Variable selection wrapper in presence of correlated input variables for Random Forest models

Marta Rotari[a], Murat Kulahci[a,b]

[a]Department of Applied Mathematics and Computer Science, Technical University of Denmark, Kgs. Lyngby, Denmark
[b]Department of Business Administration, Technology and Social Sciences, Luleå University of Technology, Luleå, Sweden

Abstract

In most data analytic applications in manufacturing, understanding the data-driven models plays a crucial role in complementing the engineering knowledge about the production process. Identifying relevant input variables, rather than only predicting the response through some "black-box" model, is of great interest in many applications. There is, therefore, a growing focus on describing the contributions of the input variables to the model in the form of "variable importance", which is readily available in certain machine learning methods such as random forest (RF). Once a ranking based on the importance measure of the variables is established, the question of how many variables are truly relevant in predicting the output variable rises. In this study, we focus on the Boruta algorithm, which is a wrapper around the RF model. It is a variable selection tool that assesses the variable importance measure for the RF model. It has been previously shown in the literature that the correlation among the input variables, which is often a common occurrence in high dimensional data, distorts and overestimates the importance of variables. The Boruta algorithm is also affected by this resulting in a larger set of input variables deemed important. To overcome this issue, in this study, we propose an extension of the Boruta algorithm for the correlated data by exploiting the conditional importance measure. This extension greatly improves the Boruta algorithm in the case of high correlation among variables and provides a more precise ranking of the variables that significantly contribute to the response. We believe this approach can be used in many industrial applications by providing more transparency and understanding of the process.

Keywords: Random Forest; Conditional Importance; Variable selection algorithm; Boruta Algorithm; Additive manufacturing

1. Introduction

In many industrial applications, the production machines are equipped with several sensors in each stage of the production process, resulting in large amounts of data being collected. It is crucial to determine which variables from the process data actually influence the response variable, which is typically related quality of the final product. In this context, the interest is focused on the development and use of Machine Learning methods aimed at identifying the contribution of each process variable to the model, also known as variable importance, and in identifying the all-relevant variables, also known as variable selection, which play a crucial role in the mechanism of learning algorithms for predicting the response variable.

Variable selection is a challenging issue in high-dimensional regression and classification problems. Most of the input variables are usually irrelevant, and their relevance is unknown in advance. Typically, there are three goals in variable selection. The first is purely technical; dealing with large sets of variables slows down algorithms, consumes excessive resources and is inefficient [1, 2]. A second aim is to find a small number of variables that maximize the model accuracy [3]. Numerous machine learning algorithms show a reduction in accuracy when the number of variables is significantly higher than optimal [4]. Therefore it is preferable to select the smallest set of variables that yields the best model. This problem, known as the minimal-optimal problem, has been explored extensively [5].

The third aim is to improve understanding of the underlying process that generated the data [6]. The identification of all variables, which are in some circumstances relevant for classification or regression purposes, is the so-called all-relevant problem. The goal is to identify and prioritize all influential variables for further investigation with domain expertise. This is especially important when the goal is to understand mechanisms related to the subject of interest rather than simply building a black box predictive model. In the biomedical research, for example, when dealing with the results of gene expression measurements in the context of particular diseases, identifying an influential set of genes as genetic markers might be useful. In manufacturing, detecting all relevant variables of the production process could be very pertinent to understand and have a better overview of the process. It could complement the engineering knowledge about the production process and facilitate process improvement effects. Further details on the relevance of variable selection are described in Nilsson et al. [5].

Variable selection gets considerably more challenging in the presence of strongly correlated input variables, as it is common in real-world data. Consider two relevant variables which are strongly correlated. The second and third aims may seem similar but lead to different outcomes depending on which of these two objectives is of interest. The variables convey the same statistical information, so only one should be chosen if the goal is to maximize predictive accuracy with a small number of variables, second aim. On the other hand, these two variables may be collected in different ways and represent distinct physical quantities. Consequently, domain experts may interpret them differently, hence both should be preserved.

In machine learning for variable selection in the presence of high-dimensional data, Random Forest (RF) [7] model has been commonly used as a modeling method. The RF model consists of a collection of trees created using bagging or bootstrap aggregation. It is a nonparametric model for classification and regression issues. It has been applied to a wide range of problems, including high-dimensional problems with multi-class responses, categorical variables, imbalanced data and multiple adapted versions have been proposed [8, 9, 10, 11, 12]. The model is widely used due to its capacity to internally consider the interaction between input variables while providing variable importance measures. For variable importance in RF models, two types of measures have been proposed: the Mean Decrease Accuracy (MDA) and the Mean Decrease Impurity (MDI). Both measures are non-parametric and there have been several studies considering the theoretical formalization of these methods [13] [14] [15] [16] [17]. Nonetheless, further studies have been conducted on the influence of correlation on both variable importance measures [18] [19] [20] [21], which conclude that correlation among the input variables leads to a significant overestimation of the importance scores. According to simulation studies carried in [18] [22] [23], highly correlated variables can erroneously show high MDA scores even when there is no dependence between the response variable and these variables. The MDA may fail to detect some relevant variables in the presence of correlation among the variables [24] [25] [26] [20] [21] [23].

While RF provides variable importance values, there is no built-in solution for variable selection based on a variable importance threshold. In the industrial context, the final decision on the subset of selected variables is manifest for the complete understanding of the processes. Moreover, a precise threshold is crucial to identify the most important variables without discarding any relevant information. Many techniques have been suggested on how to discard non-significant variables [17] [25] [4] [27]. The most successful algorithms for this purpose are the wrapper methods [24] [28] [29, 30] [31] [32] [33] [34], which return a final subset of all-relevant variables. To efficiently use a wrapper method, the model to be used should be both computationally efficient and simple, with no user-defined parameters if possible, which is the case for the RF models.

Boruta algorithm [35] is a wrapper method that aims to identify a clear threshold in the variable importance ranking provided by the RF model. It uses the MDA and MDI importance measures. Recently, a novel Python implementation of Boruta has been proposed, allowing the selection of any Tree-based learner [36]. Such importance measures quantify the relevance of an input variable towards a response variable by perturbing the values of the former. However, these do not consider the correlation among the inputs, therefore, the performance of the Boruta algorithm is compromised when correlated input variables are present. In particular, the problem of correlation leads to overestimated variable importance values. This overestimation can result in an incorrect ranking of the variables, producing a larger set of input variables deemed important.

In order to adjust the variable selection result in the presence of correlated input variables, we present an extension of the Boruta algorithm to effectively exploit the advantages of RF models with wrapper variable selection methods. This extension uses a conditional variable importance measure. The proposed algorithm significantly improves the variable importance ranking that results in a more precise final variable selection in comparison to the existing Boruta algorithm. In the following, we present the proposed extension of the Boruta algorithm and application on simulated dataset and on a real-world case in additive manufacturing to highlight the advantage of the current proposal.

2. Methods

2.1. Model-based variable importance and variable selection

The first category of variable selection methods are modeling techniques that have the selection already built-in. The two most predominant in linear regression are Lasso and Elastic Net [37, 38]. These methods make use of regularization to provide a measure of variable importance. The Lasso method includes a penalty term in its optimization criterion, which constrains the size of the estimated coefficients. As the penalty increases, the coefficients with the lowest values are set to zero. Similar to Lasso, Elastic Net can build reduced models by producing zero-value coefficients. Both methods produce reduced models with only the relevant variables.

Some Machine Learning methods, such as Random Forest (RF) [7], have embedded variable importance measures. Specifically, there are two commonly employed variable importance measures: mean decrease in impurity (MDI) also called the Gini importance and mean decrease in accuracy (MDA) [7]. The former relies on the concept of impurity reduction followed in most traditional classification tree algorithms. First, the weighted impurity decrease over all nodes that split on a given variable are summed. The value is then averaged over all trees in the forest to obtain the MDI value. Furthermore, other analyses as in [26] and [39] have highlighted that MDI is consistent only under a strong and restrictive assumption of an additive regression function and independence of the input variables. Strobl et al. [26] claim that the MDI variable importance measure is biased when input variables vary in their number of categories or scale of measurement. On the other hand, MDA, also called permutation importance, is widely considered a more efficient variable importance measure for Random Forests [13] [26]. This method can be more computationally intensive than MDI, but it may provide a more accurate estimate of variable importance when there are interactions among variables or when correlation among variables is present [40]. This paper focuses on regression settings and correlated data; therefore, the MDA measure will be considered.

