Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Simulation Framework for Evaluating Statistical Methods for Quality Control in Manufacturing

Niklas Fries (niklas.fries@umu.se)
Umeå Universitet https://orcid.org/0000-0001-6184-8951

Patrik Rydén
Umeå Universitet

Research Article

Keywords: Process Control, Manufacturing, Quality Prediction, Simulation, Variable Selection

Posted Date: November 8th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-1036908/v1

License: 🅭 🅯 This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

A Simulation Framework for Evaluating Statistical Methods for Quality Control in Manufacturing

Niklas Fries[1*] and Patrik Rydén[1]

[1]Dept. of Mathematics and Mathematical Statistics, Umeå University, Umeå, SWEDEN.

*Corresponding author(s). E-mail(s): niklas.fries@umu.se;

Abstract

Statistical methods are commonly used to monitor quality in manufacturing processes. We consider a set of problems where the probability that a unit is defect, i.e., the defect probability, is influenced by a large number of sub-processes, and where the overall aim is to monitor the defect probability and link abnormalities in quality to observed production variables. We developed a simulation framework for studying the performances of statistical methods used to solve the considered problem. Fourteen prediction procedures were obtained by combining six prediction methods (linear regression, logistic regression, LASSO, penalized logistic regression, support vector machines, gradient boosting decision trees) and two pre-processing procedures. These prediction procedures were evaluated on four types of simulated datasets with different relationships between the explanatory variables and the defect probabilities. Additionally, two established methods for variable selection were compared to a novel method called mixed moments selection (MMS). MMS was more robust than the other methods, performed well on all dataset types, and can easily be combined with any type of prediction method. Additionally, it was shown that it can be advantageous to complement the original explanatory variables with their squared values prior to analyzing the data. Overall, a procedure combining MMS, including additional quadratic terms and using PLR had the best performance. The proposed framework can be applied to evaluate any type of prediction procedure for the general problem we consider. This would increase the understanding of different procedures and facilitate the selection of procedures for a specific problem.

Keywords: Process Control, Manufacturing, Quality Prediction, Simulation, Variable Selection

1 Introduction

Statistical learning and machine learning methods are commonly used to analyze high-dimensional datasets from manufacturing processes. A common objective is to use process data as explanatory variables to model the quality of the product, either as a regression or classification problem [5, 12, 19, 13, 11]. In recent years a large number of algorithms that predict this quality with a high accuracy has been developed, including support vector machines [6], tree-based algorithms such as random forests [2] and gradient boosting [8], as well as numerous flavors of neural networks [15]. These algorithms share the property that they are *complex* in the way the predictions are calculated, in the sense that it is hard to understand how the individual explanatory variables contribute to the predictions. Besides prediction, statistical analysis of these manufacturing datasets is used for *interpretation*, i.e., to associate the quality with individual process variables [1]. It is generally accepted that the predictive power of the aforementioned complex methods come with a tradeoff that is lower interpretability [10, Chp. 2.1.3]. This tradeoff has been addressed before, perhaps most interestingly by model-independent methods examplified by: LIME [18] learns an interpretable model locally; Štrumbelj and Kononenko [22] suggested perturbing all input features to account for interactions; Shapley values [14] use game theory to reduce the impact of collinearity among the regressors; DeepLIFT [21] attempts to decompose the predictions of neural networks. If a complex model has a good model fit, these methods may help to provide a useful interpretation. However, if the fit is poor there will be little information on why the model failed to describe the data.

Model fitting is usually preceeded by some sort of preprocessing such as feature selection, transformation, and extraction [5, 16]. One purpose of this preprocessing is to make the model fitting easier, e.g., by reducing the number of parameters to estimate. However, this preprocessing may exacerbate the problem of understanding the predictions of the fitted model. Feature extraction methods such as principal component analysis create new features from a high number of the original explanatory variables, but these features will have loadings on every single original variable. Penalized matrix decomposition methods serve to find a sweeter spot in the prediction/interpretation tradeoff, but they do not eliminate the tradeoff entirely [24]. Kernel methods make this even worse by making these loadings vary in the parameter space [20]. In conclusion, even though the aforementioned methods have proven to be quite powerful, there is still a need for simple and robust algorithms that make both fitting and interpretation easier.

If the purpose is to model the probability of a quality problem with a binary response as a function of a set of process variables, the evaluation may be difficult. In particular, it can be difficult to conclude whether a bad fit is due to an improper model or due to a low explanatory power of the data. If the perfect model is fitted but the data explain a small part of the probability, the model will seem to have poor performance. On the other hand, if the data explain the entire probability but the model is a poor fit, the model will seem to be just as in the previous case. In this study we attempt to bridge this gap by simulating the process data, and specifying a function that computes the probability of a defect from the process data. The computed probabilities are used to generate an observable binary outcome, i.e., defect/not defect. Various regression and machine learning methods are used to predict the computed probabilities from the observed explanatory variables and binary outcome. The simulations are based on a visualization study from an actual manufacturing process, where the process variables were illustrated using various plots which were studied extensively. The aim of this approach is to provide an understanding of how well one can expect the statistical models to perform.

The manufacturing process which is the inspiration for this study is the Volvo Trucks cab factory in Umeå, Sweden. In this factory steel coil is stamped and welded into cabs which are surface treated and sent to other locations for vehicle assembly. We are concerned with the quality of the surface treatment, which consists of three steps: pre-treatment and electrocoating, primer application, and top coat application. Each of these steps is followed by a quality control where defects are logged and repaired. Example defects include particles, craters, and paint droplets. A large part of the repair cost is associated with the transport to and from the repair stations, which is why we categorize the cabs into defect and not defect.

This paper is organized as follows: in Section 2.1 we provide a description of the simulation framework and the simulated datasets, in Section 2.2 we describe the variable selection methods and experiment, and in Section 2.3 we describe the prediction experiment and the evaluated predictive methods. In Section 3.1 we present the results of the variable selection experiment, and in Section 3.2 we present the results of the prediction experiment.

2 Method

2.1 Simulation framework

We consider a problem with m independent stochastic explanatory variables $\mathbf{X} = (X_1, ..., X_m)$, and the corresponding observations $\mathbf{x}_i = (x_{i,1}, ..., x_{i,m})$, $i = 1, ..., n$. Furthermore, we have a vector of response variables $Y = (Y_1, ..., Y_n)$, and the corresponding observations

$y = (y_1, ..., y_n)$, where $y_i \in \{0, 1\}$, $i = 1, ..., n$. Here a y-value of 0 and 1 would represent a non-defect and defect observation, respectively. Finally, let $p = (p_1, ..., p_n)$ denote the *defect probabilities*, i.e., $p_i = P(Y_i = 1)$, $i = 1, ..., n$. Here we assume that the defect probabilities p can be expressed as a function of the observed explanatory variables, i.e., $p_i = h(\mathbf{x}_i)$, $i = 1, ..., n$.

In order to simulate y-values, we need to specify the function h. This would allow us to draw y-values from the Bernoulli distribution with parameters $p_i = h(\mathbf{x}_i)$, $i = 1, ..., n$. In the simplest case, h can be derived from the standard logit function, i.e.,

$$\text{logit}(p_i) = \ln\left(\frac{p_i}{1 - p_i}\right) = g(\mathbf{x}_i), \tag{1}$$

such that

$$p_i = h(\mathbf{x}_i) = \text{logit}^{-1}(g(\mathbf{x}_i)) = \frac{1}{1 + \exp(-g(\mathbf{x}_i))}, \tag{2}$$

where

$$g(\mathbf{x}_i) = \alpha + \sum_{j=1}^{m} \beta_j x_{i,j}, \tag{3}$$

$i = 1, ..., n$, where α and $\{\beta_j\}_{j=1,...,m}$ are parameters.

