Field-Reliability Predictions Based on Statistical System Lifecycle Models

Lukas Felsberger, Dieter Kranzlmüller, Benjamin Todd

Field-Reliability Predictions based on Statistical System Lifecycle Models*

Lukas Felsberger[1,2], Dieter Kranzlmüller[1], and Benjamin Todd[2]

[1] Institut für Informatik, Ludwig Maximilians Universität Muenchen, Oettingenstr. 67, D-80538 München, Germany
[2] CERN, Route de Meyrin, 1211 Genève, Switzerland

Abstract. Reliability measures the ability of a system to provide its intended level of service. It is influenced by many factors throughout a system lifecycle. A detailed understanding of their impact often remains elusive since these factors cannot be studied independently. Formulating reliability studies as a Bayesian regression problem allows to simultaneously assess their impact and to identify a predictive model of reliability metrics.
The proposed method is applied to currently operational particle accelerator equipment at CERN. Relevant metrics were gathered by combining data from various organizational databases. To obtain predictive models, different supervised machine learning algorithms were applied and compared in terms of their prediction error and reliability. Results show that the identified models accurately predict the mean-time-between-failure of devices – an important reliability metric for repairable systems - and reveal factors which lead to increased dependability. These results provide valuable inputs for early development stages of highly dependable equipment for future particle accelerators.

Keywords: Reliability Prediction · System Lifecycle · Bayesian Learning.

1 Introduction

Reliability measures the ability of a system to perform as expected during its intended lifetime. The field-reliability of complex repairable systems is a result of all actions during all stages of its system lifecycle. These stages are (1) conceptual design, (2) detailed design and testing, (3) manufacturing, (4) installation, (5) operation and maintenance, and (6) phase-out and disposal. At each stage an interplay of complex technical, organizational, and human processes leads to a more or less desirable outcome in terms of system reliability.
An assessment of all stages and processes is not feasible, since models capturing the interactions between all relevant processes in system development do not exist. Therefore, most common reliability methods focus on certain stages and

* This work has been sponsored by the Wolfgang Gentner Programme of the German Federal Ministry of Education and Research (grant no. 05E12CHA)

aspects during a system lifecycle, which can be modeled and understood - we provide an overview in section 2. However, such methods struggle to quantify the overall uncertainty of reliability predictions in a systematic way since relevant contributions during a system lifecycle might have been disregarded and are not straight-forward to include.
Instead of focusing on models for certain stages and aspects of a system we propose to learn a statistical model of the whole product lifecycle to predict the observed field-reliability with machine learning techniques as depicted in figure 1. For a set of existing comparable systems with known field-reliability so-called *quantitative reliability indicators* are gathered. Using the reliability indicators as input variables and the field-reliability metric as target variables, a statistical reliability model is learned by a supervised machine learning algorithm.
The learned model will always be an approximation of the true underlying system lifecycle processes. The lost accuracy due to the statistical model and the limited granularity of the reliability indicators can be quantified by Bayesian methods. Thereby, the overall predictive certainty can be quantified in an efficient way based on the available data.

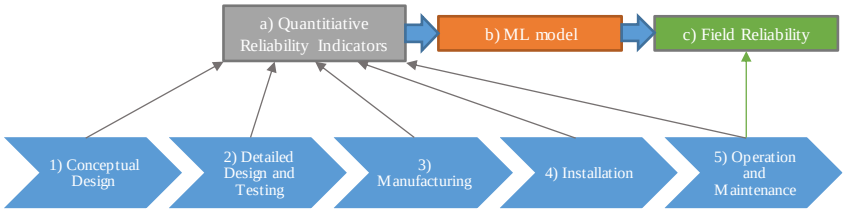


Fig. 1. Illustration of the proposed approach. The achieved field-reliability (c) can be seen as the result of relevant processes during the whole product lifecycle (1-5). It is not feasible to capture and model all of the relevant processes. Instead, it is proposed to learn a reduced-order statistical lifecycle model (b) with machine-learning algorithms based on *quantitative reliability indicators* (a).

We demonstrate that the learned models accurately predict reliability metrics even with a limited set of reliability indicators (as is the case at early stages of a system's lifecycle). Compared to traditional reliability assessment methods, this leads to a reduced workload for reliability predictions and to a systematic quantification of uncertainties. Furthermore by an appropriate choice of reliability indicators and machine learning algorithms one can study the influence of each individual reliability indicator. This information assists engineers in design decisions for highly reliable systems.
The rest of the paper is structured as follows: In section 2, we present related

methods to reliability predictions. In section 3, we explain the methodology of our approach and in section 4 we apply it to a use-case.

2 Literature Review

A general review of the challenges in reliability studies is given in [23]. The author of [23] concludes that the two major challenges in reliability studies are complexity and uncertainty. Reliability studies must consider technical, organizational and human factors each of which influences the field-reliability of systems. In the following paragraph a selection of reliability prediction methods to tackle these problems is given.

Reliability Engineering Methods Scientific literature on reliability engineering prediction methods of electronic systems is numerous. An attempt to classify and evaluate the existing methods is given in the IEEE standard 1413 [19, 6] and its successors. In this standard they have been classified as based on

- handbooks,
- stress and damage models (often referred to as physics-of-failure based), and
- field-data.

Most methods are based on early designs of the considered system and the selected components.
A common criticism for handbook based models is that they do not consider interactions of components but only single-component faults. However, faults due to single-component failures are not dominant [18, 5, 7, 1, 14]. As a result the actual field-reliabilities can deviate from the predicted ones by orders of magnitude [12]. The author of [5] argues that some methods should not be used to predict the field-reliabilities but rather as part of a review process at a stage when limited information on the final design is available.
Stress- and damage models are in general more accurate than handbook-based methods. However, the development of such methods requires more effort [18].
Instead of assessing the system on the component level, some approaches use a top-down approach in which the field-reliability of new systems is estimated from field-data of similar systems in operation [9, 11].

Reliability Program Assessment A different approach to evaluate the field-reliability of systems is taken in [16]. The likelihood of achieving the required field-reliability is estimated by a review of the design processes. Each system is assigned a score depending on its design processes and it is shown that this score correlates with the probability of fulfilling field-reliability requirements. Thereby organizational aspects of reliability are taken into account.