Consider the input variables $X = (X_1, .., X_j, .., X_p)$ and the response variable Y. In the Random Forest algorithm, each tree $t \in 1, \cdots, n_{tree}$ is built with a bootstrap sample of the original data, $\{(X_1, Y_1), \cdots, (X_n, Y_n)\}$ where $X_i = (X_{i1}, \cdots, X_{ip})$. The leftover data from the bootstrap sample represents the out-of-bag (OOB) data, that we denote by $\mathbb{B}^{(t)}$. The OOB sample is used to evaluate the model prediction performance, also referred to as OOB error:

$$e\hat{r}r(\hat{f}^{(t)}, \mathbb{B}^{(t)}) = \frac{1}{|\mathbb{B}^{(t)}|} \sum_{i:(X_i,Y_i)\in\mathbb{B}^{(t)}} (Y_i - \hat{f}^{(t)}(X_i))^2 \tag{1}$$

where $\hat{f}^{(t)}(X_i)$ is the prediction for observation i and $|\cdot|$ is the cardinality function. To compute the MDA for variable X_j for a single tree, the values of variable X_j are permuted, yielding $\mathbb{B}^{(t,\pi_j)}$, following permutation π. The OOB error is computed again. The difference between the OOB error for the new data and the original OOB error is defined as the importance of the j-th variable for the tree t. The average of this difference over all trees in the forest constitutes the MDA importance value for variable X_j and can be written as:

$$I(X_j) = \frac{1}{n_{tree}} \sum_{t=1}^{n_{tree}} \left(e\hat{r}r(\hat{f}^{(t)}, \mathbb{B}^{(t,\pi_j)}) - e\hat{r}r(\hat{f}^{(t)}, \mathbb{B}^{(t)}) \right) \tag{2}$$

It is worth noticing that various MDA implementation exists. In standard random forest implementations, an additional version of the permutation importance (often referred to as the z-score) is obtained by dividing the importance by its standard error. Bérnard et al. [14] provide an exhaustive analysis of the various MDA implementations. Besides providing a more rigorous formalization, the authors propose a new, augmented version called Sobol-MDA that offers improved reliability and accuracy in measuring the variable's importance scores.

The random permutation of the X_j values breaks the initial relationship between the independent variable Y and the dependent variable X_j. It should simulate the lack of the variable in the model. If the original variable X_j is associated with the response variable, the OOB error increase when the permuted variable X_j and the other non-permuted input variables are used to predict the response for the OOB observations. The difference between the accuracy of prediction before and after permuting the values of X_j can be viewed as a measure of the importance of X_j in predicting the response variable Y. When there is almost no difference in the accuracy of the forecast before and after permutingX_j, X_j is said to be unimportant [25, 26, 18].

2.2. Variable selection based on the Boruta algorithm

The Boruta algorithm [35] [36] [41] is a wrapper algorithm developed for the RF model. In a wrapper method, an algorithm is used as a black box that returns a variable ranking. The random forest regression and classification algorithm is relatively quick, can usually be run without tuning any parameters, it is sensitive to the interaction between variables without explicit settings and gives a numerical estimate of the variables' importance. The Boruta algorithm is a sequential selection algorithm, and each step consists in running RF and obtaining the variables' importance values. It offers a precise threshold in the RF variables ranking in order to decide a final set of relevant variables to predict the output variable.

The algorithm is an extension of the original proposal in [17], that determines the relevance of a variable by comparing the importance of the original variables to that of the random noise variables. Based on the values obtained from the RF model, Boruta evaluates which variables are relevant. The reference for deciding which variables are truly relevant is given by the set of the noise variables. The main goal is to find all variables for which their association with the response variable is higher than that of the noise variables. Numerous RF realizations in the Boruta algorithm produce a more stable output than a single RF run. The original algorithm description can be found in [35] [41].

2.3. The influence of the correlation on the permutation importance measure

Strobl et al. [18] formalize the MDA interpretation under the assumption of no correlation among input variables and the response variable Y. They argue that if there is independence between the variable X_j and the response Y and marginal independence between X_j and the other variables $X \setminus X_j$, then the permutation of X_j would not affect the prediction accuracy. An MDA importance value close to zero validates the hypothesis of marginal independence. Consequently, a large value can be indicative of dependence between X_j and Y or between X_j and other variables, or both.

The effect of the correlation among input variables on the MDA measure and its bias has been studied in the literature [18] [19] [20] [21]. However, there is no agreement on how to interpret the importance measures when the input variables are correlated and even less agreement on how this correlation affects the importance measures [42] [43]. Nicodemus et al. [22] show through simulated studies that highly correlated variables acquire high MDA values even when there is no dependence on the response variable. Strobl et al. [18] highlighted two issues in the high MDA values of correlated variables. The first reason is identified in the tree-building process that prefers the selection of correlated variables. The second is identified in the computation of the MDA value and the advantage of the correlated data induced by the unconditional permutation scheme. Toloşi and Lengauer [23] identify this effect as "correlation bias", which does not correspond to a statistical bias. They observe a critical effect of the correlation on the permutation importance measure that depends on the size of the correlated group. Similarly, other empirical studies show that MDA fails to identify some of the relevant variables when highly correlated variables are present [24] [25] [26] [20] [21] [23].

Following that, Gregorutti et al. [25] provides a more formal description and proof of this effect, limited to a regression setting. In the case of an additive regression model, it is possible to express the permutation importance measure as a function of the correlation between input variables. Consider (X, Y) random vectors which satisfy the following additive regression model

$$Y = \sum_{j=1}^{p} f_j(X_j) + \epsilon \tag{3}$$

where ϵ is such that $\mathbb{E}[\epsilon|X] = 0$ and f_js are measurable functions. In this model setting, for any $j \in 1, \cdots, p$ the permutation importance measure satisfies

$$I(X_j) = 2\mathbb{V}[f_j(X_j)]. \tag{4}$$

That is, the variable importance score of the X_j variable is equivalent to two times the variance of $f_j(X_j)$. Assume moreover that for some $j \in 1, \cdots, p$ the variable $f_j(X_j)$ is centered. Then

$$I(X_j) = 2\mathbb{C}[Y, f_j(X_j)] - 2\sum_{i \neq j} \mathbb{C}[f_j(X_j), f_i(X_i)]. \tag{5}$$

where $\mathbb{C}$ denotes the covariance. Note that equation 5 reveals the strong dependence on the additive structure of the regression function f.

Even if restricted to special model settings, the above equations describe how the correlation among the variables impacts the MDA. Therefore the correlation among the variables is to be considered when interpreting the variable importance values in Random Forest models. Furthermore, high correlation among input variables leads to a considerable overestimation of the variable importance values of the correlated variables. Consequently, all correlated but non-influencing variables will be highly ranked in importance. The Boruta algorithm is also highly affected by this effect as RF and in particular its measures are the fundamental of Borutas' algorithm. The algorithm deem a large number of variables to be relevant. The majority of the selected variables are variables with a high empirical correlation. As a consequence, some irrelevant variables might be deemed relevant and some variables with low relevance might be missed. We next present the extension of the Boruta algorithm involving the correlation among the input variables.

2.4. Extension of Boruta algorithm to the case of high correlated input variables

2.4.1. The use of conditional variable importance

Strobl et al. [18] propose a new importance measure for RF models that is based on a conditional permutation scheme that better reflects the impact of input variables on the response variable when a high correlation among the input variables is present. The aim is to evaluate the deviation from the null hypothesis, that X_j and Y are independent based on the correlation structure between X_j and the other variables.

The strategy of this new measure is to build a conditional permutation scheme in the dataset based on the correlation among variables, with the aim to preserve the data correlation structure. To calculate the variable importance of a variable X_j, the values of X_j are permuted based on the so-called conditional permutation scheme. The nature of conditional permutation comes from the fact that X_j is permuted only within groups of observations with $Z^{(j)} = z$, where $Z^{(j)}$ represent the set of variables that are correlated with X_j. The exact variables that should be included in the subset $Z^{(j)}$ are those whose correlation with X_j exceeds a certain threshold. Another option is to allow the user to specify which variables to condition on, for example, if a hypothesis of interest contains certain independencies.