The linearity of Eqn. (3) is restrictive, and we therefore consider a more general function, i.e.,

$$g(\mathbf{x}_i) = \alpha + \sum_{j=1}^{m} \beta_j s_j(x_{i,j}), \tag{4}$$

where the *shape functions* s_j can be non-linear. For this study we consider three different shape functions denoted s_1, s_2 and s_3. First, we introduce the *raw shape functions* s'_1 to s'_3, where

$$s'_1(x) = x, \tag{5}$$

$$s'_2(x) = x^2, \tag{6}$$

$$s'_3(x) = \begin{cases} 0, & x \le 0 \\ x, & x > 0. \end{cases} \tag{7}$$

In order to facilitate comparisons between different shape functions, we want the impact of each variable to be of similar size regardless of the choice of shape function used to transform that variable. To this purpose, the raw shape functions are standardized with respect to the transformed X-variable, so that the transformed variable has mean 0 and variance 1. Thus, we get the shape functions used in the simulations by

$$s_j(x) = \frac{s'_j(x) - E[s'_j(X_j)]}{\sqrt{V[s'_j(X_j)]}}, \tag{8}$$

$j = 1, ..., m$. The standardized functions s_1, s_2 and s_3 will be referred to as the *linear*, *quadratic* and *ramp* functions, respectively. The shape functions for some distributions of X are shown in Fig. 1.

Next, we consider the β-parameters in (4), which need to be chosen such that the influence of each explanatory variable can be controlled. First, we rewrite (4) as

$$g(\mathbf{x}_i) = \alpha + C \sum_{j=1}^{m} \gamma_j s_j(x_{i,j}), \tag{9}$$

where

$$\sum_{j=1}^{m} \gamma_j^2 = 1. \tag{10}$$

Here the parameter C denotes the *overall importance* of the explanatory variables, i.e., to what degree they explain the outcome of Y. The distribution of the defect probability $P = h(\mathbf{X})$ for different values of C are shown in Fig. 2. The variances of P, as well as the expected variances of $Y \mid P$ as functions of C are shown in Fig. 3. For large values of C, the defect probabilities p are close to 0 and 1, while for small values of C they are close to the prior probability. In other words, a low value of C defines a hard prediction problem.

The γ-parameters in (9) control the *relative importance* of the explanatory variables. For real world applications, we believe that it is reasonable to assume that only a few variables have a large influence on the outcome. Therefore we chose the γ-parameters as described in Fig. 4, where 60% of the explanatory variables were non-informative, and for $m = 1000$ the remaining variables ranged from having a small impact ($\gamma = 0.026$) to a large impact ($\gamma = 0.18$).

If the distribution of $\mathbf{X}$ is specified, the explanatory variables can be simulated. If in addition the parameters in (9) are specified, i.e., the intercept α, the overall importance C, the shape functions s, and the relative importances γ, the

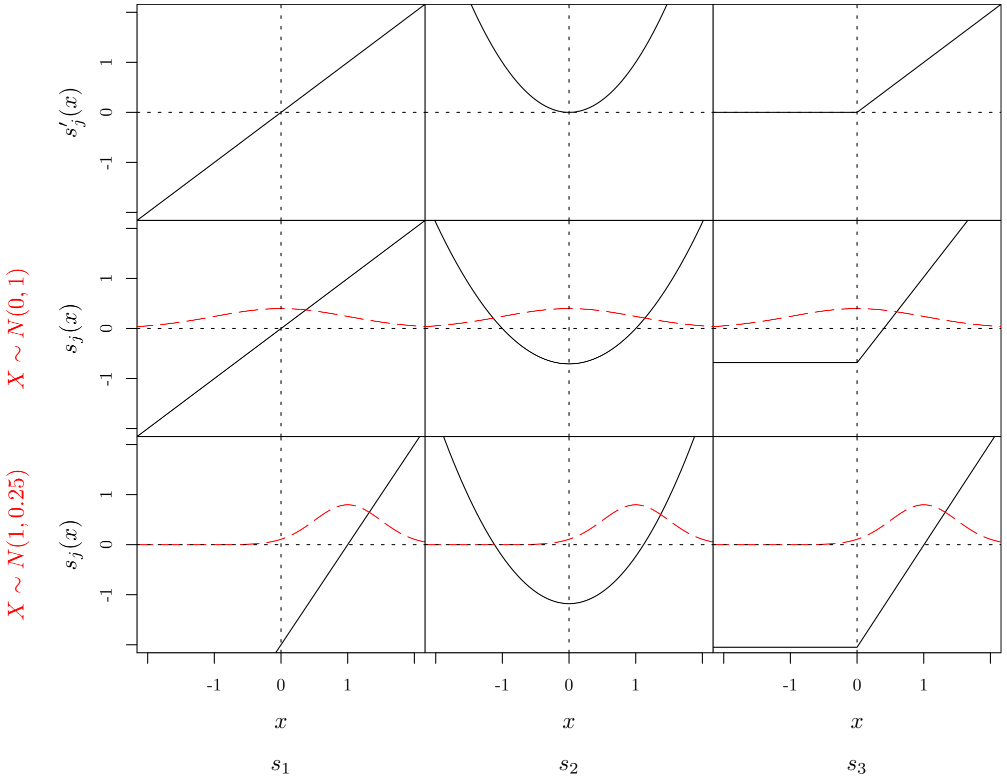

Fig. 1: The raw shape functions $s'_1(x)$ to $s'_3(x)$ (first row); the shape functions $s_1(x)$ to $s_3(x)$ where $X \sim N(0,1)$ (second row) and where $X \sim N(1,0.25)$ (third row), with the shape functions standardized so that the transformed variables have mean value 0 and variance 1. The distributions $N(0,1)$ and $N(1,0.25)$ are denoted with red lines in the second and third rows, respectively. Created using the TikZDevice R package.

defect probabilities p can be calculated, and the binary outcomes y can be simulated. For all simulations the following settings were fixed: $m = 1000$, $X_j \sim N(0,1)$, $j = 1,...,m$, $\alpha = 0$, $C = 2$, and the γ-parameters as described in Fig. 4. The number of observations n was varied with each experiment.

By varying the distribution of the shape functions, four *dataset types* were simulated, denoted the *linear*, *quadratic*, *ramp*, and *mixed* datasets. For the first three types, the shape functions were the linear, quadratic, and ramp shape functions, respectively. For the fourth type, the shape function for each x-variable was drawn from the linear, ramp, and quadratic shape functions with equal probability. For the experiments each dataset type was realized several times, and each such realization will be referred to as a *dataset realization*.

2.2 Variable selection

When analyzing the datasets in Section 2.1, variable selection was performed prior to the prediction of the defect probabilities. The selection methods considered were the *mean value selection* method (MVS), *variance selection* method (VS), and the novel *mixed moments selection* method (MMS).

MVS uses the two-sided Welch's t-test [23] to test the null hypotheses

$$H_{0,j} : E[X_j \mid y_i = 0] = E[X_j \mid y_i = 1], \qquad (11)$$

$i = 1,...,n$, $j = 1,...,m$. x-variables with p-values lower than the significance level α were included in the downstream analysis. VS uses Levene's test of equal variances [3, pp. 278–292] to test the null

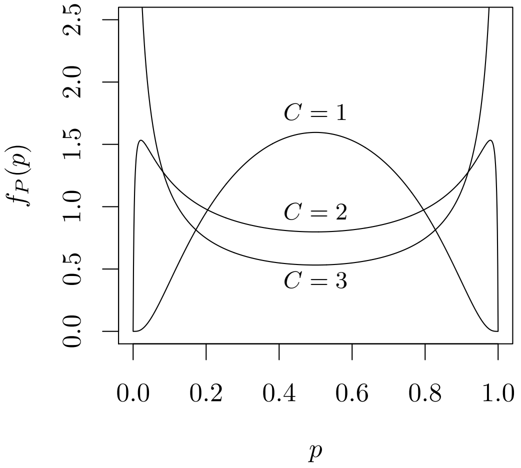

Fig. 2: The distributions of the defect probability $P = h(\mathbf{X})$ for different values of the overall importance C. The distribution of defect probability converges to these distributions for any set of shape functions when the number of explanatory variables increases, under the standard assumptions for the central limit theorem. Created using the TikZDevice R package.

