest is a robust "of the shelf" algorithm that is easily applicable to various classification and regression tasks. It has only few control parameters and usually it does not need fine tuning for the particular problem under scrutiny. In many cases it has a performance comparable or even better than state of the art classifiers and it rarely fails. A big advantage of the algorithm is that it estimates both the estimate of the classification error and of the importance of variables by internal cross-validation. To estimate the latter it measures how much the accuracy of base learners is decreased when information about variable in question is removed from the system.

The Boruta algorithm for all-relevant feature selection uses the Random Forest importance measure to infer their relevance. To this end it extends the information system by variables that are non-informative by design – the so-called contrast variables. It then compares the apparent importance of the original variables with that of the non-informative ones. It performs this multiple times using different realizations of the non-informative variables and performs a statistical test. The algorithm finds both strongly and weakly relevant variables. The notions of strong and weak relevance were introduced by (Kohavi and John, 1997) in the context of the ideal classification algorithm. The features are *strongly relevant* when removing them from the description always results in decreased classification accuracy. Features are *weakly relevant*, when their removal in some cases may decrease classification accuracy. For a more detailed discussion of relevance and the Boruta algorithm see (Kohavi and John, 1997; Rudnicki et al., 2015). Algorithm has been used in different fields, including bioinformatics, remote sensing, bacteriology and medicine (Aagaard et al., 2012; Ackerman et al., 2013; Buday et al., 2013; Duro et al., 2012; Herrera and Bazaga, 2013; Leutner et al., 2012; Ma et al., 2014; Menikarachchi et al., 2012; Saulnier et al., 2011; Strempel et al., 2013).

The climate simulations dataset is highly biased towards successful completion of simulation. Only 46 cases out of 540 are failures. Such unbalanced datasets are often difficult for classification, because the automatic selection of the majority class results in good, but useless, classification accuracy. In such a case no information is gained and hence one cannot perform feature selection. In the first test of the current study this problem was avoided by application of the following protocol, see Fig 1. Firstly eleven balanced subsamples of training set were constructed, each subsample consisted of all objects from minority class (failed simulations) and 1/11th of majority class (successful simulations). In order to check specificity of the feature selection each dataset was extended by contrast variables. To this end each original variable was duplicated and its values were randomly permuted between all objects. In this way a set of *shadow variables* that were non-informative by design was added to the original variables. Then the feature selection procedure was performed on each subsample with the help of the all-relevant feature selection algorithm, implemented in *Boruta* function of the Boruta package. The procedure was repeated 60 times. Altogether all relevant feature selection was performed 660 times. The number of times when the artificially constructed shadow variables were selected as important gives an estimate of the expected level of false discovery. The variables that were selected as important significantly more often than random were examined further, using different test.

The second test probing the importance of variables was performed by analysing the influence of variables used for model building on the prediction quality. The first experiment revealed four variables that were classified as important by Boruta in all, or nearly all, of 660 trials. These variables were considered to form a core variable set, and the model built using these variables was used as a reference. We examined whether removing one of the core variables and whether adding another variable respectively decreases or increases the classification quality measured by AUC. The extension of the core test was examined for three variables that were classified in the first test as important significantly more often than the randomised variables.

The test was performed similarly to the one reported in the original study, see Figure 2. The data set was randomly split into a training set containing 360 objects and a validation set containing 180 objects. The split was performed separately for the minority and majority class, so the number of minority class objects in each training set was 32 and in the validation set it was 14. The *randomForest* function from the identically named R package was used to perform classification and error estimate. The procedure was repeated 30 times and results of 30 repetitions were analysed.

The number of trees in the forest (parameter ntree both in *randomForest* and in *Boruta* functions) was set to 5000 both for feature selection with Boruta and classification with randomForest. In both cases the number of variables examined for each split was equal to the square root of the total number of variables. In our experience these settings are fairly robust, we have examined them internally over multiple datasets (Rudnicki et al., 2015). Moreover, we have checked whether they influence results in the initial trials. The number of trees used was 10 times higher than default, to assure that importance estimate in Random Forest converge to their asymptotic values, the number of trees for classification was the same for consistency.