We present here the derivation of the conditional mean decrease in accuracy (CMDA). Consider the input variables $X = (X_1, \cdots, X_j, \cdots, X_p)$ and the output variable Y. First, the calculation of CMDA is done for every tree of the RF model. In every tree in the model $t \in 1, \cdots, n_{tree}$ the predictions for the OOB data $B^{(t)}$ are calculated. Next we proceed with the following construction of the conditional permutation scheme:

1. Select a subset of variables Z to base the conditional permutation on. The subset is composed of variables that show high correlation with variable X_j greater than a predefined threshold. $Z^{(j)} = \{X_i \mid cor(X_i, X_j) > threshold, X_i \in X \setminus X_j\}$
2. Collect the split points from the tree t of the RF model for $Z^{(j)}$.
3. Use the split points to create a grid that bisects the input variable space.
4. Within the obtained grid permute the values of X_j and compute the OOB-prediction error again.

The value of CMDA for variable X_j for one tree corresponds to the difference in prediction accuracy before and after this conditional permutation. We then repeat the procedure for every tree in the forest ($t \in 1, \cdots, n_{tree}$). The final CMDA value of variable X_j is given by the average over all CMDA values calculated from every tree as:

$$I(X_j) = \frac{1}{n_{tree}} \sum_{t=1}^{n_{tree}} \left(e\hat{r}r(\hat{f}^{(t)}, \mathbb{B}^{(t,\pi_j|Z)}) - e\hat{r}r(\hat{f}^{(t)}, \mathbb{B}^{(t)}) \right) \tag{6}$$

where $e\hat{r}r(\cdot, \cdot)$ is defined in (1).

In the presence of strongly correlated variables, this new technique is able to determine the variable importance values more accurately. In fact, CMDA limits the variable importance values of correlated data; thus, variables with a stronger impact on the response variable have a greater chance of being identified. Even if this approach fails to totally eliminate the influence of correlated variables, by employing this new variable importance measure technique, we can obtain a more accurate representation of the variables' relevance in relation to the response variable. Compared to MDA, this method has shown substantial improvement in circumstances of highly correlated data [18].

2.4.2. Extended Boruta algorithm

Using the CMDA importance measure, we present here the complete definition of the extended Boruta algorithm. Consider the input space $X_1, \cdots, X_p$ and the response variable Y. In this algorithm the variables are classified as relevant, irrelevant and tentative. A variable that is classified as *relevant* is considered a selected variable. A variable that is classified as *tentative* is a variable neither selected nor rejected. Finally, a variable classified as *irrelevant* is a rejected variable. Just as in the case of the Boruta algorithm the noise variables are used as a benchmark for the importance of the input variables. The algorithm starts by marking all input variables as *"Tentative"* variables. The algorithm runs for a number of iterations until all the variables are classified or the maximum number of iterations is reached. At each iteration, the algorithm proceeds as follows:

1. The dataset is expanded by including noise variables. The noise variables are obtained by shuffling the values of at least 5 initial variables selected from variables marked as tentative.
2. With the expanded dataset an RF model is built.
3. The obtained RF model is used to calculate the CMDA value for each variable of the extended dataset, as explained in section 2.4.1.
4. The testing threshold is set. This threshold corresponds to the maximum of the CMDA values of the noise variables.
5. The CMDA values of each X_j variable is compared with the testing threshold. If the CMDA$_j$ value exceeds the testing threshold, the variable X_j is given a score point of 1 or 0 otherwise.
6. The obtained scores for all input variables are added to the scores from the previous iterations. This produces a vector called hints $H = (h_1, \cdots, h_j, \cdots, h_p)$, $H \in \mathbb{N}^p$.
7. A classification of the input variables is made based on the hints vector H. The classification follows the Binomial decision scheme, discussed below.
8. If not all the variables are classified as relevant or irrelevant, a new iteration starts. Otherwise the algorithm ends.

The Binomial decision scheme is based on the definition of the hints vector. Each iteration of the algorithm is assumed to be an independent experiment that gives a binary outcome. The H random vector represents the number of success in a binomial distribution, with n the number of iterations and $p = 0.5$. We assess the probability that an input variable scores better than the maximum noise variable importance value. The binomial decision scheme is taken from the original Boruta algorithm. Consider a binomially distributed random variable $B \sim Bi(n, 0.5)$ where n is the number of iterations and p is equal to 0.5. Using a significance level α, the decision proceeds as follows:

1. To classify variable X_j as *relevant* we evaluate if the obtained scores h_j:

   $$P(B > h_j - 1) < \alpha \tag{7}$$

   where $P(B > h_j - 1) = 1 - P(B \leq h_j - 1)$. The obtained p-value is adjusted with the Bonferroni correction and then compared to the significance level α.
2. To classify variable X_j as *irrelevant*, we evaluate if:

   $$P(B \leq h_j) < \alpha \tag{8}$$

   Once again, the p-value is adjusted by Bonferroni correction before comparing with α.
3. If a variable does not satisfy either of the above cases, the variable remains marked as *tentative* variable.

We recall that the Bonferroni correction is an adjustment for multiple comparison tests used in statistical analysis. When performing a hypothesis test with multiple comparisons, wrongly claiming statistical significance in at least one of these comparisons is higher than the $\alpha-$level set for each comparison. Considering a family of m hypotheses for significance testing and their corresponding p-values p_i with $i = 1, \cdots, m$, the Bonferroni correction rejects the null hypothesis for each $p_i \leq \alpha/m$.

When the maximum number of iterations is reached or all of the original variables are classified as relevant or irrelevant, the algorithm terminates. It may happen that a variable has not been classified at the end of the algorithm, i.e. it remains as a *tentative* variable. The tentative variable has an importance score that is very close to the best noise variable importance value, and the Boruta algorithm is unable to make the desired decision in the default number of iterations. In this case, there are two options: increase the number of iterations or compare the median importance variable value with the maximum importance value for the noise variables. This is based on the historical variable importance stored in memory by Boruta for each iteration. For further details of these rules see [41].

The newly introduced extension requires a longer computation time than the original version. The CDMA, as described in Strobl et al.[18, 44], offers a significant advantage when dealing with highly correlated data, but requires more computational time than other variable importance measures. Despite the computational cost, the benefits of using this measure are substantial, as discussed in the following section. The decision to employ this method should be based on the correlation among the variables and the research objectives.

The proposed method's R code is available online [45], providing researchers with an accessible tool to implement the method. The R function allows for control over fundamental parameters: the significance level α, the maximum number of iterations, `mtry` the number of variables randomly selected at each split, and `ntree` the number of trees in random forest. A brief discussion on how to choose these parameters can be found in the next section.

3. Results and Discussion

This section is devoted to evaluating the proposed model's effectiveness, which we demonstrate through two simulated datasets and an industrial case study. We also include a comparison with existing models such as the original Boruta, Lasso, Elastic net, Knockoffs variable selection and VSURF. The first two models, Lasso and Elastic Net, are two commonly used regression models with regularization. Knockoffs variable selection [46, 47] is a variables selection model that uses a set of ”knockoff” variables and it is designed to control for false discovery rate. Variable Selection Using Random Forest (VSURF) [34, 48] is also a variable selection method employing a random forest-based approach.

The first simulation aims at assessing the model's performance in detecting the most significant variables among all variables in the presence of multiple groups of correlated variables. Additionally, this simulation includes a brief discussion of the model's parameters. The second simulation evaluates the model's performance when the interactions among variables are present. The utilization of simulated data in our evaluation process serves two purposes. Firstly, it allows for greater control over the correlation among the variables, enabling a more systematic investigation of the model's performance under varying conditions. Secondly, using simulated data provides a comparison of various model selection approaches. In section 3.2, we present a real-world case in additive manufacturing. The obtained results were evaluated by process engineers and compared with the existing knowledge about the process. All the analyses were performed in R language, version 4.2.3, using functions available at [45] and publicly available R packages.