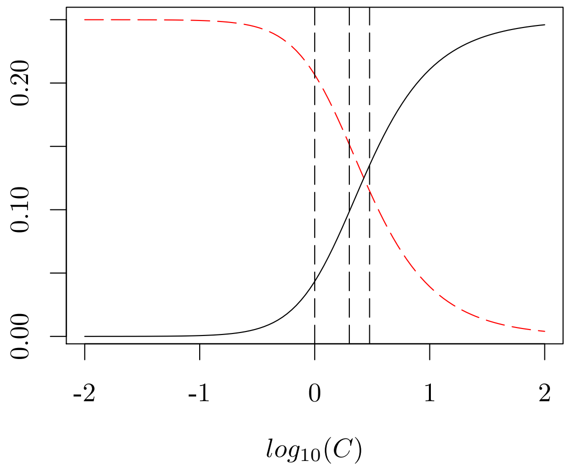

Fig. 3: The variance of the defect probability P as a function of the overall importance C (black solid line), and the corresponding expected variance of the conditional probability $Y \mid P$ (red dashed line). The vertical dashed lines for $C \in \{1, 2, 3\}$ correspond to the distributions in Fig. 2. Created using the TikZDevice R package.

hypotheses

$$H_{0,j} : V[X_j \mid y_i = 0] = V[X_j \mid y_i = 1], \qquad (12)$$

$i = 1, ..., n$, $j = 1, ..., m$, and selects variables with p-values lower than α.

The novel MMS method uses the lower of the p-values from the MVS and VS methods and selects the variable if this p-value is lower than the significance level. This method uses the Bonferroni corrected [9] significance level α' to preserve the false positive rate:

$$\alpha' = 1 - \sqrt{1 - \alpha}, \qquad (13)$$

Here, the results of the mean value and variance tests are assumed to be independent.

The variable selection methods MVS, VS and MMS were evaluated on four dataset types: linear, quadratic, ramp and mixed, each with 500, 5 000, 50 000 observations, see Section 2.1. For each dataset type, the methods were evaluated on 100 dataset realizations. For MVS and VS a significance level of $\alpha = 0.1$ was used, and the corresponding significance level for MMS was $\alpha' = 1 - \sqrt{1 - \alpha} \approx 0.0513$. Henceforth, only the unadjusted significance level α will be used in the text.

Note that when predicting defect probabilities, the true probabilities p cannot be observed, nor can the root mean square error (RMSE), i.e.,

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\hat{p}_i - p_i)^2}, \qquad (14)$$

where $\hat{p}$ and p are the predicted and true defect probabilities, respectively. However, in Fig. 9 we showed that the Y-RMSE, i.e.,

$$\text{Y-RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\hat{p}_i - y_i)^2} \qquad (15)$$

can be a useful proxy, where y is the observable binary outcome. Additionally, we showed that $\alpha = 0.1$ is an appropriate level for our simulations. Also note that a lower α will result in more informative variables being excluded, while a higher α will result in more uninformative variables being included.

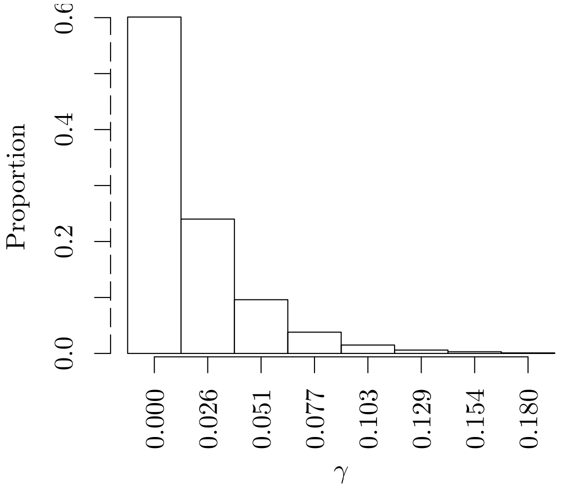

Fig. 4: The distribution of the relative importances γ used in the simulations. Created using the TikZDevice R package.

For each number of observations, dataset type, and selection method, a vector of the total relative importance of the selected variables was computed, with one value for each dataset realization. For each number of observations, dataset type, and pair of selection methods, the differences between these vectors was computed. One-sample t-tests were performed to test the null hypotheses that these differences had expected value 0, against two-sided alternative hypotheses, using a 5% significance level. The results of these tests are presented in Section 3.1.

2.3 Prediction

For each of the dataset types described in Section 2.1, the simulated explanatory variables $\mathbf{x}_i$ and the binary outcomes y_i were used to predict the defect probabilities p_i, $i = 1, ..., n$, $j = 1, ..., m$, where $m = 1000$. Each dataset type was realized 10 times each, and each dataset realization consisted of a training, validation, and test part, each of size $n = 50\,000$. The predictive models were adjusted to the training part, and any hyperparameters were tuned on the validation data. The prediction performance was evaluated by computing the RMSE for each dataset realization, see (14). Since the defect probabilities in (14) are not observable in real world problems, any hyperparameters were optimized with respect to the Y-RMSE, see (15).

The defect probabilities were estimated using six methods: linear regression (LinR), LASSO regression (LASSO), logistic regression (LogR), penalized logistic regression (PLR), support vector machine (SVM), and gradient boosting decision trees (GBDT). For LinR, an ordinary least squares model was adjusted to the binary y-values, and any predictions smaller than 0 or larger than 1 were set to 0 and 1, respectively. LinR is commonly not used when the outcome is a binary variable, and the predicted values cannot formally be regarded as estimates of the defect probabilities. However, we chose to include LinR as a negative control. Additionally, as the overall importance of the model decreases, the logit function can be approximated as a linear function, and the performance of LinR should be similar to or better than that of LogR. LinR was performed using the *lm* function in the R package *stats*, version 4.0.2.

LASSO solves the problem

$$\min_{\alpha,\beta} \left\{ \frac{1}{n} \sum_{i=1}^{n} (y_i - \alpha - \mathbf{x}_i'\beta)^2 + \lambda \|\beta\|_1 \right\}, \tag{16}$$

where α, $\beta = \{\beta_1, ..., \beta_m\}$ and λ are the intercept, slope coefficients, and penalty hyperparameter, respectively. Similarly to LinR, LASSO is not commonly used for predicting binary response variables, and the predicted values cannot formally be regarded as probabilities. However, it is a well-established method, and is therefore included.

PLR solves the problem

$$\max_{\alpha,\beta} \; l(\alpha, \beta \mid X) - \lambda \|\beta\|_{l_1}, \tag{17}$$

where $l(\alpha, \beta \mid X)$ is the log-likelihood function

$$\begin{aligned} l(\alpha, \beta \mid X) = \; & \frac{1}{n} \sum_{i=1}^{n} y_i(\alpha + \mathbf{x}_i^T \beta) \\ & - \log\left\{1 + \exp(\alpha + \mathbf{x}_i^T \beta)\right\}, \end{aligned} \tag{18}$$

$$X = \begin{pmatrix} x_{1,1} & \dots & x_{1,m} \\ \vdots & & \vdots \\ x_{n,1} & \dots & x_{n,m} \end{pmatrix}, \tag{19}$$

and α, $\beta = \{\beta_1, ..., \beta_m\}$ and λ are the intercept, slope coefficients, and penalty hyperparameter, respectively. PLR is equivalent to LogR if $\lambda = 0$ [7]. LASSO and PLR were performed using the *glmnet* function in the R package *glmnet*, version 4.0.2. Both methods were used with the default parameters and the built-in hyperparameter tuning. For LogR we used the function *glm* in the R package *stats*, version 4.0.2.

SVMs use the kernel trick to map the observations into a higher dimensional space with the aim of making them linearly separable. In this kernel space a maximum margin classifier with the slackness parameter C is used. For posterior probabilities a logistic regression model is adjusted in the kernel space [17]. We used the implementation in the R package *kernlab* 0.9.29, with the self-tuning radial basis function kernel [4].