3 Results and Discussion

The summary of the results of the study is presented in the Table 1. The V1 and V2 variables were deemed important in all 660 cases. Variables V13 and V14 were deemed important in nearly all cases — 593 and 623 cases, respectively. All these variables were also indicated as most important by Lucas et al. However, the results do not agree so well for other variables. Lucas et al. indicated variables V4, V5, V16 and V17 as important but their influence on the final result was much weaker than that of the first group. In the current study the variables V4 and V16 were deemed important by Boruta for 44 and 66 subsamples, respectively. In both cases the number is significantly higher than the average for the random variables, which was obtained as 25±9. On the other hand variables V5 and V17 were deemed important for 19 and 17 subsamples respectively, and these numbers are lower than the average for random variables. Moreover, variable V9, which was not indicated as important by Lucas et al., was deemed important for 62 subsamples.

Hence the first experiment confirmed the importance of variables V1 and V2, has shown that importance of V13 and V14 is nearly universal, it also confirmed the weak importance of variables V4 and V16. On the other hand the importance of variables V5 and V17 was not confirmed with our method, instead variable V9 was found to be weakly important. The example result of the Boruta run for an interesting sample is presented in Figure 3. In this sample the importance was confirmed for variables V9 and V16, whereas variable V13 was deemed irrelevant. The importance of V4 was higher than that of highest random variable, but only barely so, and hence the final decision of Boruta was "tentative". One should note, that the importance returned by Boruta is the averaged importance obtained from the underlying Random Forest algorithm. It is not directly interpretable in terms of the fraction of variance explained by given variable.

One should note, that Boruta is an all-relevant feature selection algorithm that aims at finding both strongly and weakly relevant variables, as defined by Kohavi and John. The second test aimed at discerning between strongly and weakly relevant variables. In the case of V1, V2 the removal of the variable from the core dataset resulted in a dramatic drop of AUC, confirming that these variables are truly informative, see Table 1 and Figure 2. In the case of V14 the difference in AUC – referenced further as $\Delta(\mathrm{AUC})$ – was smaller, but still statistically significant, whereas for the V13 the $\Delta(\mathrm{AUC})$ was much smaller than the standard deviation. Similarly, adding either of the three remaining variables, namely V4, V9 and V16, to the core set, lead to an increase of the AUC by insignificant amount, see Table 1 and Figure 4. Another auxiliary metric that can be used to evaluate the relevance of variables is the number of samples in which the AUC for the model containing the variable is higher than that for the model built without that variable. The results of this metric are consistent with results for the Δ(AUC) – it is 30 for both V1 and V2 and 26 for V14 and these are the only results that are significantly different from random ones. Therefore one can conclude, that only three variables, namely V1, V2 and V14 are *strongly relevant*, whereas the remaining variables are *weakly relevant*.

One should note that the results of the second test were highly variable and largely dependent on the split of data between test and validation sets. It is illustrated in Figure 5 and examples of the results from several samples are given in Table 2. The highest AUC obtained in the experiment was 0.990 for model built using core variables and V16 in sample #12. In the same sample the AUC for model built from core-V2 was 0.888. On the other hand for sample #1 the highest AUC was obtained for the model built on core+V9 and it was 0.879. Also the relative importance of variables depends strongly on the test sample. For example adding variable V4 to the core set can improve AUC by as much as 0.032 (sample #22) or decrease it by 0.006 (sample #6). Similarly for V16 AUC can decrease by 0.016 (sample #6) or increase by 0.016 (sample #22). Most interestingly removing variable V13, which was deemed relevant by Boruta in nearly 90% of samples, can either decrease the AUC by 0.011 (sample #6) or increase it by 0.030 (sample #22). This results show that one cannot rely on a single split between the training set and test set for the estimate of influence of parameters, and that only the average over sufficiently large number of alternative splits can give robust estimates.