Organizational and Human Reliability Analysis In the review article [23] section 3.1.3 is dedicated to non-technical factors in reliability studies since its contribution to the field-reliability can be significant.

In our work we propose to infer the most relevant processes or factors in a system lifecycle from the field-reliability data of a set of systems. This includes organizational and human reliability factors. The method can be applied at any stage of a system lifecycle to guide engineering decisions.

3 Methodology

In this section we define the relevant terms, explain the methods used and describe the general methodology.

3.1 Definitions

System Reliability It is generally defined as the ability of a system to provide its intended level of services for a specified time t. For a constant failure rate and repairable systems, it is usually measured as availability A, which is defined by

$$A = \frac{MTBF}{MTBF + MTTR} \tag{1}$$

with $MTBF$ being the mean-time-between-failure and $MTTR$ being the mean-time-to-repair. The $MTBF$ is being calculated as

$$MTBF = \frac{t_{operation}}{n_{faults}} \tag{2}$$

with $t_{operation}$ being the cumulative operational time of the considered devices and n_{faults} being the total number of faults within the operational time. The $MTTR$ can be evaluated by

$$MTTR = \frac{t_{inrepair}}{n_{faults}} \tag{3}$$

with $t_{inrepair}$ being the total time a system is in repair and n_{faults} the total number of faults during the operational time. The un-availability U_A is given by $U_A = 1 - A$.

System Lifecycle It is the overall process describing the lifetime of a system. It is a concept from systems engineering to address all stages of a product from its beginning to end. Here these stages shall be divided into (1) conceptual design, (2) detailed design and testing, (3) manufacturing, (4) installation, (5) operation and maintenance, (6) and phase-out and disposal.[3]

[3] Depending on the system under study the definitions of the stages may change. The proposed methodology is not restricted to this specific choice of stages.

System Definition This discussion is focused on repairable electronic systems. A more precise definition will be given for the use-case in section 4. [4]

3.2 Method

The central assumption is that the observed field-reliability is the outcome of all technical, organizational and human processes during all stages of a system's lifecycle. It is unfeasible to model all these interactions due to their complexity and non-linearity. Therefore, we restrict ourselves to learning statistical models of the observed field-reliability of comparable systems based on reliability indicators collected throughout the system lifecycle. Modern machine learning algorithms are capable of learning accurate predictive models of field-reliability based on the relevant reliability indicators. The loss of information due to the limited availability of data and the intrinsic uncertainty of the problem can be assessed by using Bayesian machine learning methods.

Lifecycle Analysis by Machine Learning To arrive at a firm mathematical description of the proposed method, let us hypothesize the existence of a deterministic model $\mathbf{F} : \mathcal{Z} \mapsto \mathcal{Y}$ to determine any field-reliability metric $\mathbf{Y} \in \mathcal{Y}$ from all relevant input variables $\mathbf{Z} \in \mathcal{Z}$ in the form of

$$\mathbf{Y} = \mathbf{F}(\mathbf{Z}). \tag{4}$$

This would be a model to quantify the contribution of all relevant processes towards the field-reliability during the whole system lifecycle. Since it is not possible to derive such a formula or to gather all relevant inputs, we try to approximate the true field-reliability metrics $\mathbf{Y}$ by a reduced model

$$\mathbf{Y} \approx \mathbf{y} = f(\mathbf{x}), \tag{5}$$

with $\mathbf{x} \in \mathcal{X}, dim(\mathcal{X}) \ll dim(\mathcal{Z})$ being the set of collected reliability indicators and $f : \mathbf{x} \mapsto \mathbf{y}, \mathbf{y} \in \mathcal{Y}$ being an approximate model. When supplied with pairs of input and output data $\mathcal{D} = \{(\mathbf{x_1}, \mathbf{Y_1}), ..., (\mathbf{x_N}, \mathbf{Y_N})\}$, a statistical learning algorithm can learn such a model by minimizing a certain loss function $l : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$. This is essentially a regression problem which can be studied with a vast range of learning algorithms.

There are three additional requirements, which render algorithms fitter for the intended purpose. Firstly, to quantify the uncertainty of the predictions of the reliability metrics, probabilistic models shall be learned

$$p(\mathbf{Y}|\mathbf{x}). \tag{6}$$

[4] There is no implicit restriction for the proposed method to electronic repairable systems. It can also be used for non-repairable systems and for mechanic, electric, electronic, or software systems. However, the definitions of the fault metrics must be adapted.

Our method is based on an arbitrary non-linear mapping from reliability indicators to features $\Phi : \mathcal{X} \mapsto \mathbb{R}^n$. Since it is of interest which features are relevant, secondly, algorithms of parametric form will yield that additional information,

$$p(\mathbf{Y}|\mathbf{w} \cdot \Phi(\mathbf{x})), \tag{7}$$

with $\mathbf{w} \in \mathbb{R}^n$ being a weight vector indicating the relevance of each feature. Thirdly, methods learning sparse models based on fewer features are preferred from a practical point of view, since they require a reduced data collection effort for predicting field-reliability. A general justification of such methods on philosophical grounds is given by Occam's razor [8].
We present concrete algorithms fulfilling these criteria in section 4. Even though the outlined requirements are not mandatory, they facilitate the data collection and model assessment process by providing additional feedback.

Data Collection, Model Selection and Reliability Prediction The collection of data and the training and selection of a model should be seen as an integrated process. The problem domain and a-priori available expert knowledge allows to draw guidelines for the data collection. We present these guidelines in the paragraphs below. After that, we show how to learn a predictive model with the collected data and how further refinements of the data collection are assisted by properly selected learning algorithms.

Collection of Training Systems Since the method is based on the field-reliability of existing comparable systems, the choice of the collected systems will have an influence on the accuracy of the predictions for future systems. Two general recommendations can be given for this selection:

- Only systems which have been in use for a significant exploitation period with accurately monitored reliability metrics shall be used.
- The choice of systems for which a field-reliability model is learned shall include systems which are comparable to the system for which a field-reliability shall be predicted. In reliability studies, comparable systems are similar in terms of technical, organizational, and human factors throughout their lifecycle.

Collection of Reliability Indicators The choice of these indicators largely influences the quality of the models in terms of their accuracy and interpretability. The following statements can be made:

- Based on expert knowledge, recommendations can be given for indicators which carry important reliability information; e.g. operational conditions such as load, temperature or humidity can contribute significantly to the failure rate. Systems, which are mass-produced, will achieve different field-reliabilities than prototypes. Different organizational structures or project management strategies influence the reliability of a final system. [17, 22].