GBDT adjusts a set of regression trees, where each tree is adjusted to compensate for the error of the preceeding trees. GBDT uses gradient descent and works for an arbitrary differentiable loss function, in this case the mean squared error. The hyperparameters include the number of trees as well as the learning rate, where a higher learning rate leads to a faster overfitting, and thus a lower optimal number of trees [8]. We used the implementation in the R package *gbm* version 2.1.8.

LASSO and PLR perform their own variable selection, and there is no rationale for performing separate variable selection prior to the model fitting. However, for LinR and LogR prediction was done both with and without preceeding variable selection using the MMS method. To reduce the computation time, SVM and GBDT were only evaluated using MMS. For MMS a significance level of $\alpha = 0.1$ was used, see Section 2.2.

Linear and generalized linear models such as LinR, LASSO, LogR and PLR are not suited for the prediction of non-monotone responses. On the other hand, SVM and GBDT are designed to be robust for this type of modelling. Thus, for LinR, LASSO, LogR and PLR, these methods were evaluated only for the simulated x-variables, as well as for both these variables and their squares. That is, for the latter case the matrix X of explanatory variables was replaced by the matrix

$$X' = \begin{pmatrix} x_{1,1} & \dots & x_{1,m} & x^2_{1,1} & \dots & x^2_{1,m} \\ \vdots & & \vdots & \vdots & & \vdots \\ x_{n,1} & \dots & x_{n,m} & x^2_{n,1} & \dots & x^2_{n,m} \end{pmatrix}. \tag{20}$$

Here, the x^2-variables are treated in the same way as the x-variables during prediction, and as a result we consider a dataset with $2m$ explanatory variables. The addition of the squares of the explanatory variables was done after variable selection. In total, the six prediction methods in combination with the aforementioned preprocessing methods resulted in 14 *prediction procedures*, see Table 1.

Table 1: The prediction methods considered in the evaluation are shown in the first column. A plus and minus sign in the second column indicates that the method was used with or without variable selection using the MMS method, respectively. A plus and minus sign in the third column indicates that the method was used with and without squared explanatory variables, respectively. The last column summarizes the number of prediction procedures per method.

Method	MMS	x^2	
LinR	+-	+-	4
LASSO	-	+-	2
LogR	+-	+-	4
PLR	-	+-	2
SVM	+	-	1
GBDT	+	-	1
Total			14

All predictions were compared to a negative and a positive control. The negative control used the mean of the binary outcomes as the predicted defect probabilities, i.e.,

$$\hat{p}_i = \frac{1}{n} \sum_{k=1}^{n} y_k, \tag{21}$$

$i = 1, ..., n$. This predictor gets worse the larger the variance of the true defect probabilities.

As a positive control, the defect probabilities p_i were estimated by

$$\text{logit}(\hat{p}_i) = \hat{\alpha} + \sum_{\{j\,:\,\gamma_j > 0\}} \hat{\beta}_j s_j(x_{i,j}), \tag{22}$$

$i = 1, ..., n$, where $\gamma = (\gamma_1, ..., \gamma_m)$ and $s_j(\cdot)$ are the coefficients and shape functions used to compute the true defect probabilities, respectively. Here the $\hat{\alpha}$ and $\hat{\beta}$-estimates were computed using maximum likelihood estimation. This means that the true shape functions were known, and that only the variables with non-zero importance were selected.

Next, we investigated whether there were differences in performance between prediction procedures. To compare two procedures, both procedures were used to predict the defect probabilities for each dataset realization, and the corresponding RMSE-values were calculated. A two-sided t-test was performed on the pairwise RMSE-differences with a zero difference null hypothesis and a significance level of 5%.

When evaluating the relative difference in performance between two prediction procedures, the difference in RMSE was divided by the RMSE of the reference procedure. This was computed separately for each dataset realization, and the results were averaged. That is, the relative change (RC) in RMSE when changing from procedure a to procedure b was

$$\mathrm{RC}_{a,b} = \frac{1}{10}\sum_{r=1}^{10}\frac{\mathrm{RMSE}_{r,b} - \mathrm{RMSE}_{r,a}}{\mathrm{RMSE}_{r,a}}, \qquad (23)$$

where $\mathrm{RMSE}_{r,a}$ and $\mathrm{RMSE}_{r,b}$ is the RMSE of procedure a and procedure b for dataset realization r, respectively. The results of the prediction experiments are presented in Section 3.2. An overview of the simulation, prediction and evaluation framework is shown in Fig. 5.

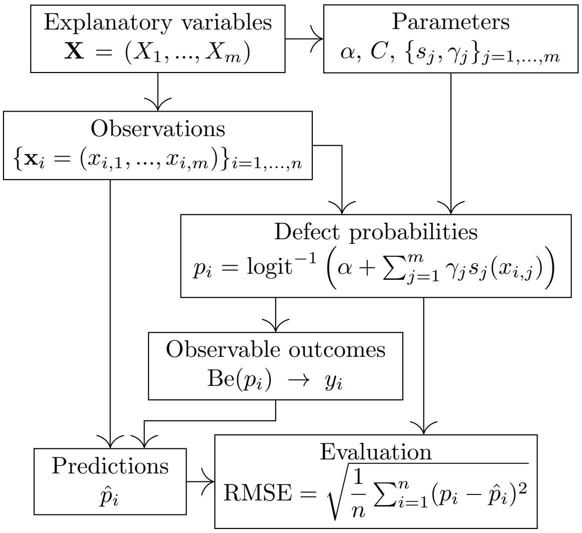
Explanatory variables
$\mathbf{X} = (X_1, ..., X_m)$
Parameters
$\alpha,\ C,\ \{s_j, \gamma_j\}_{j=1,...,m}$
Observations
$\{\mathbf{x}_i = (x_{i,1}, ..., x_{i,m})\}_{i=1,...,n}$
Defect probabilities
$p_i = \mathrm{logit}^{-1}\left(\alpha + \sum_{j=1}^{m}\gamma_j s_j(x_{i,j})\right)$
Observable outcomes
$\mathrm{Be}(p_i) \ \rightarrow \ y_i$
Predictions
$\hat{p}_i$
Evaluation
$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - \hat{p}_i)^2}$

Fig. 5: The framework for the dataset simulation, prediction and evaluation. The specified stochastic X-variables were used to simulate explanatory x-variables. The defect probabilities $p = (p_1, ..., p_n)$ were computed from the simulated x-values and the parameters, i.e., the intercept α, overall importance C, shape functions $s = (s_1, ..., s_m)$, and relative importances $\gamma = (\gamma_1, ..., \gamma_m)$. Note that the shape functions s were standardized with respect to the distributions of the X-variables, so that the transformed variables had mean value 0 and variance 1. Observable binary outcomes $y = (y_1, .., y_n)$ were simulated from Bernoulli distributions with probabilities p. The x-variables and y-values were used by the prediction procedures to compute predictions $\hat{p}$ of the defect probabilities p. The prediction procedures were evaluated using the RMSE. Created using PGF/TikZ.

3 Results

This section is organized as follows: The results of the variable selection experiment from Section 2.2 are presented in Section 3.1. The results of the prediction experiment from Section 2.3 are presented in Section 3.2.

3.1 Variable selection results

For the selection of x-variables, three variable selection methods were considered: MVS, VS and MMS, see Section 2.2. The selection methods were evaluated on four dataset types: *linear*, *quadratic*, *ramp* and *mixed*, each with 500, 5 000, 50 000 observations, see Section 2.1. For each dataset type 100 dataset realizations were simulated and analyzed with all selection methods, and their performances were measured by considering the false positive rate and the true positive rate for the non-informative and informative x-variables, respectively. Additionally, the total amount of relative importance of the selected variables was measured, i.e., the sum of the γ-values described in Section 2.1.

For all dataset types and selection methods the false positive rate was close to the significance level 0.1, and the true positive rate increased both with the number of observations, and with the relative importance of the x-variables. The exceptions were that VS and MVS completely failed to analyze the linear and quadratic datasets, respectively, see Fig. 6, Fig. 7.