The average of the cross-validated AUC obtained for three strongly important variables, namely V1, V2 and V13, was 0.924. The highest average AUC was obtained for model built using five variables, namely {V1, V2, V9, V13, V14}, nevertheless the value AUC=0.931 was not significantly higher than the value obtained for simpler model built using only three variables. The small differences in AUC arise due to small improvements for assigning the probability of failure of the simulation. Such improvement results in small shift in the ranking from least probable to most probable to fail, without actually improving the error rate at the cost of including two more variables in the model.

A single run of the Boruta algorithm in the first test took 2 minutes on a server equipped with Intel Xeon E5620@2.4GHz CPU. The entire protocol took less than 24 hours of single CPU core. The second test is far less computationally demanding. A single run of the randomForest function takes less than 20 seconds on the same CPU, therefore, computations for the entire protocol take less than 10 minutes. This effort is negligible in comparison with the time required to run 540 simulations of the climate model itself.

The results of the study are mostly in good agreement with the results of Lucas et al., however, importance of the variables is not identical. The most important difference is the importance of the variable V13 in both studies. This variable is more important than V14 in the SVM-based model by Lucas et al., whereas our analysis deems it relevant but redundant. However, one should note that in the first test V13 was deemed relevant in nearly 90% of cases, only slightly less than in the case of V14. Only the second test revealed that V13 contains mostly redundant information and on average it does not improve quality of Random Forest predictions. The difference is most likely due to the underlying classifier used in each approach. The SVM is essentially a linear classifier, which can be applied to nonlinear problems using some nonlinear, continuous kernel transformation. On the other hand the Random Forest is based on nonlinear and discrete decision trees. Figure 2 in the Lucas et al. suggests that the decision space of the system under scrutiny is non-continuous. The Random Forest can treat such systems more efficiently using less variables, whereas SVM needs higher dimensional spaces to build hyper-plane separating two classes. We have observed such effects in other systems, for example in our earlier study of the recognition of musical instruments, (Kursa et al. 2009). The other differences are less important, since they involve variables with marginal relevance.

Conclusions

Our reanalysis of the results of 540 simulations is in general qualitative agreement with the results of Lucas et al. The results of the simulation can be predicted with fairly good accuracy using the machine learning approach, and the two different methods give very close results. The cross-validated AUC reported by Lucas et al. by ensemble of SVM classifiers was 0.93. In the current study the average of the cross-validated AUC obtained for three strongly important variables, was 0.924.

We have shown by cross-validation that the AUC reported for the prediction experiment performed by Lucas et al. falls within the range of values that can be expected in such a prediction, however, one should not assign any weight to the particular value obtained. If the split between the training set and test set was set differently the resulting AUC for prediction could be any number between 0.88 and 0.99.

The three most important conclusions for the climate modelling community are following. Firstly, the efforts on improving the numerical stability of simulations should be concentrated on 3 parameters of the CCSM4 parallel ocean model, namely *vconst_corr*, *vconst_2*, and *bckgrnd_vdc1*, that were earlier reported as most important by Lucas et al. The remaining parameters indicated as important in that study are either redundant or not relevant. Secondly – the machine learning methods in general, and all-relevant feature selection in particular are useful tools for analysis of influence of simulation parameters on the final outcome. Finally, application of machine learning should involve cross-validation, and all important modelling steps should be included in the cross-validation loop.

Author contributions. W. Paja performed most computations and drafted the first version of the manuscript, M. Wrzesien and R. Niemiec performed computations and contributed to the writing. W. R. Rudnicki designed the experiments, and wrote the manuscript.

References

Aagaard, K., Riehle, K., Ma, J., Segata, N., Mistretta, T.-A., Coarfa, C., Raza, S., Rosenbaum, S., den Veyver, I., Milosavljevic, A., Gevers, D., Huttenhower, C., Petrosino, J. and Versalovic, J.: A Metagenomic Approach to Characterization of the Vaginal Microbiome Signature in Pregnancy, PLoS One, 7(6), e36466, doi:10.1371/journal.pone.0036466, 2012.

Ackerman, M. E., Crispin, M., Yu, X., Baruah, K., Boesch, A. W., Harvey, D. J., Dugast, A. S., Heizen, E. L., Ercan, A., Choi, I., Streeck, H., Nigrovic, P. A., Bailey-Kellogg, C., Scanlan, C. and Alter, G.: Natural variation in Fc glycosylation of HIV-specific antibodies impacts antiviral activity, J. Clin. Invest., 123(5), 2183–2192, 2013.