– In engineering practice the collection of data is facing practical limitations due to time or other restrictions. Therefore, a natural choice is to begin to collect the indicators based on a trade-off between collection effort and expected information content. For the use-case in section 4, we show that accurate predictions can be obtained from a very limited set of meta-variables as reliability indicators. Furthermore, one always needs to consider the availability of the indicators for the systems in the data-set.[5]

Collection of System Reliability Metrics The choice of reliability metrics is usually given by the system under study. For our choice of system and assuming a constant failure rate,[6] these are given by $MTBF$ and $MTTR$. Based on these other metrics can be derived.

Model Selection and Validation Using the collected data, one is able to compile a data-set $\mathcal{D} = \{(\mathbf{x_1}, \mathbf{Y_1}), ..., (\mathbf{x_N}, \mathbf{Y_N})\}$ for which $\mathbf{x_i}$ and $\mathbf{Y_i}$ are the collected reliability indicators and the field-reliability metrics for system i, respectively. A reliability model shall be learned with this data-set. We use a general model selection and assessment approach as is e.g. discussed in chapter 7 of [10] with minor modifications due to the particularities of the problem setting.
The first step is to split the data-set into a training data-set $\mathcal{D}_{train}$ and a test data-set $\mathcal{D}_{test}$. This splitting is not performed arbitrarily. Instead the training data-set shall contain systems with an age higher than a certain threshold age a_s and the test data-set shall exclusively contain systems younger than the threshold age. Thereby, we test the approach for its applicability to future systems.
For the model selection and assessment the training data-set will be used exclusively.[7] With a five-fold cross-validation method we compare different learning algorithms in terms of their applicability to the problem setting and their prediction errors. For algorithms which additionally require the tuning of hyperparameters, we used so-called nested cross-validation in which the hyperparameters are optimized in a five-fold inner cross-validated grid-search nested within each of the five outer cross-validation folds [4]. The expected mean and variance of the cross-validation error Err_{CV} is reported for each of the evaluated models. It serves as an estimate for the expected generalization error [10].
The confidence of the predictions and the relevance of the selected reliability indicators can be studied with a learning algorithm which satisfies equations 6 and 7. Investigating the identified model parameters and predictions obtained by such an algorithm for one or several cross-validation folds gives this additional information. The confidence or uncertainty of the predictions provides feedback

[5] If availability indicators are unavailable for some of the selected systems, supervised learning techniques for incomplete data-sets can be employed.

[6] This assumption can be relaxed by e.g. predicting a parameterized failure rate distribution over time. Then, instead of $MTBF$ and $MTTR$ the reliability metrics are the parameters of the distribution. This requires a different data collection and can be considered for future work.

[7] Using the test data-set would lead to an over-fitting of the models and an underestimation of the generalization error.

on the amount and quality of the collected data. The weight vector $\mathbf{w}$ indicates the relevance of the features and the reliability indicators. Depending on the complexity of the mapping $\Phi : \mathcal{X} \mapsto \mathbb{R}^n$ from the reliability indicators to the features we can identify the most important reliability indicators. Using this information and expert knowledge, we can refine our data-set (choice of systems and reliability indicators) and feature mapping Φ to obtain more precise models. This idea is illustrated in figure 3.2

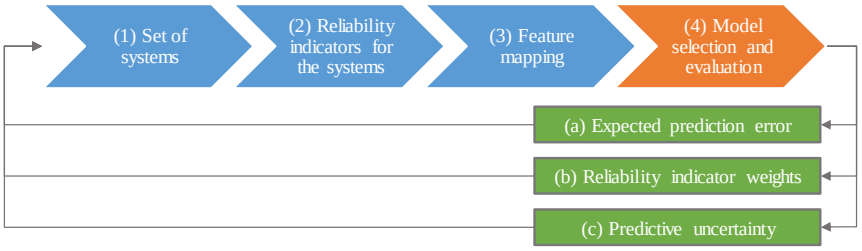


Fig. 2. Illustration of the iterative data collection and reliability prediction process. The choice of (1) systems, (2) reliability indicators and (3) feature mappings influences the quality of the predictive model (4). The learning algorithm provides feed-back in the form of an expected prediction error (a), relevance weights for the reliability indicators (b) and uncertainty bounds for the field-reliability predictions (c).

Obtaining Reliability Predictions Once satisfying models in terms of their predictive errors and interpretability are found with the procedure described above, they are tested with the full data-set. Since the data-set is split by the age of the systems, this testing simulates a prediction scenario - we identify a model based on data of systems in the past and evaluate its applicability to future systems. The predictive models are now trained with the whole training data-set.[8] Based on the input values of the test data-set $\mathbf{x_{test}}$ the models can predict the expected field-reliability $\mathbf{y_{test}}$. As the prediction is simulated, we know the observed field-reliabilities $\mathbf{Y_{test}}$ and can compare these to the predicted ones to obtain the test error Err_{test}.[9] When the test error is of the order of the expected generalization error Err_{CV} obtained during model selection and validation one can conclude that the model is capable of predicting the field-reliability for new systems. The overall data collection, model selection and reliability prediction process is

[8] Again, hyperparameters are optimized by a cross-validated grid search over a hyperparameter grid.

[9] We note that in a realistic application scenario the true observed field-reliabilities are not available. However, the available data can always be split by system age to test the generalizability of the identified models to newer systems.

summarized in the pseudo-algorithm below. The use-case in section 4 follows the presented procedure closely.

Pseudoalgorithm illustrating the overall model selection and reliability prediction process:

1. $\mathcal{D} = \{(\mathbf{x_1}, \mathbf{Y_1}), ..., (\mathbf{x_N}, \mathbf{Y_N})\} \leftarrow$ Initial data collection.
2. Sort $\mathcal{D}$ by system age.
3. Split $\mathcal{D}$ in $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$ with $a_{test} < a_s \leq a_{train}$.
4. **While** satisfying predictive model has not been found **do:**
   (a) Shuffle $\mathcal{D}_{train}$ randomly.
   (b) Evaluate Err_{CV} by (nested) CV.
   (c) Evaluate parameter weights $\mathbf{w}$ and predictive uncertainty for one fold.
   (d) **If** Model has large Err_{CV} or predictive uncertainty **then**
      – Change set of systems, reliability indicators, or feature mapping.
   (e) **Else** jump to 5.
5. Train predictive model with $\mathcal{D}_{train}$.
6. Evaluate Err_{test} and compare with Err_{CV}.
7. Evaluate parameter weights $\mathbf{w}$ and predictive distributions.