3.1. Simulated Dataset

In the first simulation study, two groups of correlated variables are introduced in the dataset, and the simulations were run at increasing correlation levels from 0 to 0.9. The 20 input variables given in X are drawn from a joint normal distribution with mean 0 and variance 1. Two groups of correlated variables are introduced: $[X_1, X_2, X_3, X_4, X_5]$ and $[X_{10}, X_{11}, X_{12}, X_{13}, X_{14}]$, as shown in Figure 1. The response variable is generated as a linear combination of X' where $X' = [X_2, X_{11}, X_{19}, X_{20}]$ and the coefficients β_i for $i \in \{2, 11, 19, 20\}$ are such that $\beta_i / sd(\beta_i) = k$, where $sd(\cdot)$ is the standard deviation and $k = [4, 3, 3, 5]$. The model for Y is given as

$$Y = \beta_2 X_2 + \beta_{11} X_{11} + \beta_{19} X_{19} + \beta_{20} X_{20} + \epsilon \tag{9}$$

where ϵ represents the added noise with $\epsilon \sim N(0, 0.1)$.

We performed an analysis to assess the model sensitivity to the choice of significance level α, the number of variables randomly selected at each split (`mtry`) and the number of trees in the random forest (`ntree`). In each case we performed 50 simulation runs. The first parameter being investigated is the significance level α, for which the default value is 0.01. Figure 2 depicts a comparison between the default value of 0.01 against 0.05, a level that is also commonly used in regression.

Figure 2 demonstrates that the total number of selected variables is expectantly higher for $\alpha = 0.05$ compared to $\alpha = 0.01$. This implies that on average a greater number of significant variables are being correctly selected. However,

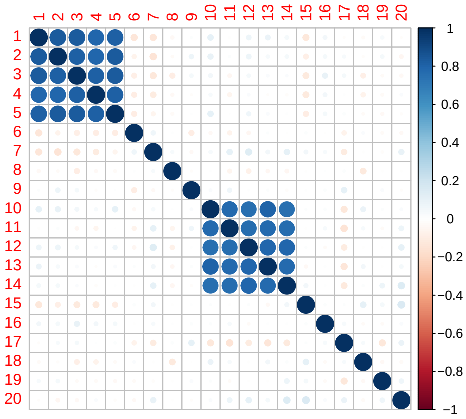
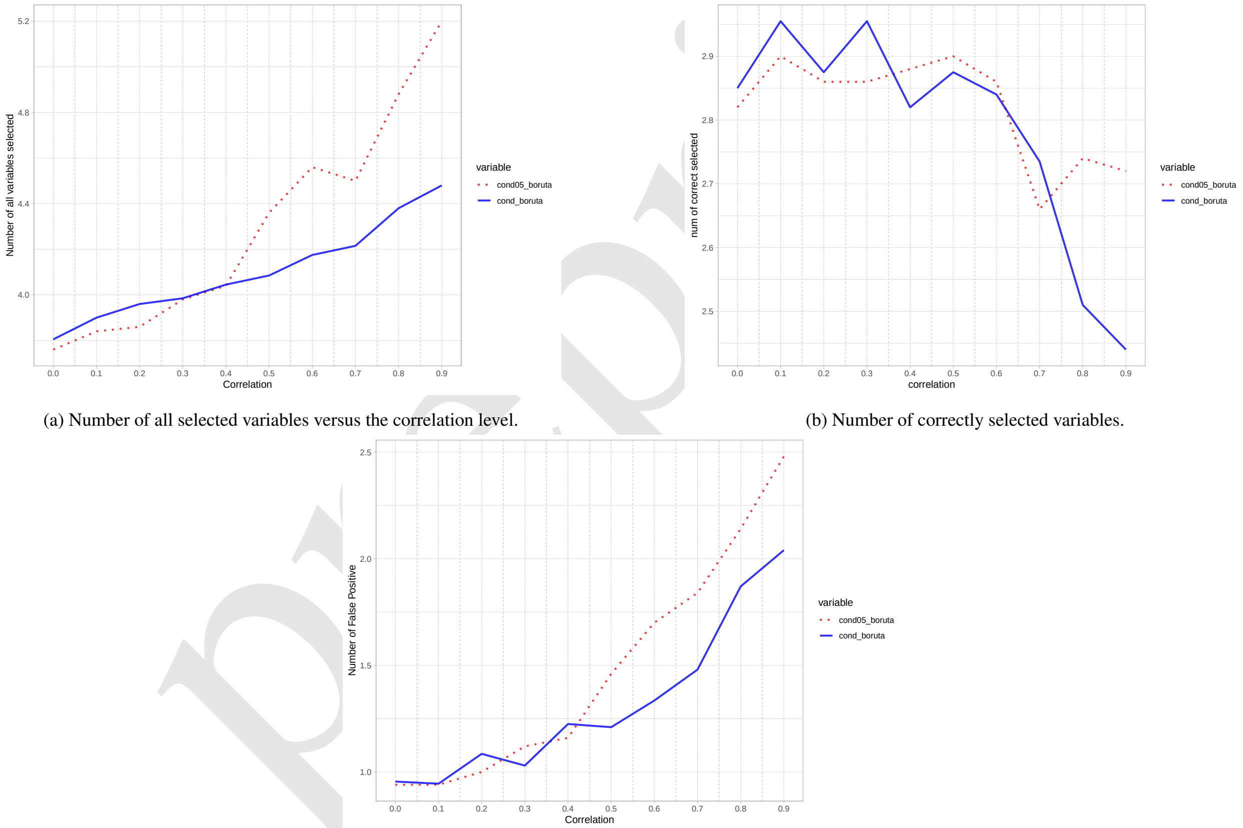


Figure 1: Correlation structure of the input variables.



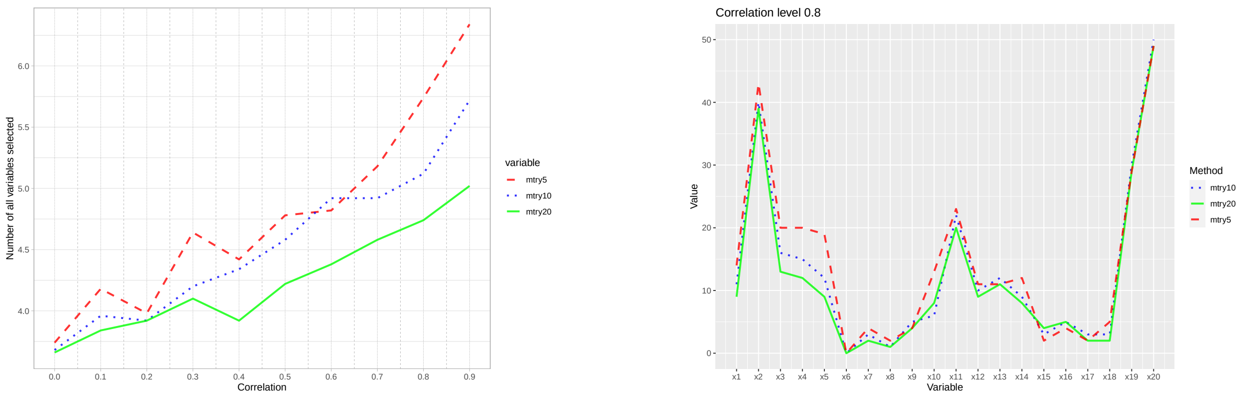
(a) Number of all selected variables versus the correlation level.

(b) Number of correctly selected variables.

(c) Number of wrongly selected variables.

Figure 2: In the solid blue line, the default value $\alpha = 0.01$ and the dotted red line $\alpha = 0.05$. (a): Number of all selected variables vs correlation. (b): Mean number of correctly selected variables over 50 runs vs correlation. (c): Number of wrongly selected variables as the correlation increases.

an increase in the number of selected variables also results in the selection of noise variables that are not relevant to the response variable, Y. Careful consideration of the positive and negative effects associated with different α-levels is necessary to make an informed decision. Determining the appropriate significance level depends on the specific case and objectives of the analysis.



(a) Number of all selected variables versus the correlation level.

(b) Total number of selected variables for 50 runs.

Figure 3: In the solid green line `mtry=20`, the dotted blue line `mtry=10` and dashed red line `mtry=5`. (a): Number of all selected variables vs correlation. (b): Number of times each variable has been selected in 50 runs.