For the linear and quadratic datasets, the MVS and VS methods performed the best, respectively. For both these dataset types the MMS method performed almost as well as the best method for the respective dataset type. For the ramp datasets, MVS and MMS performed the best. The performance difference between the methods was small, with a maximum difference of 1% in total selected relative importance for $n = 500$, see Fig. 6, Fig. 7.

For the mixed datasets, MMS outperformed both MVS and VS. For these datasets the true positive rate approached 1 as γ increased, and the selected importance approached 1 as n increased. These results were not observed for the other selection methods. This can be explaind by the fact that the defect probabilities in the mixed datasets were derived using a mixture of shape functions, see Fig. 6, Fig. 7. Here, all pairwise comparisons between selections methods with respect to the selected importance were significant at the 5% level, see Fig. 7.

Overall, the evaluation suggests that MMS is a robust variable selection method that is almost as good or considerably better than the commonly used MVS method, or the VS method.

3.2 Prediction results

The six prediction methods LinR, LASSO, LogR, PLR, SVM and GBDT were combined with different preprocessing methods including variable selection using the MMS method, see Section 2.2, and the addition of squared x-variables, see (20). This resulted in 14 prediction procedures, see Table 1. These procedures were used to predict the defect probabilities of 10 dataset realizations each of the linear, quadratic, ramp and mixed dataset types, see Section 2.1. The procedures were evaluated by looking at the RMSE of the predictions, see Section 2.3. The hyperparameters for SVM and GBDT for respective dataset type are shown in Table 2.

Table 2: The hyperparameter values for SVM and GBDT for the different dataset types.

Method	SVM	GBDT	
Parameter	$\log_{10}(C)$	Shrinkage	Trees
Linear	0.1	0.03	12 000
Quadratic	0.5	0.03	12 000
Ramp	0.1	0.04	7 500
Mixed	-0.1	0.04	9 000

Here, we consider only the regression methods, i.e., LinR, LASSO, LogR and PLR. As expected, adding squared x-variables when analyzing the linear datasets reduced the performance independently of which prediction procedure was used, with RMSE increasing between 11 and 44 percents, see Table 3. For the remaining dataset types, improved performances were observed when squared x-variables were included. For the quadratic, ramp and mixed dataset types the RMSE was reduced by between 70 and 79, 38 and 46, and 53 and 64 percents, respectively. Notably, for the quadratic dataset type not adding squared x-variables resulted in performances as bad as that of the negative control, see Table 3, Fig. 8. Hence, adding squared x-variables may be a good idea if nonlinear relationships are expected and if the number of observations is sufficiently large for estimating the parameters of the model.

Henceforth, we considered only the best choice with respect to adding squared x-variables, i.e., not adding squared x-variables for the linear dataset type, and adding squared x-variables for the other dataset types. Next, we investigated the effect of variable selection, i.e., we compared LinR, LinR with MMS, and LASSO, and we compared LogR, LogR with MMS, and PLR. For both LinR and LogR the performance was improved with MMS, with a reduction in RMSE of between 4 and 9 percents for LinR, and between 12 and 18 percents for LogR. Similarly, the intrinsic variable selection of LASSO and PLR reduced the RMSE by between 0 and 8 percents, and between 11 and 29 percents, respectively. Finally, we compared the MMS method to the intrinsic variable selection of LASSO and PLR. Here, LASSO was outperformed by LinR with MMS for all dataset types, with an increase in RMSE between 1 and 5 percents. In contrast, PLR outperformed LogR with MMS for the quadratic and mixed dataset types, with a reduction in RMSE of 13 and 6 percents, respectively. Notably, no significant difference in performance between LinR with MMS and PLR was observed for the linear and ramp

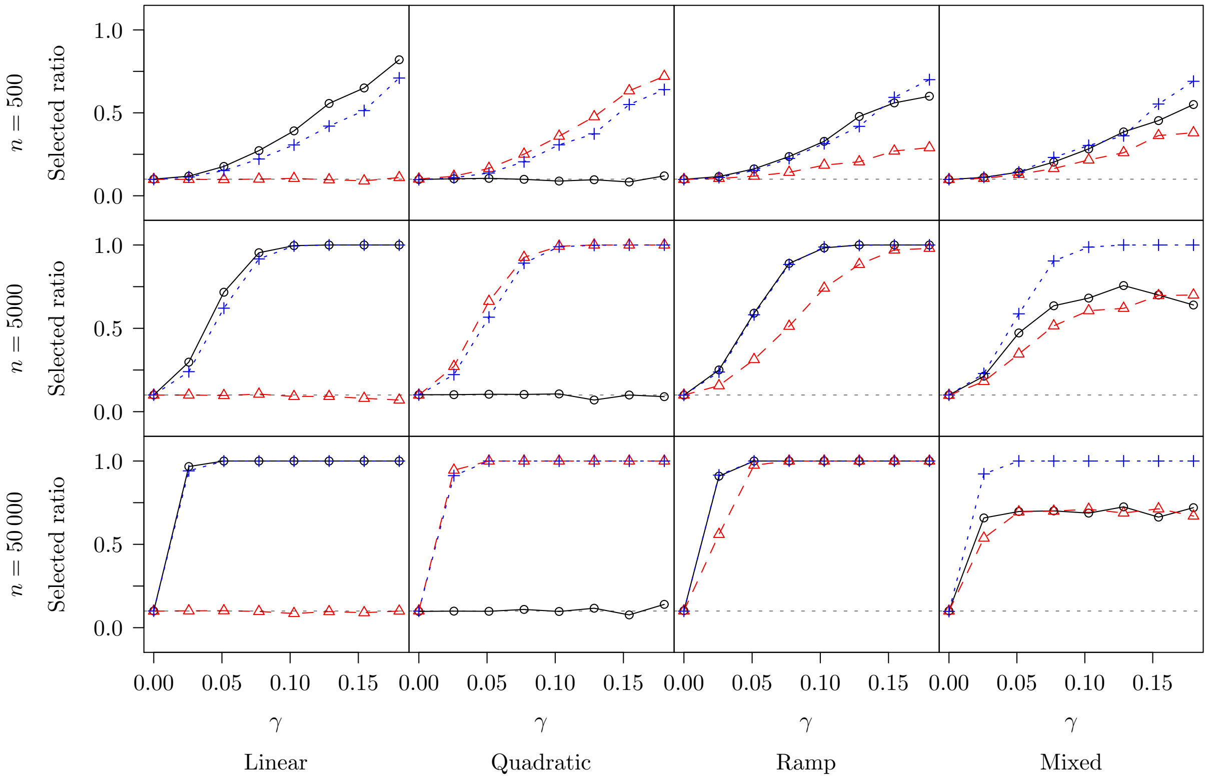

Fig. 6: The ratio of variables selected for each level of relative importance γ (horizontal axes). Each row represents a number of observations n, and each column a dataset type, from left to right: *linear*, *quadratic*, *ramp*, *mixed*. The black circles/solid lines, red triangles/long dashes and blue crosses/short dashes indicate the *mean value*, *variance*, and *mixed moments* variable selection methods, respectively. Each data point is the average over 100 repetitions. The horizontal dashed lines is the significance level used in the variable selection. Created using the TikZDevice R package.

dataset types. This is most likely due to MMS being performed before the addition of squared x-variables. This reduces the performance in case of quadratic shape functions, for which the inclusion of the original x-variables confer no advantage, see Table 3 and Fig. 8.

Next, we investigated the effect of the link function, i.e., the identity function for LinR and LASSO, and the logit function for LogR and PLR. Here, we only compared the best methods from the previous comparison, i.e., LinR-MMS and PLR. Unsurprisingly, PLR outperformed LinR with MMS for all dataset types, with a decrease in RMSE of between 9 and 31 percents. The smallest difference was observed for the ramp dataset type. The relatively high RMSE of PLR for this dataset type is probably a result of the fact that the ramp shape function is the only shape function that cannot be perfectly modeled by PLR. This suggests that LinR and LASSO are relatively robust for less well-behaved datasets, see Table 3 and Fig. 8.