4 Use-Case

This section describes how the proposed method was used to learn a model for the expected field-reliability of accelerator power converters. The system of interest, the collected data and features, the used learning algorithms and the results are discussed.

System Definition The considered systems are magnet power converters at the CERN particle accelerator facilities. A power converter is a device to transform electrical energy. The conversion is in terms of voltage, current and waveform. Magnet power converters control the flow of current through particle accelerator magnets. In order to achieve precise magnetic fields these converters generally need to control the output current very precisely.

Dataset, Reliability Metrics and Reliability Indicators

Set of Systems At CERN there are currently more than 6000 power converters of approximately 600 different types in use. Their field-reliability is continuously tracked by a centralized computerized maintenance management system (CMMS). After removal of converter types with a cumulative operational time $t_{operation}$ of less than ten years and cleaning the data, approximately 300 power converter types remained for reliability analysis. Table 1 gives an overview of minimal and maximal characteristic attributes of power converters in the

dataset. Considering the vast range of converter types one would not expect a global model to accurately predict the field-reliability. Therefore, both local- and global-models were trained.

Table 1. Illustration of characteristic power converter attributes of the studied dataset.

	Power [W]	**Current [A]**	**Voltage [U]**	**Age [yrs]**	**MTBF [hrs]**
Minimum	10^{-6}	10^{-4}	10^{-3}	2.2	10^{3}
Maximum	10^{8}	$4 \cdot 10^{4}$	10^{5}	49.7	$6 \cdot 10^{5}$

Reliability Indicators for the Systems The initial choice of reliability indicators depends on

- the system development stage at which the prediction shall be carried out,
- recommendations from system experts,
- the time or effort which can be attributed to the data collection.

The following list shows the collected reliability indicators. The selection is based on recommendations from CERN engineers in charge of the complete lifecycle of the studied systems. Naturally, the selection is also limited by the availability of data:

- I: Rated current of the converter. Depending on the rated current different converter technologies have to be used. One major stress effect of high currents in terms of reliability is an increased heat load which requires a proper heat management [17, 13].
- U: Rated voltage of the converter. Higher voltages require the appropriate electrical insulation and can cause failure mechanisms such as arcing or corona discharge [17, 13].
- P: Rated power of the converter. Similarly to currents increased power leads to increased heat loads.
- Quantity: This refers to the quantity of each type of power converter that is used at CERN. The quantity of a power converter is not related to a physical wear-out mechanism. However, throughout the lifecycle converter types produced and operated in large quantities are treated differently than power converters of small quantities both in terms of technical and organizational matters.
- Avg. Age: The average age of converters for each converter type. Depending on the maintenance strategy a decreasing or constant availability as a function of the age is expected, respectively.
- Cum. Age: The cumulative age of converters for each converter type. A dependency of the availability on the cumulative age could indicate both a organizational learning curve in terms of a more efficient maintenance and a degradation with age of the converters.

- Pol 0-9: The polarity of the converter. This indicates the operating modes, technology and complexity of the converter.[10]
- Acc. 1-9: The accelerator in which the converter type is used. Depending on the accelerator the converter type is exposed to different operating conditions[11] and operation modes.
- in Acc.: The number of different particle accelerators in which each power converter is used.

We probed different indicators for their information content by appropriate Bayesian learning methods. The required learning algorithms are introduced later in this section.

Reliability Metrics for the Systems The studied field-reliability metrics are $MTBF$ and $MTTR$ as defined in section 3.[12] These are directly computed in the CMMS with the necessary variables for power converter type i which are defined as follows:

- $t_{operation,i}$: Cumulative time in operation of all converters of converter type i. Note that commissioning and testing times are not counted towards operation time.
- $n_{faults,i}$: Cumulative number of faults of all converters of converter type i during the operational time $t_{operation,i}$. Note that only internal faults of the system which require an external action to alleviate the problem are included. Internal faults which are automatically resolved or are very short and faults due to external reasons are not included. This ensures that a model for the reliability of the considered systems itself is learned and not of its surroundings.
- $t_{inrepair,i}$: Cumulative time in repair of all converters of converter type i during the operational time $t_{operation,i}$. The repair time starts by a request from the system operators to the system experts and ends when the problem was resolved and the system can continue to operate.

Algorithms By formulating the reliability prediction problem as a supervised machine learning problem we can choose from a range of existing learning algorithms to generate the desired statistical model for predictive purposes. Since the uncertainty in the field-reliability predictions shall be quantified (i.e. finding

[10] The discrete set of polarities is given by: (1) Unipolar, (2) Bipolar Switch Mechanic, (3) Bipolar I - Unipolar U - 2 Quadrants, (4) Unipolar I Bipolar U 2 Quadrants, (5) Bipolar Pulse-Width-Modulation, (6) Bipolar Relay, (7) Bipolar Electronic I/U, (8) Bipolar Anti-Parallel 4 Quadrants, (9) Bipolar I-circulation 4 Quadrants and, (0) un-specified or other Polarity.

[11] E.g. the radiation levels differ on the kind of accelerator. However, there is also different operation conditions within each of the accelerators.

[12] Note that the Availability A and Un-Availability U_A can be directly obtained from the $MTBF$ and the $MTTR$.

a model as presented in equation 6), the choice of algorithms is narrowed down. Furthermore, sparse parametric models (as in equation 7) are preferred since they potentially require fewer reliability indicators to be collected and - more importantly - since they allow an estimation of the relevance of the choice of reliability indicators and the generated features.

A summary of the chosen algorithms is given in table 2. Note that the scikit-learn python implementations of the algorithms were used [20]. A detailed description of each algorithm can be found on their website and in their user-guide [3]. Since the algorithms are standard implementations, only references to detailed documentation are given:

Table 2. Summary of learning algorithms.