The sensitivity of the model to the changes in `mtry` is depicted in Figure 3. As discussed in [18, 44], the CMDA method is moderately sensitive to the choice of `mtry`, albeit less so than the MDA method. This phenomenon is because correlated variables are favoured in the split selection process. Consequently, for low values of `mtry`, correlated variables are more likely to be chosen, resulting in a higher variable importance score than the uncorrelated and noise variables. As can be seen in Figure 3a, a higher number of variables are selected for low `mtry` values. This increase is partly due to the selection of noise variables that are highly correlated with the relevant variables. As shown in Figure 3b, in the case of correlation level 0.8 and low `mtry` values, the entire correlation group associated with $X2$ is selected, in addition to $X2$ itself. This effect is less pronounced for the second correlation group, as $X11$ has a lower regression coefficient. The sensitivity of the CMDA method to the choice of `mtry` suggests that careful consideration of this parameter is necessary to ensure an appropriate model.

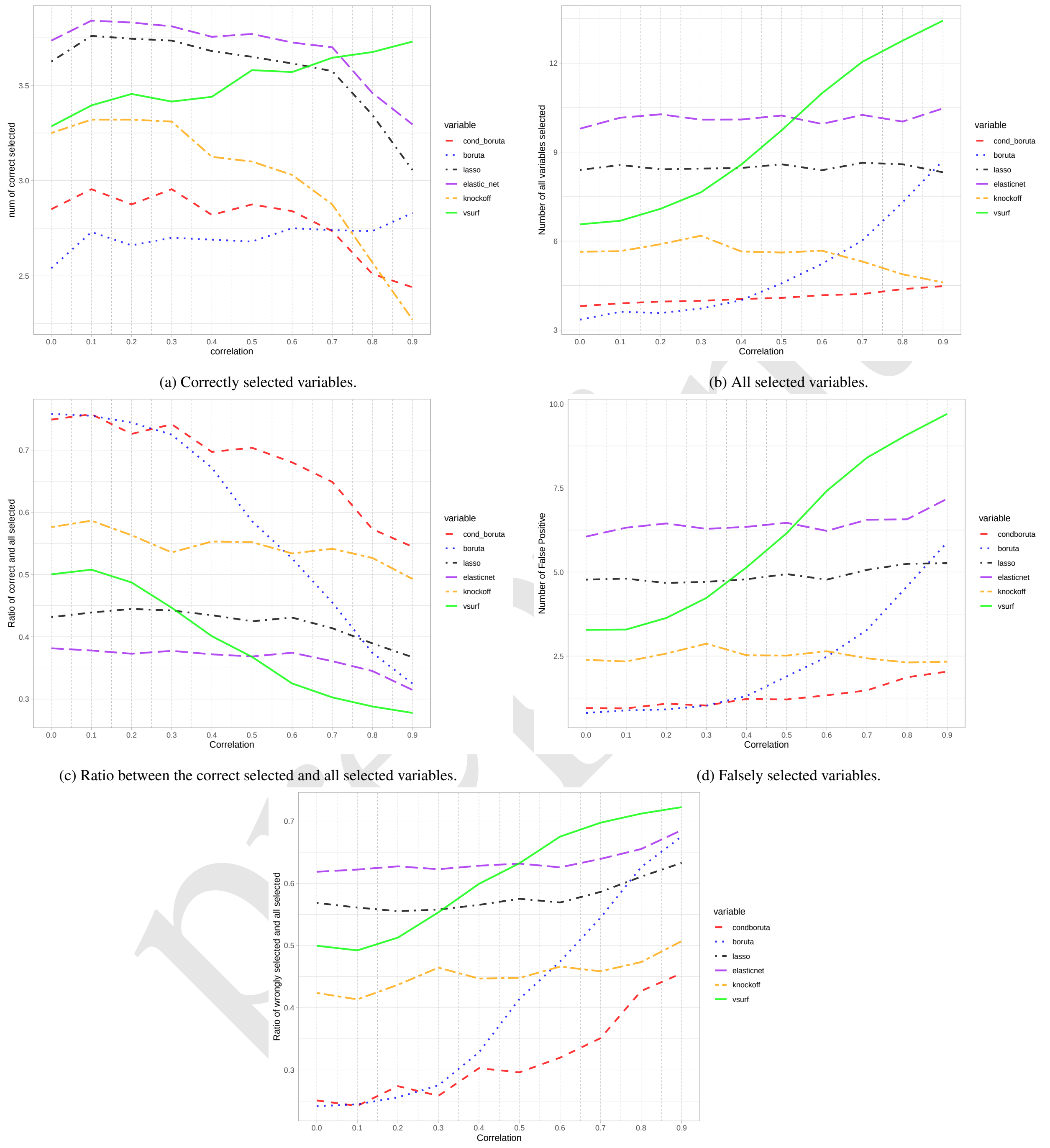
We also assessed the impact of `ntree` parameter by evaluating the model's performance for `ntree = 500, 1000` and 2000 trees. We observed no substantial changes in the model's performance as the number of trees increased. Given that the proposed method is computationally more demanding compared to the original method, we recommend using the default value of `ntree = 500`. This choice strikes a reasonable balance between model accuracy and computational efficiency.

In the second simulation study, the data set is similar, except for the addition of an interaction term in the model. That is, the response variable is generated as a linear combination of X' where $X' = [X_2, X_7X_{11}, X_{11}, X_{19}, X_{20}]$ and the coefficients β_i is such that $\beta_i/sd(\beta_i) = k$, where $sd(\cdot)$ is the standard deviation and $k = [4, 5, 3, 3, 5]$. The model for Y is

$$Y = \beta_2 X_2 + \beta_{7,11} X_7 X_{11} + \beta_{11} X_{11} + \beta_{19} X_{19} + \beta_{20} X_{20} + \epsilon \tag{10}$$

with ϵ represents the added noise with $N(0, 0.1)$. At each correlation level, the proposed model as well the original Boruta, Lasso, Elastic Net, Knockoffs and VSURF models were run 200 times each. The Lasso and Elastic Net model parameters were estimated through five-fold cross-validation. We show the model comparison results from the first simulation dateset for illustration purposes in Figure 4. The other simulation runs show similar results and hence omitted.

In both simulation studies, the Elastic Net and Lasso models gave similar performances. Both methods, on average, select one more correct variable than the other models, as can be seen in Figure 4a. However, Figure 4b shows that both models on average select a larger number of variables even for low correlation among variables. The number of selected variables is more than double the number of truly significant ones. Yet so, the number of the selected variables is nearly constant for increasing levels of correlation. This is further supported by Figure 4c, which shows that the ratio between variables correctly selected and the total number of selected variables related to the correlation level generally remains the same. The number of wrongly selected variables shown in Figure 4d, is stable for increasing correlation levels and this trend is also reflected in the ratio between wrongly selected and all selected variables as shown in Figure 4e.



(a) Correctly selected variables.