Finally, we compared SVM and GBDT to the best regression method, i.e., PLR. Note that SVM and GBDT were only evaluated using MMS, which has been shown to perform worse than the intrinsic selection of PLR for the quadratic and mixed dataset types. Both SVM and GBDT performed worse than PLR for all dataset types, which is to be expected since PLR yields close to the correct model with variable selection and the correct link function. For the linear, quadratic, ramp and mixed dataset types, SVM increased the RMSE by 72, 209, 35 and 123 percents, respectively, while GBDT increased the RMSE by 149, 175, 37 and 89

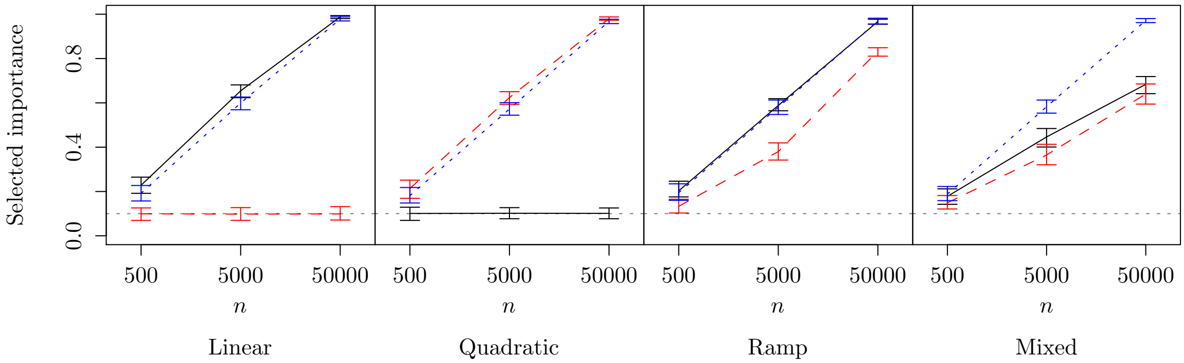

Fig. 7: The ratio of total relative importance of the selected variables as in Fig. 6. The black solid lines, red long dashes, and blue short dashes indicate the *mean value*, *variance* and *mixed moments* selection methods, respectively. The lines are the average ratio for each parameter combination, and the error bars indicate the 5th and 95th percentiles. The horizontal dashed line is the significance level used in the variable selection. Created using the TikZDevice R package.

percents, respectively. The relatively good performance of SVM for the linear datasets is most likely a result of the posterior probabilities being computed by logistic regression in the kernel space, which should be similar to the observation space for an optimally tuned kernel function. Again, the relatively low increase in RMSE of SVM and GBDT for the ramp datasets when compared to PLR can be explained by the inability of PLR to model the ramp shape function. Similarly to LinR and LASSO, SVM and GBDT seem to be relatively robust for less well-behaved datasets, see Table 3 and Fig. 8.

4 Discussion

In this paper we have introduced a framework for simulating datasets similar to those collected from real world manufacturing processes. The simulated datasets consist of simulated explanatory x-variables, unobservable *defect probabilities* computed from the x-variables, and observable binary y-variables, where a value of 0 and 1 would represent a non-defect or defect observation, respectively. The framework allows us to evaluate methods for predicting the defect probabilities from the x-variables and y-variables, and allows us to evaluate methods for interpreting the predictive models. We believe that this approach is useful when the collected explanatory data explain a small part of the variation in the defect probability. This would facilitate the evaluation of the predictive models by giving an indication of whether the remaining variation should be attributed to the missing explanatory data, or to poor model fitting.

When using the framework to simulate datasets, various parameters can be modified. This includes the distributions of the x-variables, the prior probability of a defect distribution, the distribution of the importances of the x-variables, the overall explanatory power of the x-variables, and the relationships between the x-variables and the defect probability. For our simulated datasets, we have chosen our parameters both with the aim of making the datasets similar to our real world application, as well as making the results easily interpretable. The linear, quadratic and ramp shape functions used in our simulations could represent a negative control as well as a misspecified control limit, a correct control limit, and a one-sided control limit, respectively. We suggest a data-driven approach based on visualizations of the collected data when choosing parameters for the proposed framework.

The proposed framework allows for large variations in simulation parameters. The x-variables can be simulated i.i.d. as in our case, the distributions can be estimated from real data including correlations and autocorrelations, or real data can be used as is. The relationships between the x-variables can be varied, and interaction terms can be added through simple extensions. Additionally, noise can be added to the observable x-variables,

Table 3: A comparison of the 14 prediction procedures when predicting the defect probabilities of the *linear*, *quadratic*, *ramp* and *mixed* datasets. The procedures were obtained by combining the prediction methods linear regression (LinR), LASSO, logistic regression (LogR), penalized logistic regression (PLR), support vector machines (SVM), gradient boosting decision trees (GBDT) with variable selection using the *mixed moments selection* method (MMS) and the addition of the squares of the explanatory variables as separate explanatory variables (x2). The table shows the results of pair-wise comparison between different procedures, where the reference procedure is compared to the alternative procedure, using the one-sample t-test based on the differences in RMSE for 10 dataset realizations. For each comparison the relative change (RC) was observed together with the corresponding p-value: $*$: $p < 0.05$; $**$: $p < 0.01$; $***$: $p < 0.001$. Investigated in the five blocks from top to bottom: the effect of adding squared x-variables; the effect of adding variable selection; a comparison between the variable selection methods; the choice of link function, i.e., linear for LinR, and logistic for PLR; a comparison between the best regression method, i.e., PLR and the classification methods, i.e., SVM and GBDT. †: squared x-variables were added for the quadratic, ramp and mixed dataset types.

Procedures		Linear		Quadratic		Ramp		Mixed	
Reference	Alternative	RC	p	RC	p	RC	p	RC	p
LinR	LinR-x2	0.260	$***$	-1.715	$***$	-0.689	$***$	-1.076	$***$
LinR-MMS	LinR-MMS-x2	0.160	$***$	-1.827	$***$	-0.766	$***$	-1.172	$***$
LASSO	LASSO-x2	0.150	$***$	-1.791	$***$	-0.704	$***$	-1.133	$***$
LogR	LogR-x2	0.525	$***$	-1.823	$***$	-0.735	$***$	-1.152	$***$
LogR-MMS	LogR-MMS-x2	0.380	$***$	-2.099	$***$	-0.894	$***$	-1.378	$***$
PLR	PLR-x2	0.236	$***$	-2.283	$***$	-0.875	$***$	-1.463	$***$
LinR†	LinR-MMS†	-0.057	$***$	-0.128	$***$	-0.092	$***$	-0.115	$***$
LinR†	LASSO†	-0.006	$***$	-0.113	$***$	-0.026	$***$	-0.073	$***$
LogR†	LogR-MMS†	-0.194	$***$	-0.293	$***$	-0.181	$***$	-0.249	$***$
LogR†	PLR†	-0.211	$***$	-0.498	$***$	-0.170	$***$	-0.338	$***$
LinR-MMS†	LASSO†	0.051	$***$	0.015	$*$	0.066	$***$	0.042	$***$
LogR-MMS†	PLR†	-0.017		-0.205	$***$	0.012		-0.089	$***$
LinR-MMS†	PLR†	-0.527	$***$	-0.477	$***$	-0.134	$***$	-0.300	$***$
PLR†	SVM-MMS	0.779	$***$	1.627	$***$	0.429	$***$	1.157	$***$
PLR†	GBDT-MMS	1.316	$***$	1.458	$***$	0.451	$***$	0.920	$***$

or some can be excluded intentionally when predicting the defect probabilities. Through simple variations in the simulations a more thorough understanding of a given dataset can be developed. Moreover, if no model with a good fit can be obtained when analyzing a real world dataset, the cost of utilizing our framework for investigating why is low.