	UQ (6)	**Feature Weights (7)**	**Sparsity**	**Global/Local**
ARD	yes	yes	yes	Global
BAR	yes	yes	balanced	Global
GP	yes	no	no	Local
ENCV	no	yes	yes	Global
SVR	no	only for linear kernel	no	Local

- ARD - Automatic Relevance Determination Regression: Sparse Bayesian regression technique as described in [2] - Chapter 7.2.1. The implementation is taken from [3] - Chapter 1.1.10.2.
- BAR - Bayesian Ridge Regression: A Bayesian regression method as introduced in [15]. It is similar to the ARD Regression but fewer parameters have to be determined from the data. The implementation is taken from [3] - Chapter 1.1.10.1.
- GP - Gaussian Process Regression. A kernel-trick based Bayesian Regression technique. The implementation is described in [21] - Algorithm 2.1 and was taken from [3] - Chapter 1.7.1. The kernel is based on a combination of a radial-basis-function kernel and a white-kernel. The kernel parameters were optimized in the learning process.
- EN: Elastic Net Regression. The implementation is taken from [3] - Chapter 1.1.5 - which includes a description of the algorithm. Hyperparameters were optimized in a cross-validated grid-search.
- SVR - Support Vector Machine Regression: A kernel-trick based regression method. A description is given in [3] - Chapter 1.4.2. Linear basis functions were used and the hyperparameters were optimized by a cross-validated grid-search.

Model Selection and Validation This section closely follows the procedure presented in section 3. The data-set $\mathcal{D}$ was compiled from the data collection

described above including 281 collected systems, nine reliability indicators and two field-reliability metrics. To simulate a prediction scenario the whole data-set of 281 different converter types was split into a training set $\mathcal{D}_{train}$ with 210 converter types which are at least fifteen years old and a test set with 71 converter types which are less than fifteen years old.[13]
For the model selection and validation we restricted ourselves to the training data which we shuffled randomly. A scaling operator re-scaled the features or inputs $\mathbf{x_{train}}$ to zero mean and unit variance. The same scaling operator was later applied to the features in the test data-set $\mathbf{x_{test}}$. Furthermore, the logarithms of the reliability metrics $\log(\mathbf{Y})$ were taken instead of their nominal value for the full data-set.
Based on the introduced (nested) cross-validation we compared the follfowing different choices of the set of systems, reliability indicators and feature mappings for all the introduced algorithms:

- Choice of systems: We trained models with the complete set of power converter types and with a random sub-selection of only 42 converter types.
- Choice of reliability indicators: We trained models with the complete set of reliability indicators and a set in which the quantity of converters per type was removed.
- Choice of feature mapping: Based on the reliability indicators, following features were generated:
  - Based on the numeric indicators $\mathbf{x_{num}}$ linear features and logarithmic features were chosen - $\Phi(\mathbf{x_{num}}) = [\mathbf{x_{num}}, \log(\mathbf{x_{num}})]^T$.
  - The categorical indicators $\mathbf{x_{cat}}$ were split into binary features, whereas the number of binary variables corresponds to the number of categories per categorical variable.

  A feature vector of 34 dimensions was obtained by combining all features. This was the first choice for the feature mapping and we refer to it as first-order feature mapping.
  The second choice of feature mapping accounts for second-order interactions of the numeric variables and we refer to it as second-order feature mapping:

$$\Phi(\mathbf{x_{num}}) = \left[\mathbf{x_{num}}, \log(\mathbf{x_{num}}), [\mathbf{x_{num}}, \log(\mathbf{x_{num}})] \cdot [\mathbf{x_{num}}, \log(\mathbf{x_{num}})]^T\right]^T. \tag{8}$$

  By this more complex mapping we obtain 629 features. One could expect that a more accurate model can be learned when including second-order interactions which is balanced by a lack of interpretability of the individual feature weights.

In the following we report the results of our model selection procedure. For each algorithm the cross-validation error Err_{CV}[14] is reported and the feature weights $\mathbf{w}$ of the learned models and the obtained predictions are plotted for the

[13] In other words we pretended to be in 2003 and tried to predict the field-reliability of power converters between 2003 and 2018.
[14] Note that the mean-squared-error was used throughout.

last cross-validation fold.[15] All results are provided in terms of the two chosen reliability metrics $MTBF$ and $MTTR$.

Reference Configuration The first configuration we studied is based on the complete set of power converters, the complete set of reliability indicators and the first-order feature mapping. The cross-validation errors Err_{CV} are given in table 3a for the $MTBF$ and in table 4a for the $MTTR$. As the values for the target variables were not scaled to unit-variance but simply by a logarithmic function, the values from the $MTBF$ table cannot be compared with values of the $MTTR$ table. Only values within a table are comparable. We noted that all algorithms yielded models with comparable cross-validation errors.
The obtained parameter weights $\mathbf{w}$ for the last cross-validation fold are shown in figure 3b for the $MTBF$ and in figure 3d for the $MTTR$. All algorithms identified similar models. For the $MTBF$ the dominant parameter was the logarithm of the quantity of converters per type $\log(Qty)$ for all models and the rated power P was dominant for the $MTTR$. From the predictions obtained with the BAR algorithm for the last fold, figure 3a for the $MTBF$ and figure 3c for the $MTTR$, we noted that the model for the $MTTR$ did not identify a significant variation whereas the $MTBF$ was predicted properly. We concluded that a precise model for the $MTBF$ had been learned with the collected data, the selected feature mappings, and algorithms. For the $MTTR$ no such model could be identified and a further refinement would be necessary.
In the following we present variations of the reference configuration in terms of selected systems, reliability indicators and feature mappings.

Reduced Set of Training Systems The second configuration is similar to the reference configuration except for using a random sub-selection of only 42 converter types in the training data-set. This illustrates the dependence of the confidence levels of the identified feature weights $\mathbf{w}$ and the predictions for the Bayesian algorithms (ARD and BAR) on the amount of training data.
The cross-validation errors Err_{CV} in table 3b for the $MTBF$ and table 4b for the $MTTR$ were larger than those of the reference configuration. The obtained parameters weights $\mathbf{w}$ for the last cross-validation fold in figure 3f ($MTBF$) and in figure 3h ($MTTR$) deviated slightly in absolute terms and largely in terms of their confidence levels from the reference configuration for the ARD and BAR algorithm. The predictive uncertainties of the BAR algorithm in figure 3e ($MTBF$) and figure 3g ($MTTR$) increased only slightly in comparison with the reference configuration. Again, no predictive model of the $MTTR$ could be identified. We concluded that a reduced set of training data manifests itself in increased uncertainties in parameters or predictions.