Boyle, J. S., Klein, S. A., Lucas, D. D., Ma, H. Y., Tannahill, J., & Xie, S. The parametric sensitivity of CAM5's MJO. J. Geophys. Res. Atmos. 120(4), 1424-1444, 2015.

Breiman, L.: Random forests, Mach. Learn., 5–32, doi:10.1023/A:1010933404324, 2001.

Buday, B., Pach, F. P., Literati-Nagy, B., Vitai, M., Vecsei, Z. and Koranyi, L.: Serum osteocalcin is associated with improved metabolic state via adiponectin in females versus testosterone in males. Gender specific nature of the bone-energy homeostasis axis., Bone, 57(1), 98–104, doi:10.1016/j.bone.2013.07.018, 2013.

Duro, D. C., Franklin, S. E. and Dubé, M. G.: Multi-scale object-based image analysis and feature selection of multi-sensor earth observation imagery using random forests, Int. J. Remote Sens., 33(14), 4502–4526, 2012.

Herrera, C. M. and Bazaga, P.: Epigenetic correlates of plant phenotypic plasticity: DNA methylation differs between prickly and nonprickly leaves in heterophyllous Ilex aquifolium (Aquifoliaceae) trees, Bot. J. Linn. Soc., 171(3), 441–452, 2013.

Kohavi, R. and John, G. H.: Wrappers for feature subset selection, Artif. Intell., 97(1-2), 273–324, doi:http://dx.doi.org/10.1016/S0004-3702(97)00043-X, 1997.

Kursa, M., Rudnicki, W., Wieczorkowska, A., Kubera, E., & Kubik-Komar, A. Musical instruments in random forest. In Foundations of Intelligent Systems, LNCS 5722, 281-290. Springer Berlin Heidelberg, 2009.

Kursa, M. B., Jankowski, A. and Rudnicki, W. R.: Boruta - A system for feature selection, Fundam. Informaticae, 101(4), 271–285, 2010.

Kursa, M. B. and Rudnicki, W. R.: Feature Selection with the Boruta Package, J. Stat. Softw., 36(11), 1–13 [online] Available from: http://www.jstatsoft.org/v36/i11/paper, 2010.

Leutner, B. F., Reineking, B., Müller, J., Bachmann, M., Beierkuhnlein, C., Dech, S. and Wegmann, M.: Modelling forest α-diversity and floristic composition - on the added value of LiDAR plus hyperspectral remote sensing, Remote Sens., 4(9), 2818–2845, 2012.

Liaw, A. and Wiener, M.: Classification and Regression by randomForest. R News 2(3), 18—22, 2002.

Lucas, D. D., Klein, R., Tannahill, J., Ivanova, D., Brandon, S., Domyancic, D. and Zhang, Y.: Failure analysis of parameter-induced simulation crashes in climate models, Geosci. Model Dev., 6(4), 1157–1171 [online] Available from: http://www.geosci-model-dev.net/6/1157/2013/\npapers2://publication/doi/10.5194/gmd-6-1157-2013, 2013.

Ma, J., Prince, A. L., Bader, D., Hu, M., Ganu, R., Baquero, K., Blundell, P., Alan Harris, R., Frias, A. E., Grove, K. L. and Aagaard, K. M.: High-fat maternal diet during pregnancy persistently alters the offspring microbiome in a primate model., Nat. Commun., 5(May), 3889 [online] Available from: http://www.ncbi.nlm.nih.gov/pubmed/24846660, 2014.

Menikarachchi, L. C., Cawley, S., Hill, D. W., Hall, L. M., Hall, L., Lai, S., Wilder, J. and Grant, D. F.: MolFind: A Software Package Enabling HPLC/MS-Based Identification of Unknown Chemical Structures, Anal. Chem., 84(21), 9388–9394, doi:10.1021/ac302048x, 2012.

R Development Core Team: R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org, 2008.