In this study we have proposed two preprocessing methods: the *mixed moments selection* (MMS) method for variable selection, and the addition of squared explanatory x-variables before prediction. MMS uses both the sample mean and variance for its inclusion criterion, and we compared this method to the established *mean value selection* and *variance selection* methods, that use only the sample mean and sample variance, respectively. We utilized the proposed framework to show that MMS can be as good or better than the other selection methods, and is robust with respect to varying relationships between the x-variables and the defect probabilities. Note that we had a large number of observations, and a similar number of non-defect and defect observations. If this was not the case, the p-value threshold criterion of the MMS method might have to be reevaluated. Furthermore, the addition of squared x-variables seemed to be a cheap and effective approach for alleviating the effect of non-linear relationships between the x-variables and the defect probabilities. Finally, it is worth noting that MMS followed by the addition of squared x-variables respects the hierarchy principle, i.e., that if higher order terms are included in a model, the corresponding lower order terms should be included as well.

When predicting the defect probabilities, we have showed that it is possible to successfully predict the defect probabilities even though a minority of the variance in the binary y-variables can be explained by the explanatory data. We have also showed that the Y-RMSE (14) can be a useful proxy for the RMSE (15) when predicting probabilities both when tuning hyperparameters, and when evaluating a model fit. For our simulated datasets, when predicting the defect probabilities the regression methods, i.e., linear regression, logistic regression, LASSO and penalized logistic regression, outperformed the classification methods, i.e., support vector machines and gradient boosting decision trees by a large margin. This is most likely due to the almost ideal conditions for the regression methods, especially with respect to the distributions of the explanatory x-variables. We expect the performances of the regression methods to deteriorate faster for less ideal conditions, such as replacing the x-variables with real world data. This hypothesis is supported by the relatively good performances of the classification methods for the ramp and mixed datasets, which cannot be perfectly modeled by any of the regression methods.

We have proposed a general simulation framework that can be used to evaluate and develop methods for monitoring quality in a feature-rich manufacturing process. Arguably, monitoring and optimizing this type processes is crucial for combining high quality with low cost. The framework enables strategic development rather than reactive actions, which is likely to result in better modeling approaches, reduction in development time, and the facilitation of the understanding of the utilized approaches. The proposed framework can be used to develop methods for root-cause analysis and real-time quality surveillance systems, which makes it an attractive tool when facing some of the current challenges in intelligent manufacturing.

5 Declarations

5.1 Conflicts of interest

The authors declare that the have no conflicts of interest.

5.2 Funding

This study was funded by:

- Volvo Group Truck Operations (grant ID: FS 2.1.6-510-15)[1]
- Industrial Doctoral School for Research and Innovation, Umeå University (grant ID: FS 2.1.6-510-15)[2]
- Faculty of Science and Technology, Umeå University (no grant ID)[3]
- Vinnova FFI (grant ID: 2015-03706)[4]

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

5.3 Availability of data and material

The datasets generated and analysed during the current study are available from the corresponding author on reasonable request.

5.4 Code availability

Demonstration example is available at: github.com/niklasfries/prediction-framework-demo/

5.5 Authors' contributions

- Niklas Fries: Design, programming, writing original draft, review and editing, final approval.
- Patrik Rydén: Design, review and editing, final approval.

Conflicts of interest

The authors declare that they have no conflicts of interest.

References

[1] Bai Y, Sun Z, Deng J, et al (2017) Manufacturing quality prediction using intelligent learning approaches: A comparative study. Sustainability 10(2):85. https://doi.org/10.3390/su10010085, URL https://doi.org/10.3390/su10010085

[1] www.volvogroup.se

[2] www.umu.se/en/research/doctoral-studies/forskarskolor/industrial-doctoral-school-for-research-and-innovation/

[3] www.umu.se/en/faculty-of-science-and-technology/

[4] www.vinnova.se/en/m/strategic-vehicle-research-and-innovation/

[2] Breiman L (2001) Random forests. Machine learning 45(1):5–32. https://doi.org/https://doi.org/10.1023/A:1010933404324

[3] Burkholder D (1962) Contributions to probability and statistics: Essays in honor of harold hotelling. https://doi.org/https://doi.org/10.2307/2282455

[4] Chang CC, Lin CJ (2011) LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2:27:1–27:27. https://doi.org/https://doi.org/10.1145/1961189.1961199, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[5] Choudhary AK, Harding JA, Tiwari MK (2008) Data mining in manufacturing: a review based on the kind of knowledge. Journal of Intelligent Manufacturing 20(5):501–521. https://doi.org/https://doi.org/10.1115/1.1763182, URL https://https://doi.org/10.1007/s10845-008-0145-x

[6] Cortes C, Vapnik V (1995) Support-vector networks. Machine learning 20(3):273–297. https://doi.org/https://doi.org/10.1007/BF00994018

[7] Friedman J, Hastie T, Tibshirani R (2010) Regularization paths for generalized linear models via coordinate descent. Journal of statistical software 33(1):1

[8] Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Annals of statistics pp 1189–1232

[9] Hsu J (1996) Multiple comparisons: theory and methods. CRC Press

[10] James G (2013) An introduction to statistical learning : with applications in R. Springer, New York, NY

[11] Kim H, Lin Y, Tseng TLB (2018) A review on quality control in additive manufacturing. Rapid Prototyping Journal https://doi.org/https://doi.org/10.1108/RPJ-03-2017-0048

[12] Köksal G, Batmaz I, Testik MC (2011) A review of data mining applications for quality improvement in manufacturing industry. Expert Systems with Applications 38(10):13,448–13,467. https://doi.org/https://doi.org/10.1016/j.eswa.2011.04.063, URL https://https://doi.org/10.1016/j.eswa.2011.04.063

[13] Liang S, Rajora M, Liu X, et al (2016) Intelligent manufacturing systems: A review. International Journal of Mechanical Engineering and Robotics Research 7(2):324–330. https://doi.org/https://doi.org/10.18178/ijmerr.7.3.324-330, URL http://dx.https://doi.org/10.18178/ijmerr.7.3.324-330

[14] Lipovetsky S, Conklin M (2001) Analysis of regression in game theory approach. Applied Stochastic Models in Business and Industry 17(4):319–330. https://doi.org/https://doi.org/10.1002/asmb.446, URL https://https://doi.org/10.1002/asmb.446

[15] Paliwal M, Kumar UA (2009) Neural networks and statistical techniques: A review of applications. Expert systems with applications 36(1):2–17. https://doi.org/https://doi.org/10.1016/j.eswa.2007.10.005

[16] Pilario KE, Shafiee M, Cao Y, et al (2020) A review of kernel methods for feature extraction in nonlinear process monitoring. Processes 8(1):24. https://doi.org/https://doi.org/10.3390/pr8010024

[17] Platt J, et al (1999) Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Advances in large margin classifiers 10(3):61–74

[18] Ribeiro MT, Singh S, Guestrin C (2016) ”why should i trust you?”. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, https://doi.org/https://doi.org/10.1145/2939672.2939778, URL https://https://doi.org/10.1145/2939672.2939778

[19] Rostami H, Dantan JY, Homri L (2015) Review of data mining applications for quality assessment in manufacturing industry: support vector machines. International Journal of Metrology and Quality Engineering 6(4):401. https://doi.org/https://doi.org/10.1051/ijmqe/2015023

[20] Schölkopf B, Smola A, Müller KR (1997) Kernel principal component analysis. In: International conference on artificial neural networks, Springer, pp 583–588, https://doi.org/https://doi.org/10.1007/BFb0020217

[21] Shrikumar A, Greenside P, Kundaje A (2017) Learning important features through propagating activation differences. arXiv preprint arXiv:170402685

[22] Štrumbelj E, Kononenko I (2013) Explaining prediction models and individual predictions with feature contributions. Knowledge

and Information Systems 41(3):647–665. https://doi.org/https://doi.org/10.1007/s10115-013-0679-x, URL https://https://doi.org/10.1007/s10115-013-0679-x