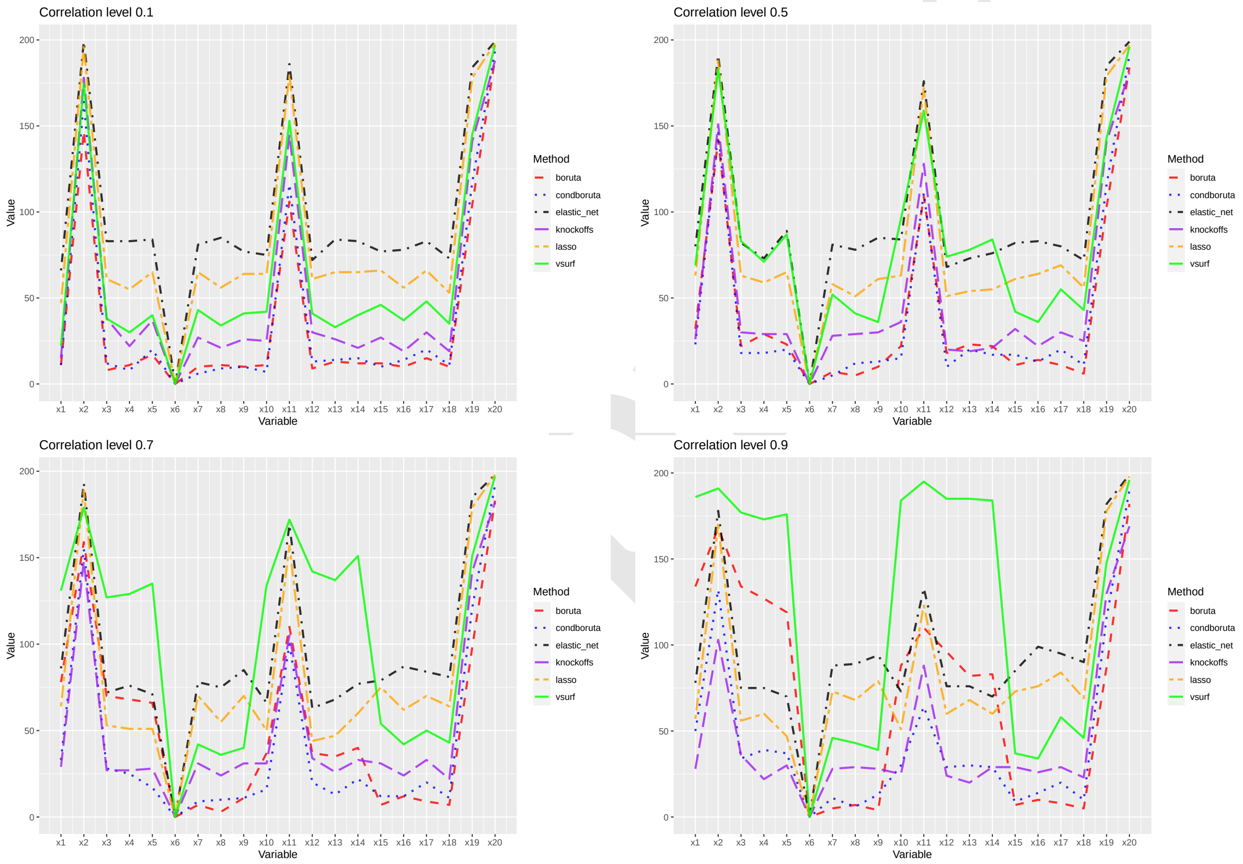
(b) All selected variables.

(c) Ratio between the correct selected and all selected variables.

(d) Falsely selected variables.

(e) Ratio between falsely and all selected variables.

Figure 4: (We present the results from only one of the simulated data sets for illustration.) (*a*): Number of correctly selected variables by six different methods vs the correlation among the variables. (*b*): Number of all selected variables vs the correlation among the variables. (*c*): The ratio between the correctly selected and all selected variables vs the correlation among the variables. (*d*): Number of falsely selected variables vs the correlation among the variables. (*e*): The ratio between falsely selected and all selected variables vs the correlation among the variables.



Correlation level 0.1

Correlation level 0.5

Correlation level 0.7

Correlation level 0.9

Figure 5: (We present the results from only one of the simulated data sets for illustration.) The frequency of the variables being selected in 200 runs for correlation levels: $cor = 0.1, 0.5, 0.7, 0.9$.

The Knockoffs variable selection method results are also presented in Figure 4. One of the key parameters of the model is the target false discovery rate (FDR), with a default value of 0.1. However, with this setting, only 0.5% of the models selected any variables, meaning that 99.5% of the models were empty and had selected 0 variables. Therefore, we had to increase the value of the FDR to 0.5 to ensure that at least one variable was selected in at least 90% of the models. We observed that the number of models with no terms increases proportionally with the correlation levels. In other words, only a few models contained no terms for low correlation levels, however for high correlation levels, an increasing number of models ended up with no terms. Therefore we conclude that, as the correlation between variables increases, the method may become less effective in identifying relevant variables and may result in an increasing number of cases with no terms being selected. Figure 4b shows that the Knockoffs model selects a constant number of variables. This is also reflected in the ratio between correctly selected and all selected variables shown in Figure 4c, as well as the ratio between wrongly selected and all selected shown in Figure 4e. These ratios remain relatively constant but at a lower level than for the proposed model.

In Figure 4 we also present the results for the VSURF model. We notice that as the correlation increases, the number of selected variables rises drastically, negatively impacting the ratio in Figure 4c. As correlation increases, the number of incorrectly selected variables also increases, indicating a growing number of noise variables being selected. In Figure 4e, we can notice that more than half of the variables selected are noise variables.

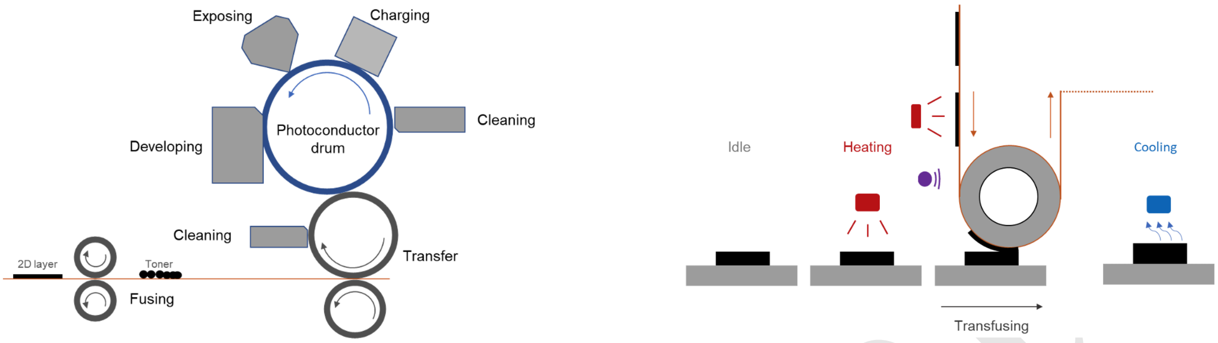
We also observe a difference between the original Boruta algorithm and the proposed extension. On average, the extended Boruta algorithm correctly selects a greater number of significant variables. We can also see that as the correlation increases, the total number of variables selected by the original Boruta grows quickly. Instead, even in the case of high correlation, the proposed extension maintains a constant number of selected variables as shown in Figure 4b. This is reflected, also, in the ratio between correctly selected variables and all selected variables. The conditional Boruta selects a large number of significant variables while keeping the total number of variables selected at a low level. This is also supported by the ratio of correctly chosen variables to the total number of variables selected, shown in Figure 4c. Moreover, the proposed model exhibits a very low number of wrongly selected variables in Figure 4d, in fact the lowest among all the models studied in this work. The ratio between wrongly selected and all selected variables is also the lowest among all models. These findings suggest that the selected variables accurately reflect the variables used to construct Y, even in challenging scenarios with high correlation among variables for which other models tend to over select variables.

In Figure 5, we present the frequency of each variable being selected by different models. The input variables are displayed on the x-axis of each figure, and the y-axis represents the frequency with which each variable was selected by the algorithms across 200 iterations. We can see that all models, on average, pick the relevant variables correctly. However, even at low levels of correlation among variables, Lasso, Elastic Net and VSURF also select noise variables as significant. As the correlation increases, the original Boruta algorithm selects the entire group of correlated variables. In general, we also notice a difference in the frequency of selection of the two correlated variables $X2$ and $X11$. The variable $X11$ is selected less frequently than the variable $X2$, this is due to a lower coefficient used in (9). The same happens for the uncorrelated variables $X19$ and $X20$, for which the former is not selected in all models having a lower coefficient.

The Boruta algorithm, at each iteration, stores the variable importance value for the original variables and averages them at the end of the algorithm in a final table. The extension we propose in this paper gives us a more accurate selection of variables and a more representative final table. Further analysis could be carried out by exploiting the ranking of the variable's importance values in the final table, particularly the variables close to the Boruta threshold. This possibility can be pursued if Boruta's outcome is not entirely satisfactory. We can increase the number of variables or, on the contrary, be more restrictive in our decision by utilizing subject matter knowledge.