Rudnicki, W. R., Wrzesień, M. and Paja, W.: All Relevant Feature Selection Methods and Applications, in Feature Selection for Data and Pattern Recognition, edited by U. Stańczyk and C. J. Lakhmi, pp. 11–28, Springer-Verlag Berlin Heidelberg, Berlin., 2015.

Saulnier, D. M., Riehle, K., Mistretta, T.-A., Diaz, M.-A., Mandal, D., Raza, S., Weidler, E. M., Qin, X., Coarfa, C., Milosavljevic, A., Petrosino, J. F., Highlander, S., Gibbs, R., Lynch, S. V, Shulman, R. J. and Versalovic, J.: Gastrointestinal microbiome signatures of pediatric patients with irritable bowel syndrome., Gastroenterology, 141(5), 1782–91, doi:10.1053/j.gastro.2011.06.072, 2011.

Smith, R., Jones, P., Briegleb, B., Bryan, F., Danabasoglu, G., Dennis, J., Dukowicz, J., Eden, C., Fox-Kemper, B., Gent, P., Hecht, M., Jayne, S., Jochum, M., Large, W., Lindsay, K., Maltrud, M., Norton, N., Peacock, S., Vertenstein, M. and Yeager, S.: The Parallel Ocean Program (POP) reference manual: Ocean component of the Community Climate System Model (CCSM), LAUR-10th–01 ed., Los Alamos National Laboratory. [online] Available from: http://nldr.library.ucar.edu/repository/collections/OSGC-000-000-000-954, 2010.

Strempel, S., Nendza, M., Scheringer, M. and Hungerbühler, K.: Using conditional inference trees and random forests to predict the bioaccumulation potential of organic chemicals, Environ. Toxicol. Chem., 32(5), 1187–1195, 2013.

UCAR: The Community Climate System Model Version 4, [online] Available from: http://www.cesm.ucar.edu/models/ccsm4.0/ (Accessed 31 March 2015), 2010.

Vapnik, V. N.: The Nature of Statistical Learning Theory., 1995.

Table 1. Summary of results.

The variables indicated as important by Lucas et al. are marked with *, the variables that were indicated as important in the first test are highlighted in bold face. Δ(AUC) is given in 0.0001 units.

Three values are reported, the number of times the variable was deemed relevant, mean difference in AUC due to adding variable to set of variables and number of times AUC was improved by adding variable to set of variables. The first value is reported for all variables, two other are reported only for these variables that were deemed relevant significantly more often than randomised variables. The unit for Δ(AUC) is 0.0001.

Variable	**V1***	**V2***	V3	**V4***	V5*	V6	Reference
# relevant	**660**	**660**	0	**44**	19	33	25±9
Mean Δ(AUC)	**905 ± 80**	**749 ± 90**	—	20 ± 70	—	—	
# improved	**30**	**30**	—	16	—	—	
Variable	V7	V8	**V9**	V10	V11	V12	Reference
# relevant	2	17	**62**	11	3	5	25±9
Mean Δ(AUC)	—	—	60±70	—	—	—	
# improved	—	—	**22**	—	—	—	
Variable	**V13***	**V14***	V15	**V16***	V17*	V18	Reference
# relevant	**593**	**623**	26	**67**	19	2	25±9
Mean Δ(AUC)	11 ± 60	**180 ± 80**	—	6 ± 60	—	—	
# improved	16	**26**	—	**14**	—	—	

Table 2. Results of experiment 2.

Average AUC obtained for all tested models, as well as examples for five interesting cases. #1 – the sample with lowest AUC from core model, #12 the sample with highest AUC obtained in the study, samples #6, #22 and #30 – samples with core model close to the mean that show variance of AUC for other models.