[23] Welch BL (1947) The generalization of 'Student's' problem when several different population variances are involved. Biometrika 34(1-2):28–35. https://doi.org/https://doi.org/10.1093/biomet/34.1-2.28, URL https://https://doi.org/10.1093/biomet/34.1-2.28

[24] Witten DM, Tibshirani R, Hastie T (2009) A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. Biostatistics 10(3):515–534. https://doi.org/https://doi.org/10.1093/biostatistics/kxp008, URL https://https://doi.org/10.1093/biostatistics/kxp008

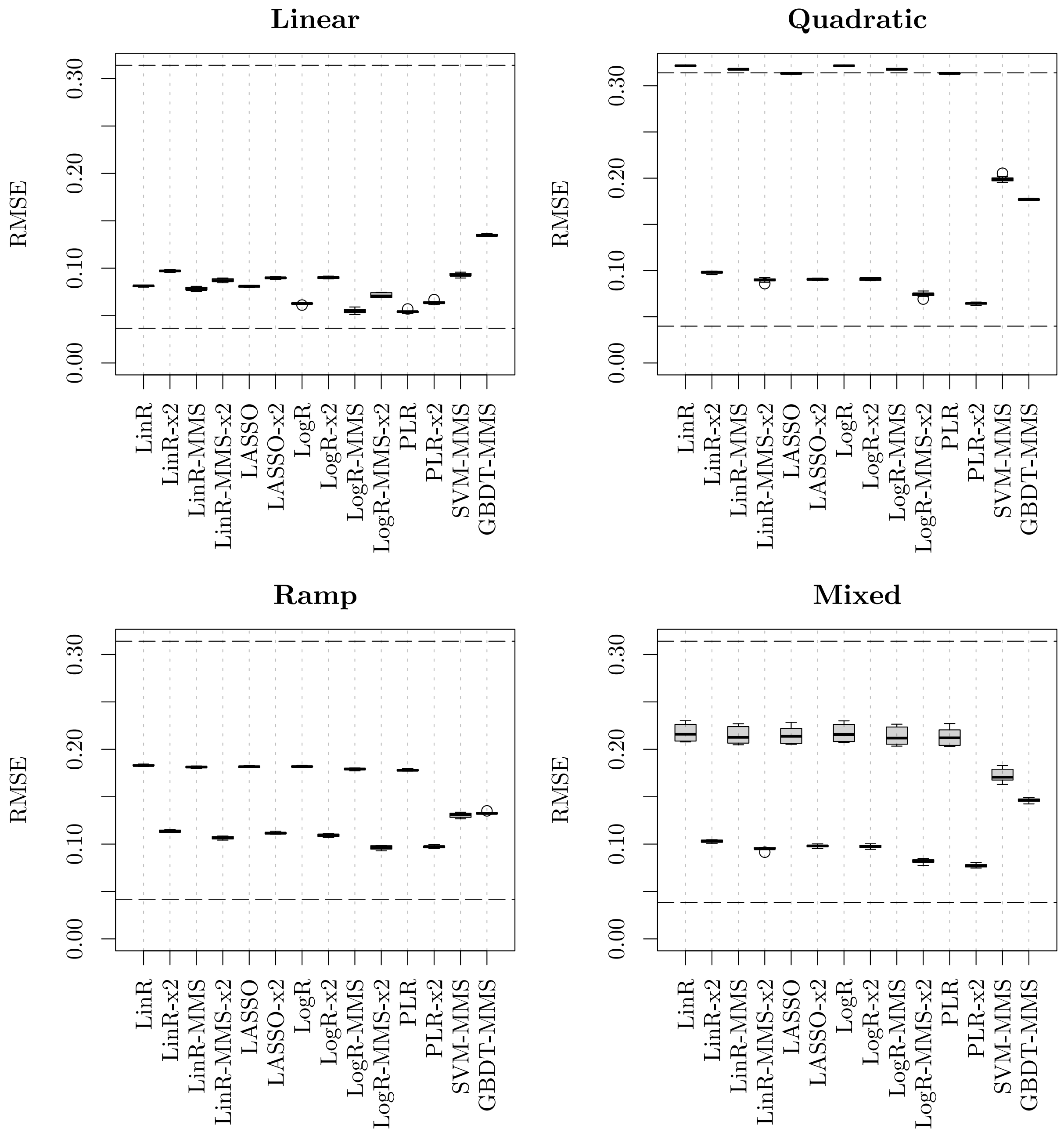

Fig. 8: Boxplots of the out-of-sample performances of the 14 prediction procedures when used to predict the defect probabilities for the *linear*, *quadratic*, *ramp*, and *mixed* dataset types. The top dashed lines are the performances of the negative control, i.e., the constant model. The bottom dashed lines are the performances of the positive control, i.e., the correct model with perfect variable selection and correct contribution shapes. The prediction methods are linear regression (LinR), LASSO, logistic regression (LogR), penalized logistic regression (PLR), support vector machines (SVM), and gradient boosting decision trees (GBDT). The suffix "MMS" indicates that variable selection was performed using the *mixed moments selection* method. The suffix "x2" indicates that the squares of the explanatory variables were added as separate explanatory variables. Created using the TikZDevice R package.

A Supplementary Experiment 1

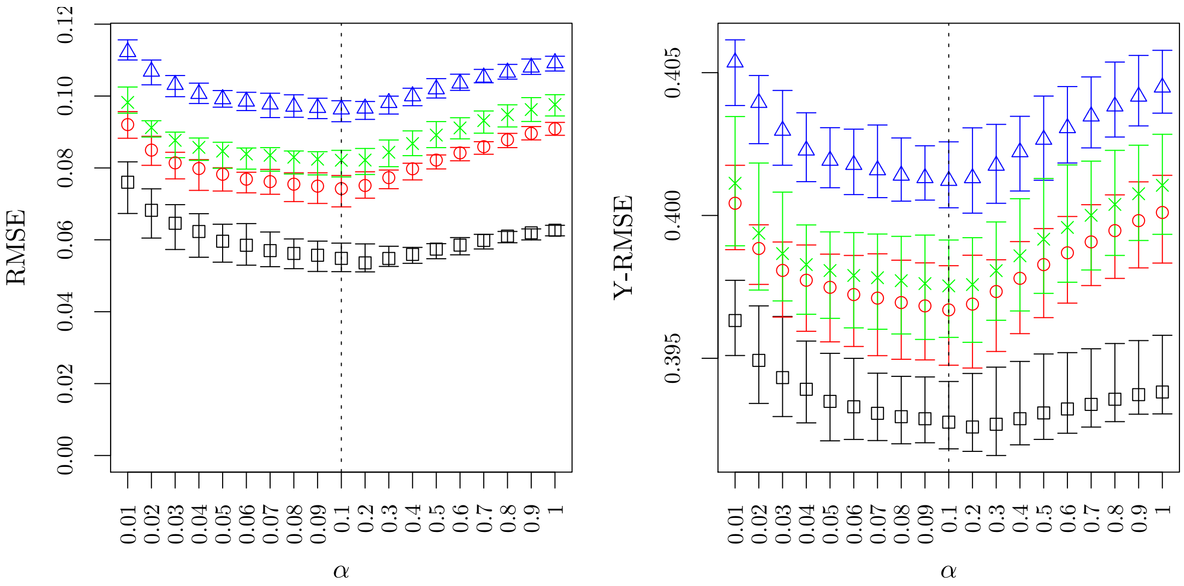

Fig. 9: Left: The out-of-sample RMSEs for the *mixed moments* variable selection method with varying significance level thresholds α, see (14), Section 2.2, and logistic regression predictor, see Section 2.3. The α values shown are those before the Bonferroni correction. The black squares, red circles, blue triangles, and green crosses are the RMSE-values for the linear, quadratic, ramp, and mixed dataset types, respectively, see Section 2.1. Squared x-variables were included except for the linear dataset type. Squared explanatory variables were included for all datasets except for the linear type, see (20). Right: the Y-RMSE-values for the same predictors, see (15). For all datasets the number of observations for the training and test parts were 50 000 each, and the number of explanatory variables were 1000. Created using the TikZDevice R package.