[15] Note that only predictions obtained by the BAR algorithm are illustrated due to space limitations. It assigns relevance weights to the feature functions and and it quantifies uncertainties of both the field-reliability predictions and the feature function weights. Therefore, it is suited to study the earlier mentioned additional information provided by algorithms which satisfy equations 6 and 7.

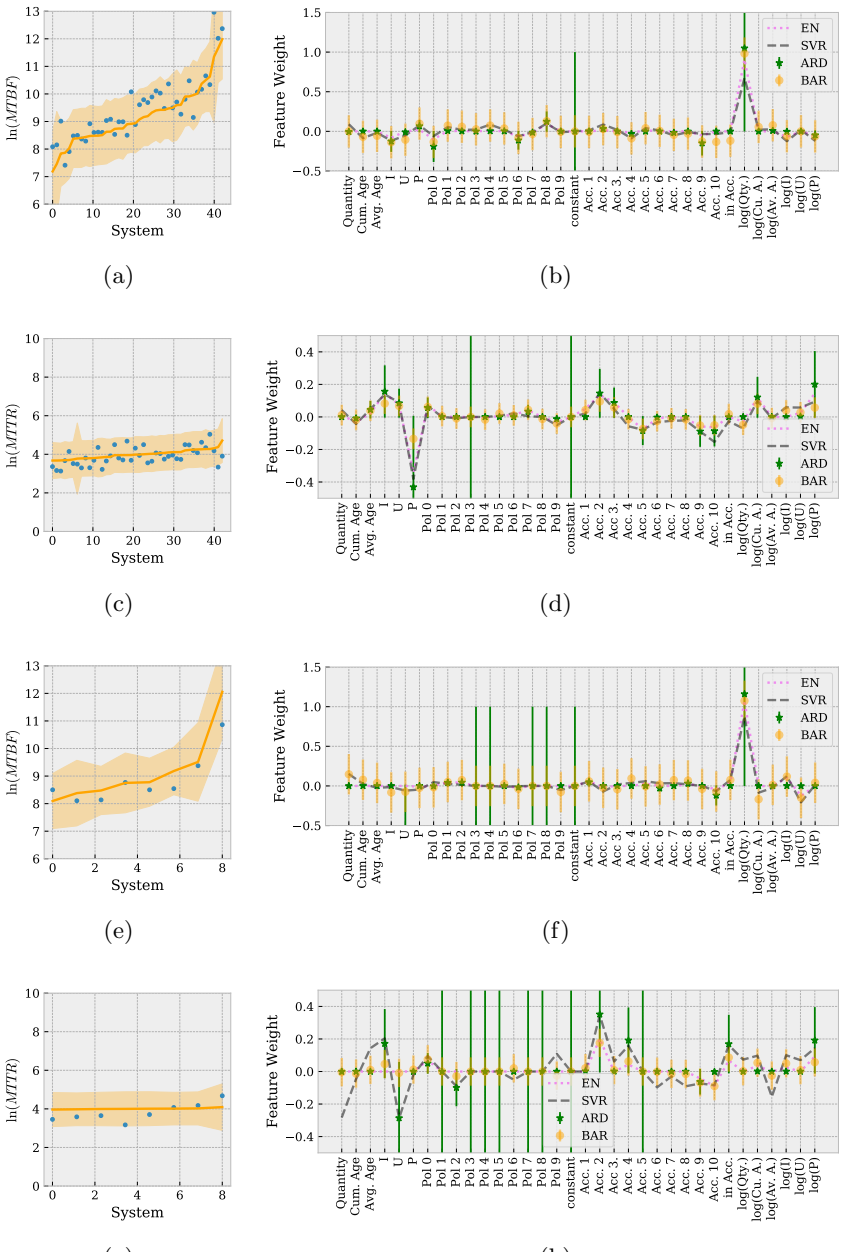


Fig. 3. (a),(c),(e),(g):Prediction of the $\log(MTBF)$/$\log(MTTR)$ for the last fold of the cross-validation procedure. The orange line depicts the mean of the predictive distribution and the orange shaded area the 95% confidence intervals. The blue dots mark the actual observed field-reliabilites. Note that the different converter types were ordered by the mean of the predictive distribution for illustration purposes.(b),(d),(f),(h):Estimated feature weights for the parametric models. Figures (a),(b),(c),(d) are for the reference configuration and figures (e),(f),(g),(h) for a reduced set of data items in the learning data.

Reduced Set of Reliability Indicators The third configuration resembles the reference configuration except for the removal of the variable indicating the quantity of systems per converter type. This variable had been identified as the single most important reliability indicator for $MTBF$ predictions.
The cross-validation errors Err_{CV} in table 3c for the $MTBF$ were much larger than those of the reference configuration and slightly larger for the $MTTR$ models (table 4c). The obtained parameter weights $\mathbf{w}$ for the last cross-validation fold in figure 4b for the $MTBF$ models were totally different than the reference configuration. The weights for the $MTTR$ models (figure 4d) were similar to the reference configuration. The predictive uncertainties of the BAR algorithm in figure 3e increased drastically for the $MTBF$ and only slightly for the $MTTR$ (figure 3g) in comparison with the reference configuration. This is consistent with our expectation, since we removed the most important reliability indicator for the $MTBF$ models. This time no proper predictive model of either the $MTTR$ or the $MTBF$ could be identified. We concluded that the choice of reliability indicators has a strong influence on the quality of the models.

Second-Order Feature Mapping In the fourth configuration the second-order feature mapping replaces the first-order mapping of the reference configuration. The cross-validation errors Err_{CV} in table 3d for the $MTBF$ and table 4d for the $MTTR$ were of the same order as those of the reference configuration except for the model learned with the ARD algorithm. The 629 obtained parameter weights $\mathbf{w}$ were not illustrated. The predictions of the BAR algorithm in figure 4e ($MTBF$) and in figure 4f ($MTTR$) were comparable with the reference configuration. No model could be identified for the $MTTR$. We concluded that the extended feature mapping does not improve the predictive errors and complicates the interpretation of the models.