	Sample					
Variable set	#1	#6	#22	#30	#12	Average
core	0.865	0.921	0.922	0.928	0.983	0.925 ± 0.006
core+V4	0.879	0.915	0.954	0.930	0.982	0.927 ± 0.007
core+V9	0.866	0.923	0.945	0.919	0.989	0.931 ± 0.006
core+V16	0.848	0.906	0.938	0.927	**0.990**	0.926 ± 0.007
core-V14	0.823	0.907	0.926	0.919	0.967	0.907 ± 0.007
core-V13	0.877	0.910	0.952	0.921	0.968	0.924 ± 0.006
core-V1	0.745	0.821	0.806	0.823	0.910	0.835 ± 0.007
core-V2	0.808	0.808	0.825	0.840	0.888	0.850 ± 0.009

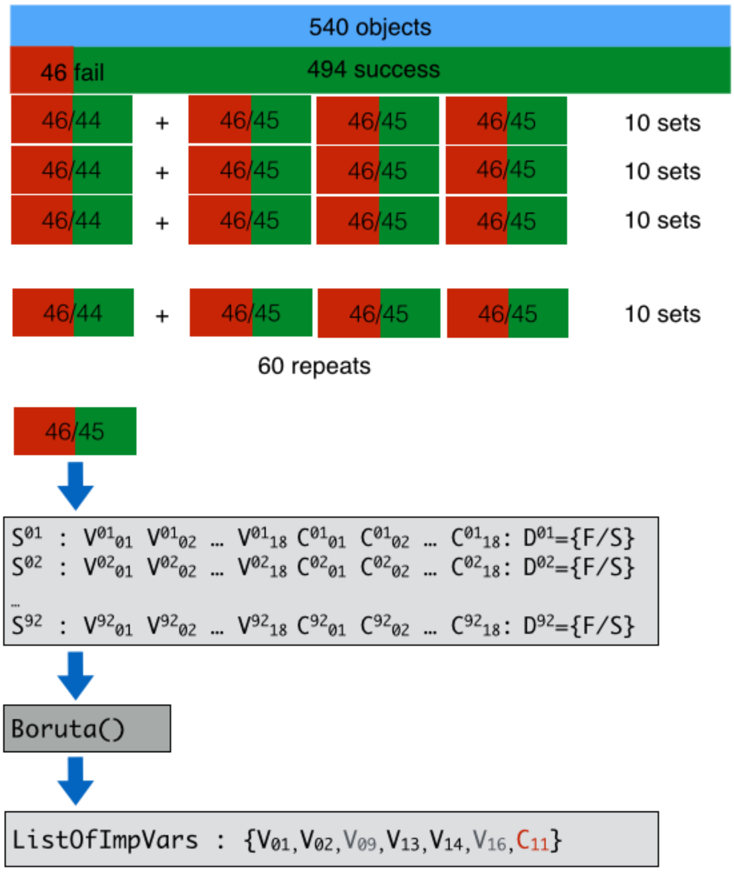

Figure 1. Protocol of the first test.

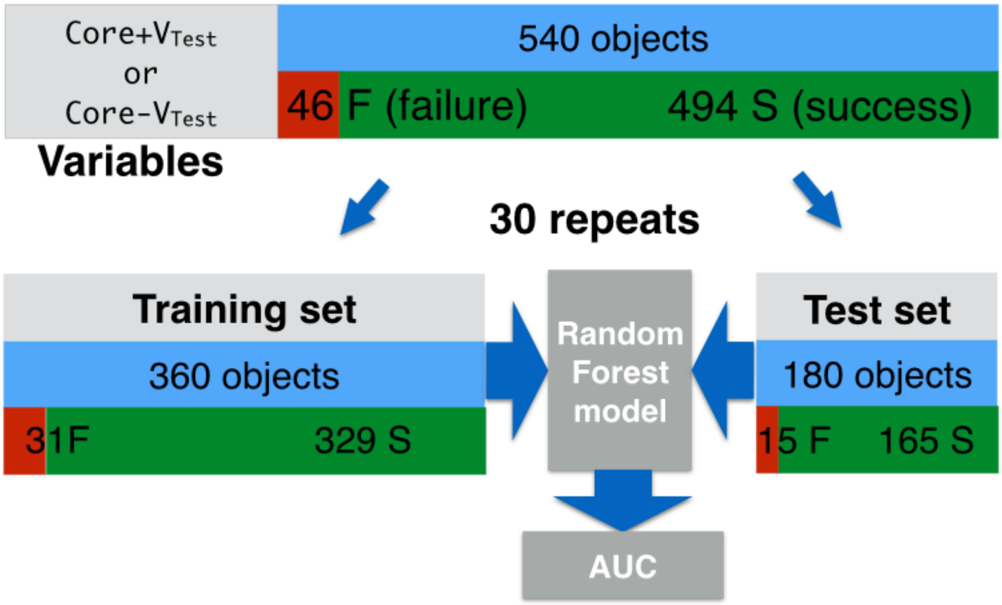

Figure 2. Protocol of the second test.