3.2. Application to Additive manufacturing case

In this section, we demonstrate the use of the proposed method for variable selection in an actual production data. The data is acquired from a brand new additive manufacturing equipment for high volume 3-D printing. The Selective Thermoplastic Electrophotographic Process (STEP) [49] is a breakthrough approach in additive manufacturing, that offers a very flexible option for complex geometries and various aspects of colouring. This new technology works by fusing and pressing super-thin, nearly two-dimensional layers produced by electrophotography into a single 3D bulk structure. With its two fundamental modules, electrophotographic and transfusion, Figure 6, this technology is able to produce a completely dense, multi-material, and multi-coloured components[50]. Multiple sensors are positioned throughout the production chain on the new manufacturing line. Examples of measured quantities include the amount of material used for each layer, the temperature before and after the melting process.



(a) Electrophotographic module.

(b) Transfusion module.

Figure 6: Additive manufacturing process: Selective Thermoplastic Electrophotographic Process (STEP). Adaptation from [49] of the Electrophotographic and Transfusion modules.

The main challenge is that the printing process is not very well understood and the goal is to identify key process variables that are related to product quality through data-driven approaches. The case study involves 18 different batches of production. As the input data, we considered 53 continuous variables collected through the sensors located throughout the printing machine. As the response variable, we consider Young's modulus as the physical quality aspect of the final products. Young's modulus is a fundamental concept in materials science that measures the stiffness of the material, and it is defined as the ratio of stress to strain in a material subjected to tensile or compressing forces. The input variables are labelled from $V1$ to $V53$ due to confidentiality.

Figure 7 displays the correlation among the production process variables, revealing differing degrees of correlation between the variables. Nonetheless we expect that particularly the high correlation, will likely cause certain variables' importance scores to be overestimated. This, in turn, could lead to the inaccurate ranking of variables and, therefore, to the selection of irrelevant variables for further studies.

All the models previously discussed in this paper were applied to the manufacturing data, showing similar prediction performances. Figure 8 shows the resulting variable importance rankings. The parameter estimates in Lasso and Elastic Net models are determined using five-fold cross-validation. We can see that these two models result in very similar rankings of the variables. Both models select most of the variables, 50 out of 53 variables, to be relevant. The original Boruta algorithm selects most of the process variables, 51 out of 53, as relevant. The VSURF model selected 19 out of 53 variables, indicating a more conservative approach when compared to the first three models. Nonetheless, the selection of a relatively high number of variables is still noteworthy. The Knockoffs variable selection model provides only the selected variables. The outcome of this selection method shows $V3, V7, V10, V39, V40, V41, V45, V49, V50$, and $V51$ as the selected variables. This model presented a more restrictive selection compared to the aforementioned methods, and the selected variables align with the ones highly ranked and selected from the previous methods. The proposed model, on the other hand, is the only one that selects a significantly smaller number of variables, i.e., 3 variables ($V16$, $V43$ and $V50$) out of 53 are selected as relevant. These variables correspond to distinct stages of the process. $V16$ is associated with the way the layers overlap. In fact, if the layers do not overlap properly, the final product's quality and, in particular, the physical characteristics may be compromised. $V43$ and $V50$ are connected to the layer positioning belt, which is subject to high degradation. The first, $V43$ is connected to the electrophotography module Figure 6a and it is related to the transfer of the image to the belt. Variable $V50$ tracks the number of hours the belt has been in operation. The engineers have observed that it becomes defiled after a specific number of production hours. Thus, additional investigation will determine the optimal number of hours to replace this component. With the engineers' approval, these variables are selected to be investigated further.

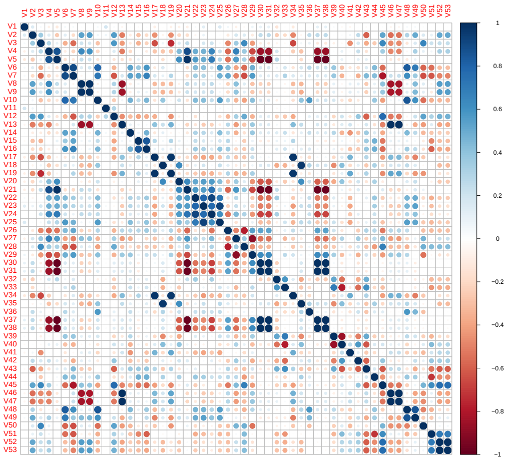
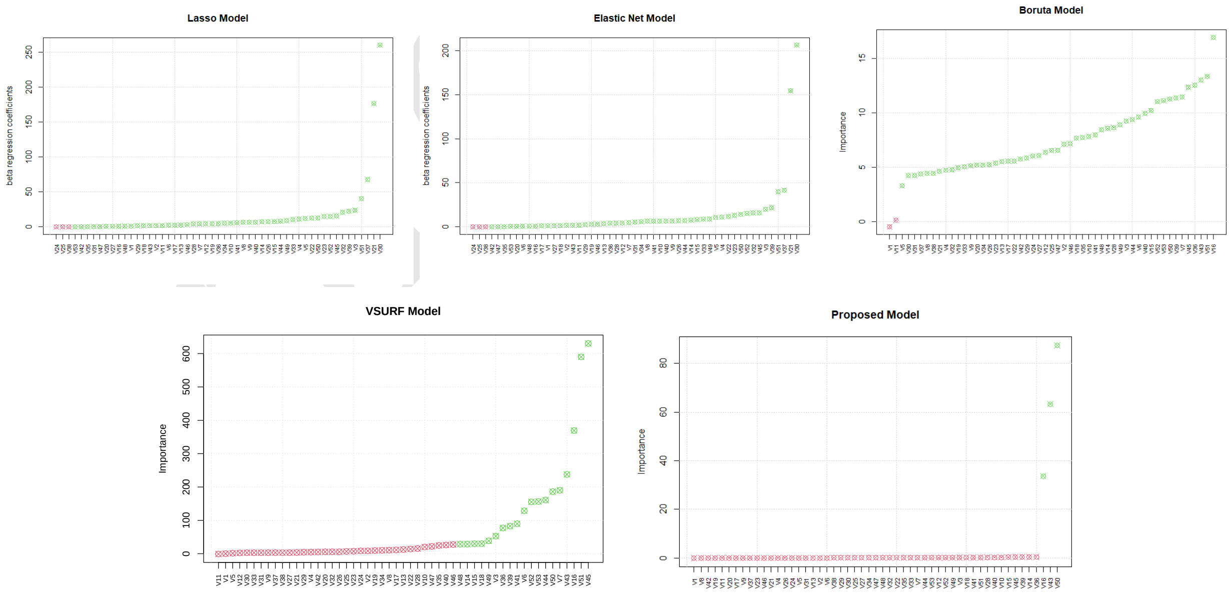


Figure 7: Additive manufacturing real-world application. Correlation among the input variables.


Lasso Model

Elastic Net Model

Boruta Model

VSURF Model

Proposed Model

Figure 8: Lasso model, Elastic Net, Boruta, VSURF and the proposed model applied to the additive manufacturing real-world application.

4. Conclusion

In this study, we present an extension of the original Boruta algorithm for the case of high correlation among input variables. This extension makes use of the conditional variable importance measure, which is a more sensitive measure in the case of highly correlated variables. To evaluate the performance of the proposed extension, two simulation studies and a real-world case are presented. The results of the proposed extension are compared against other variable selection approaches, including the original Boruta algorithm, Lasso and Elastic Net regression, Variable Selection Using Random Forest (VSURF), and the Knockoffs variable selection method. Our findings indicate that the proposed extension outperforms these other methods in terms of identifying the most relevant variables while minimizing the number of wrongly selected variables, particularly when the correlation among variables is high and in the case of variable interaction. Moreover, the extended method also exhibits superior performance in terms of the ratio of correctly selected variables to the total number of selected variables. In the industrial case study, the proposed model selects fewer variables than other models that select most of the input variables. We believe that this approach can be used in many applications, as it provides greater transparency and understanding of the process.

References

[1] H. E. Kiziloz, A. Deniz, An evolutionary parallel multiobjective feature selection framework, Computers & Industrial Engineering 159 (2021) 107481.

[2] A. Shinde, G. Church, M. Janakiram, G. Runger, Feature extraction and classification models for high-dimensional profile data, Quality and Reliability Engineering International 27 (7) (2011) 885–893.