Prediction The reference configuration was used for the prediction scenario as it had shown to be interpretable and predicted the $MTBF$ in the model selection procedure properly. The prediction of the $MTTR$ was not carried out since no suitable model had been identified. The predictive models were learned with the full training data-set and their predictions were evaluated with the test data-set.[16] Due to the splitting of the training and test data by the age of the systems this simulated a prediction scenario.
The test errors Err_{test} in table 3e were of the same order as the cross-validation errors Err_{CV} of the reference configuration (3a). We concluded that the learned models generalize to newer power converters. The feature weights (5b) and the predictions (5a) were consistent with our expectations and demonstrate that we could predict the $MTBF$ accurately.

Discussion One of the major insights created by applying the methods to the use-case is that the field-reliabilities are strongly dependent on the quantity of

[16] Hyperparamters were optimized by cross-validation over a parameter grid.

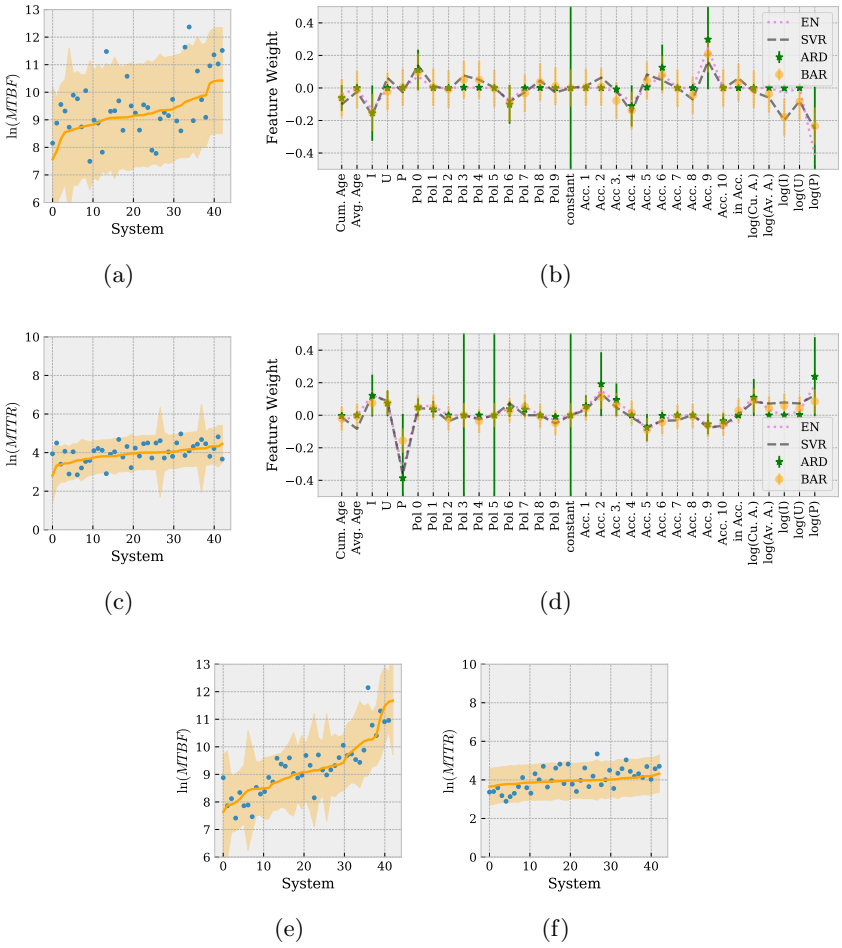
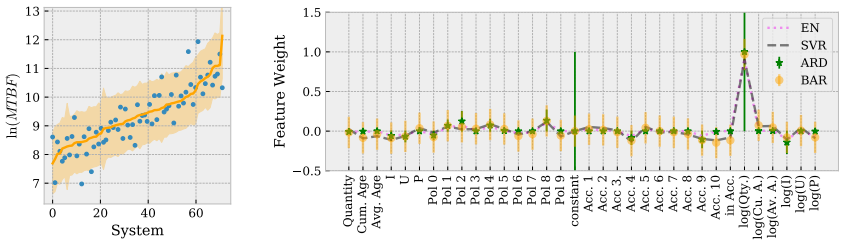


Fig. 4. (a),(c),(e),(f):Prediction of the log($MTBF$)/log($MTTR$) for the last fold of the cross-validation procedure. The orange line depicts the mean and the orange shaded area the 95% confidence intervals. The blue dots mark the actual observed field-reliabilites. Note that the different converter types were ordered by the mean of the predictive distribution.(b),(d):Estimated feature weights for the parametric models. Figures (a),(b),(c),(d) are for the configuration with a reduced set of reliability indicators and figures (e),(f) for the second-order feature mapping. Note that the illustrations of the 629 second-order feature weights were omitted.



(a) (b)

Fig. 5. (a):Predictions of the log($MTBF$) with the final models for the test data-set. The orange line depicts the mean and the orange shaded area the 95% confidence intervals. The blue dots mark the actual observed field-reliabilites. Note that the different converter types were ordered by the mean of the predictive distribution.(b):Estimated feature weights for the predictive models.

Table 3. Obtained mean-squared-errors for the log($MTBF$) - a) Err_{CV} for the reference model, b) Err_{CV} for a reduced set of systems, c) Err_{CV} for a reduced set of reliability indicators, d) Err_{CV} for non-linear numeric feature mappings, and e) Err_{test} for the predictions of the test data-set. Comparison of a) and e) indicates if the method can be extended to future converter types.

	ARD	**BAR**	**GP**	**EN**	**SVR**
Err_{CV} **a)**	0.39±0.15	0.35±0.13	0.37±0.14	**0.34±0.12**	0.46±0.16
Err_{CV} **b)**	**0.90±0.79**	0.82±0.73	0.81±0.74	0.65±0.49	**0.64±0.50**
Err_{CV} **c)**	1.03±0.24	**1.00±0.19**	**1.00±0.19**	1.01±0.22	1.02±0.24
Err_{CV} **d)**	0.59±0.23	0.37±0.05	0.38±0.05	**0.32±0.05**	0.48±0.12
Err_{test} **e)**	**0.30**	0.33	0.32	**0.30**	0.38

Table 4. Obtained mean-squared-errors for the log($MTTR$) - a) Err_{CV} for the reference model, b) Err_{CV} for a reduced set of systems, c) Err_{CV} for a reduced set of reliability indicators, d) Err_{CV} for non-linear numeric feature mappings, and e) Err_{test} for the predictions of the test data-set.