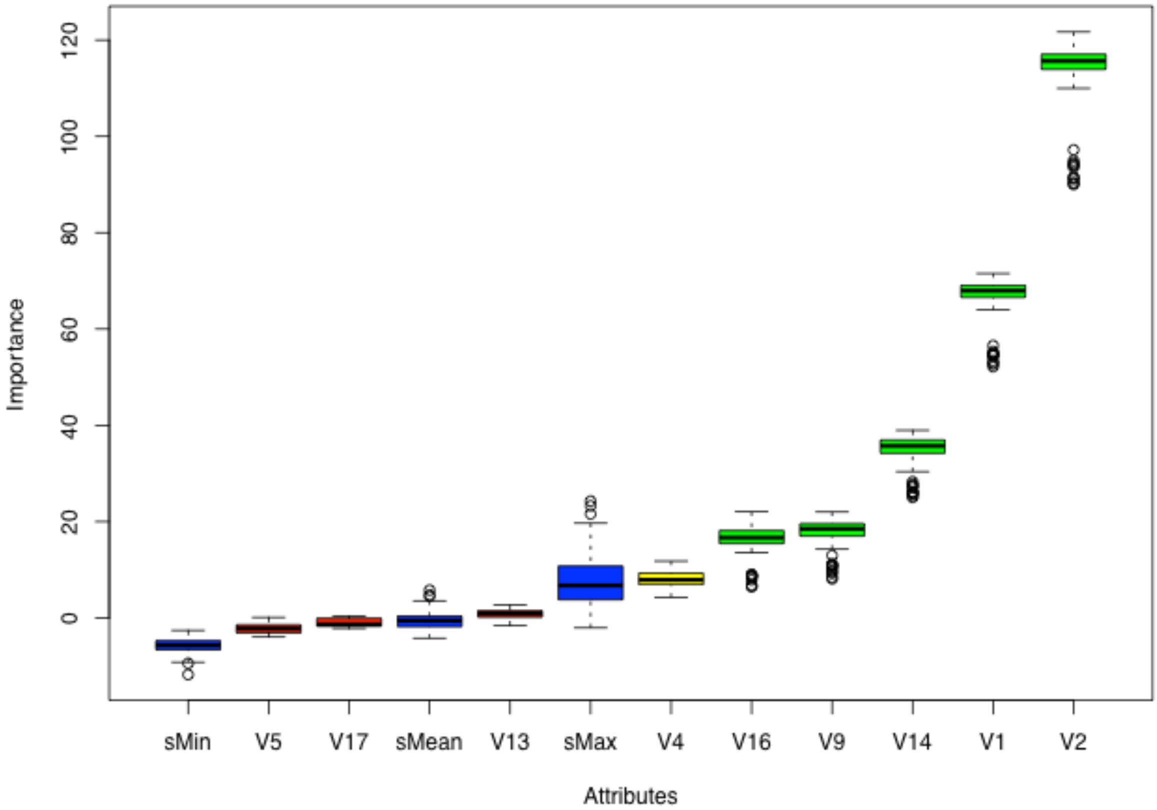

Figure 3. Summary of results of the Boruta run. Importance of the variables is shown. The variables are sorted by increasing importance. The variables coloured in green are these, which were classified as relevant. Variables coloured in red are these, which are irrelevant. The blue boxes correspond to respectively minimal (sMin), median (sMed) and maximal (sMax) importance achieved in each run by contrast variables. One can observe wide range of maximal importance values that can be achieved by random variables. In particular in many iterations it can be higher than importance of truly relevant variables.