[3] V. Atamuradov, K. Medjaher, F. Camci, N. Zerhouni, P. Dersin, B. Lamoureux, Feature selection and fault-severity classification–based machine health assessment methodology for point machine sliding-chair degradation, Quality and Reliability Engineering International 35 (4) (2019) 1081–1099.

[4] R. Kohavi, G. H. John, Wrappers for feature subset selection, Artificial intelligence 97 (1-2) (1997) 273–324.

[5] R. Nilsson, J. M. Pena, J. Björkegren, J. Tegnér, Consistent feature selection for pattern recognition in polynomial time, The Journal of Machine Learning Research 8 (2007) 589–612.

[6] A. Detzner, M. Eigner, Feature selection methods for root-cause analysis among top-level product attributes, Quality and Reliability Engineering International 37 (1) (2021) 335–351.

[7] L. Breiman, Random forests, Machine learning 45 (1) (2001) 5–32.

[8] M. Segal, Y. Xiao, Multivariate random forests, Wiley interdisciplinary reviews: Data mining and knowledge discovery 1 (1) (2011) 80–87.

[9] H. Ishwaran, U. B. Kogalur, E. H. Blackstone, M. S. Lauer, Random survival forests, The annals of applied statistics 2 (3) (2008) 841–860.

[10] D. Conn, T. Ngun, G. Li, C. M. Ramirez, Fuzzy forests: Extending random forest feature selection for correlated, high-dimensional data, Journal of Statistical Software 91 (2019) 1–25.

[11] M. Zhu, J. Xia, X. Jin, M. Yan, G. Cai, J. Yan, G. Ning, Class weights random forest algorithm for processing class imbalanced medical data, IEEE Access 6 (2018) 4641–4652.

[12] F. Tang, H. Ishwaran, Random forest missing data algorithms, Statistical Analysis and Data Mining: The ASA Data Science Journal 10 (6) (2017) 363–377.

[13] H. Ishwaran, Variable importance in binary regression trees and forests, Electronic Journal of Statistics 1 (2007) 519–537.

[14] C. Bénard, S. da Veiga, E. Scornet, Mda for random forests: inconsistency, and a practical solution via the sobol-mda, Biometrika 109 (2022) 881–900.

[15] H. Ishwaran, M. Lu, Standard errors and confidence intervals for variable importance in random forest regression, classification, and survival, Statistics in medicine 38 (4) (2019) 558–582.

[16] R. Zhu, D. Zeng, M. R. Kosorok, Reinforcement learning trees, Journal of the American Statistical Association 110 (512) (2015) 1770–1784.

[17] H. Stoppiglia, G. Dreyfus, R. Dubois, Y. Oussar, Ranking a random feature for variable and feature selection, The Journal of Machine Learning Research 3 (2003) 1399–1414.

[18] C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, A. Zeileis, Conditional variable importance for random forests, BMC bioinformatics 9 (1) (2008) 1–11.

[19] K. J. Archer, R. V. Kimes, Empirical characterization of random forest variable importance measures, Computational statistics & data analysis 52 (4) (2008) 2249–2260.

[20] K. K. Nicodemus, J. D. Malley, Predictor correlation impacts machine learning algorithms: implications for genomic studies, Bioinformatics 25 (15) (2009) 1884–1890.

[21] L. Auret, C. Aldrich, Empirical comparison of tree ensemble variable importance measures, Chemometrics and Intelligent Laboratory Systems 105 (2) (2011) 157–170.

[22] K. K. Nicodemus, J. D. Malley, C. Strobl, A. Ziegler, The behaviour of random forest permutation-based variable importance measures under predictor correlation, BMC bioinformatics 11 (1) (2010) 1–13.

[23] L. Toloşi, T. Lengauer, Classification with correlated features: unreliability of feature ranking and solutions, Bioinformatics 27 (14) (2011) 1986–1994.

[24] R. Genuer, J.-M. Poggi, C. Tuleau-Malot, Variable selection using random forests, Pattern recognition letters 31 (14) (2010) 2225–2236.

[25] B. Gregorutti, B. Michel, P. Saint-Pierre, Correlation and variable importance in random forests, Statistics and Computing 27 (3) (2017) 659–678.

[26] C. Strobl, A.-L. Boulesteix, T. Augustin, Unbiased split selection for classification trees based on the gini index, Computational Statistics & Data Analysis 52 (1) (2007) 483–501.

[27] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, Journal of machine learning research 3 (Mar) (2003) 1157–1182.

[28] V. Svetnik, A. Liaw, C. Tong, T. Wang, Application of breiman's random forest to modeling structure-activity relationships of pharmaceutical molecules, in: International Workshop on Multiple Classifier Systems, Springer, 2004, pp. 334–343.

[29] N. Louw, S. Steel, Variable selection in kernel fisher discriminant analysis by means of recursive feature elimination, Computational Statistics & Data Analysis 51 (3) (2006) 2043–2055.

[30] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Machine learning 46 (1) (2002) 389–422.

[31] A. Rakotomamonjy, Variable selection using svm-based criteria, Journal of machine learning research 3 (Mar) (2003) 1357–1370.

[32] R. Díaz-Uriarte, S. A. De Andres, Gene selection and classification of microarray data using random forest, BMC bioinformatics 7 (1) (2006) 1–13.

[33] S. Xia, Y. Yang, An iterative model-free feature screening procedure: Forward recursive selection, Knowledge-Based Systems 246 (2022) 108745.

[34] R. Genuer, J.-M. Poggi, C. Tuleau-Malot, Variable selection using random forests, Pattern recognition letters 31 (14) (2010) 2225–2236.

[35] M. B. Kursa, W. R. Rudnicki, Feature selection with the boruta package, Journal of statistical software 36 (2010) 1–13.

[36] E. Keany, Borutashap 1.0.16, accessed on 04 24, 2023 (2021).
URL https://pypi.org/project/BorutaShap

[37] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society: Series B (Methodological) 58 (1) (1996) 267–288.

[38] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, Journal of the royal statistical society: series B (statistical methodology) 67 (2) (2005) 301–320.

[39] E. Scornet, Trees, forests, and impurity-based variable importance, arXiv preprint arXiv:2001.04295 (2020).

[40] K. K. Nicodemus, On the stability and ranking of predictors from random forest variable importance measures, Briefings in bioinformatics 12 (4) (2011) 369–373.

[41] W. R. Rudnicki, M. Kierczak, J. Koronacki, J. Komorowski, A statistical method for determining importance of variables in an information system, in: International Conference on Rough Sets and Current Trends in Computing, Springer, 2006, pp. 557–566.

[42] U. Grömping, Variable importance assessment in regression: linear regression versus random forest, The American Statistician 63 (4) (2009) 308–319.

[43] P. Neville, Controversy of variable importance in random forests, Journal of Unified Statistical Techniques 1 (1) (2013) 15–20.

[44] D. Debeer, C. Strobl, Conditional permutation importance revisited, BMC bioinformatics 21 (1) (2020) 1–30.

[45] M. Rotari, Conditional-boruta, GitHub repository (2021).
URL https://github.com/MartaRotari/Conditional-Boruta

[46] R. F. Barber, E. J. Candès, Controlling the false discovery rate via knockoffs, The Annals of Statistics 43 (5) (2015) 2055 – 2085. doi:10.1214/15-AOS1337.
URL https://doi.org/10.1214/15-AOS1337

[47] E. J. Candès, Y. Fan, L. Janson, J. Lv, Panning for gold: Model-free knockoffs for high-dimensional controlled variable selection, Vol. 1610, Department of Statistics, Stanford University Stanford, CA, USA, 2016.

[48] R. Genuer, J.-M. Poggi, C. Tuleau-Malot, Vsurf: an r package for variable selection using random forests, The R Journal 7 (2) (2015) 19–33.

[49] H.-P. Yeh, K.Meinert, M.Bayat, J.Hattel, Part-scale thermo-mechanical modelling for the transfusion module in the selective thermoplastic electrophotographic process, in: WCCM-APCOM 2022, Volume 1000 Manufacturing and Materials Processing, 2022.

[50] T. Stichel, B. Geißler, J. Jander, T. Laumer, T. Frick, S. Roth, Electrophotographic multi-material powder deposition for additive manufacturing, Journal of Laser Applications 30 (3) (2018) 032306.