	ARD	**BAR**	**GP**	**EN**	**SVR**
Err_{CV} **a)**	0.23±0.05	0.22±0.0.04	0.22±0.04	0.22±0.04	0.23±0.05
Err_{CV} **b)**	0.32±0.17	0.24±0.11	0.24±0.12	0.23±0.09	0.25±0.17
Err_{CV} **c)**	0.30±0.16	0.23±0.06	0.23±0.06	.28±0.11	0.29±0.16
Err_{CV} **d)**	3.12±4.83	0.23±0.02	0.23±0.03	0.22±0.02	0.34±0.06
Err_{test} **e)**	0.38	0.35	0.35	0.35	0.36

converters per converter type. This fact can lead to an increased reliability for future systems. However, explanations for this dependence are plentiful and a more detailed analysis will have to be carried out.
The method is capable of learning more detailed statistical models for the whole lifecycle of systems. This requires to collect more reliability indicators than were available in this work. However, the purpose of this work was to illustrate that even with very coarse high-level data a good predictive model can be trained. The selected Bayesian algorithms which learn sparse parametric models were especially fit for this purpose. It has to be pointed out that the approach is empirical and that causal relationships have to be identified or confirmed by further studies or expert judgment.

5 Conclusion and Outlook

An approach was presented to predict the field-reliability of complex electronic systems at an early development stage based on a statistical lifecycle model learned from data collected for similar operational systems. It was demonstrated that the field-reliability can be predicted accurately based on very few reliability indicators. Compared to existing methods this implies a reduced data collection effort and an integrated quantification of predictive uncertainty based on the granularity of the available information and the implicit randomness of the investigated processes. The results of such a study uncover reliability relevant factors which lead to improved system designs at very early stages of design.
Sparse Bayesian Regression methods are the key to efficiently learn accurate models. The confidence in field-reliability predictions is automatically quantified with respect to the available data and the randomness inherent in the problem. Future research can focus on more detailed and potentially incomplete data-sets. Based on that, further relevant processes for the field-reliability of systems may be uncovered.

References

1. Barnard, R.: What is wrong with reliability engineering? In: INCOSE International Symposium. vol. 18, pp. 357–365. Wiley Online Library (2008)
2. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
3. Blondel, M., Brucher, M., Buitinck, L., Cournapeau, D., Dawe, N., Dubourg, V., Duchesnay, E., la Tour, T.D., ander Fabisch, A., Fritsch, V., Ghosh, S., Gollonet, A.S., Gorgolewski, C.F., andre Gramfort, A., Grisel, O., Grobler, J., Halchenko, Y., Holt, B., Joly, A., Jones, T.R., Kastner, K., Kumar, M., Layton, R., Li, W., Losi, P., Louppe, G., Metzen, J.H., Michel, V., Millman, J., andreas Müller, Niculae, V., Nothman, J., andre Passos, A., Pedregosa, F., Prettenhofer, P., Bertr, Thirion, anderPlas, J.V., Varoquaux, N., Varoquaux, G., Weiss, R.: Scikit-learn user guide (2018), http://scikit-learn.org/stable/user_guide.html#

4. Cawley, G.C., Talbot, N.L.: On over-fitting in model selection and subsequent selection bias in performance evaluation. Journal of Machine Learning Research **11**(Jul), 2079–2107 (2010)
5. Denson, W.: The history of reliability prediction. IEEE Transactions on reliability **47**(3), SP321–SP328 (1998)
6. Elerath, J.G., Pecht, M.: IEEE 1413: A standard for reliability predictions. IEEE Transactions on Reliability **61**(1), 125–129 (2012)
7. Foucher, B., Boullie, J., Meslet, B., Das, D.: A review of reliability prediction methods for electronic devices. Microelectronics reliability **42**(8), 1155–1162 (2002)
8. Gauch, H.G.: Scientific method in practice. Cambridge University Press (2003)
9. Gullo, L.: In-service reliability assessment and top-down approach provides alternative reliability prediction method. In: Reliability and Maintainability Symposium, 1999. Proceedings. Annual. pp. 365–377. IEEE (1999)
10. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer-Verlag New York, Inc., New York (2001)
11. Johnson, B.G., Gullo, L.: Improvements in reliability assessment and prediction methodology. In: Reliability and Maintainability Symposium, 2000. Proceedings. Annual. pp. 181–187. IEEE (2000)
12. Jones, J., Hayes, J.: A comparison of electronic-reliability prediction models. IEEE Transactions on reliability **48**(2), 127–134 (1999)
13. Kapur, K.C., Pecht, M.: Reliability engineering. John Wiley & Sons (2014)
14. Leonard, C.T., Pecht, M.: How failure prediction methodology affects electronic equipment design. Quality and Reliability Engineering International **6**(4), 243–249 (1990)
15. MacKay, D.J.: Bayesian interpolation. Neural computation **4**(3), 415–447 (1992)
16. Miller, R., Green, J., Herrmann, D., Heer, D.: Assess your program for probability of success using the reliability scorecard tool. In: Reliability and Maintainability, 2004 Annual Symposium-RAMS. pp. 641–646. IEEE (2004)
17. O'Connor, P., Kleyner, A.: Practical reliability engineering. John Wiley & Sons (2012)
18. Pandian, G.P., Diganta, D., Chuan, L., Enrico, Z., Pecht, M.: A critique of reliability prediction techniques for avionics applications. Chinese Journal of Aeronautics (2017)
19. Pecht, M.G., Das, D., Ramakrishnan, A.: The IEEE standards on reliability program and reliability prediction methods for electronic equipment. Microelectronics Reliability **42**(9-11), 1259–1266 (2002)
20. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
21. Williams, C.K., Rasmussen, C.E.: Gaussian processes for machine learning. The MIT Press **2**(3), 4 (2006)
22. Womack, J.P., Womack, J.P., Jones, D.T., Roos, D.: Machine that changed the world. Simon and Schuster (1990)
23. Zio, E.: Reliability engineering: Old problems and new challenges. Reliability Engineering & System Safety **94**(2), 125–141 (2009)