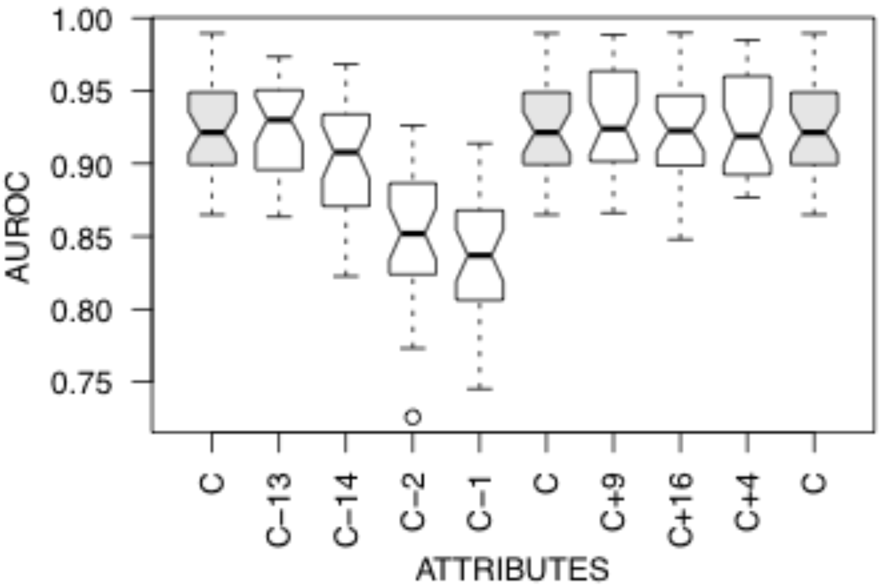

Figure 4. AUC obtained in simulations study grouped by subset of variables used for model building. The labels are coded in the following way C – core set of variables {V1, V2, V13, V14}; C+X – the core set was extended by adding variable VX, where X is one of {4,9,16}; C-X – the variable VX was removed from the core set, with X = {1,2,13,14}.

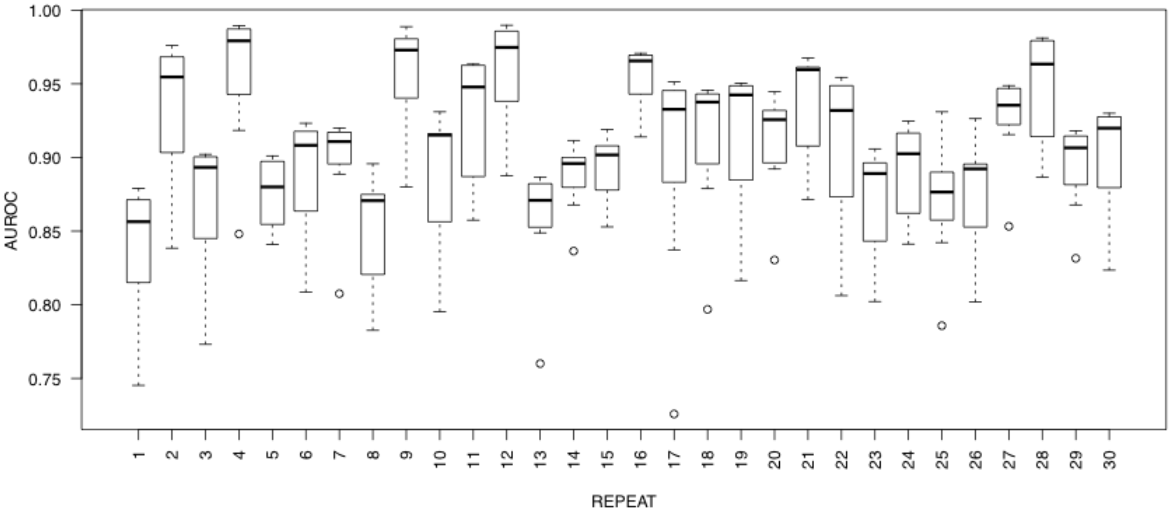

Figure 5. AUC obtained in simulations study grouped by split between training and validation set.