Combining Prediction Methods for Hardware Asset Management

Alexander Wurl[1], Andreas Falkner[1], Alois Haselböck[1], Alexandra Mazak[2*], and Simon Sperl[1]
[1]*Siemens AG Österreich, Corporate Technology, Vienna, Austria*
[2]*TU Wien, Business Informatics Group, Austria*
{*alexander.wurl, andreas.a.falkner, alois.haselboeck, simon.sperl*}@*siemens.com, mazak@big.tuwien.ac.at*

Keywords: Predictive Asset Management, Obsolescence Management, Partial Least Squares Regression, Data Analytics

Abstract: As wrong estimations in hardware asset management may cause serious cost issues for industrial systems, a precise and efficient method for asset prediction is required. We present two complementary methods for forecasting the number of assets needed for systems with long lifetimes: (i) iteratively learning a well-fitted statistical model from installed systems to predict assets for planned systems, and - using this regression model - (ii) providing a stochastic model to estimate the number of asset replacements needed in the next years for existing and planned systems. Both methods were validated by experiments in the domain of rail automation.

1 Introduction

A crucial task in Hardware Asset Management[1] is the prediction of (i) the numbers of assets of various types for planned systems, and (ii) the numbers of assets in installed systems needed for replacement, either due to end of lifetime (preventive maintenance) or due to failure (corrective maintenance). This is especially important for companies that develop, engineer, and sell industrial and infrastructural systems with a long lifetime (e.g., in the range of decades) like power plants, factory equipment, or railway interlocking and safety systems.

Predicting the number of assets for future projects is important for sales departments to estimate the potential income for the next years and for bid groups to estimate the expected system costs. Simple linear regression models achieve only sub-optimal results when heterogeneous data sources, faulty data warehouse entries, or non-standard conditions in installed systems are involved. Therefore, we combine Partial Least Square Regression with an iterative algorithm that removes anomalies introduced by non-standard conditions and faulty data so that the learned model can provide optimal prediction results for future projects.

Usually, service contracts oblige a vendor to guarantee the functioning of the system for a given time period with a failure rate or system down-time lower than a specified value. This implies that all failing hardware assets (and often also those expected to fail in near future) must be replaced without delay in order to ensure continuous service. A precise prediction of the number of asset types needed in the next n years is essential for the company to calculate service prices and to fulfill the service obligations for replacing faulty components. Often, such predictions are made by experts relying on their experiences and instinct, including a certain safety margin. A more informed prediction is based on expectation values: Add the number of assets in existing systems and the expectation values for the numbers of assets for planned systems to get a basis for calculating the expectation values of asset replacements. This kind of prediction is simple, but it cannot provide information about the confidence interval of the prediction. A prediction far below the actual need may cause that assets are not available in time with all the annoying consequences for the vendor such as penitent fees and a bad reputation. A prediction far beyond the actual need will bind costs in unnecessary asset stocks. We provide a predictive model for the asset replacements, taking into account all necessary data from installed systems, project predictions and renewal necessities. The result is a probability function representing not only the predicted number of assets but also the uncertainty of the prediction.

Summarizing our approach, in the first step, we

*Alexandra Mazak is affiliated with the CDL-MINT at TU Wien.

[1]Please note, that in this article we use the term "asset" in the sense of physical components such as hardware modules or computers but not in the sense of financial instruments.

calculate a regression model capable to predict the number of assets for planned systems. Using this regression model, the second step provides a predictive model for the number of new assets and asset replacements needed in the next n years for installed and planned systems.

This paper is organized as follows: Section 2 presents the *asset regression model* for computing the assets of a planned system. Section 3 presents the *asset prediction model* for estimating the overall number of assets needed in the next n years, including assets for planned systems and renewal assets. Section 4 presents related work and Section 5 concludes the paper.

2 Asset Regression Model

In this section, we present an iterative learning method for predicting the number of assets of a given type needed for a planned system. This model is learned from the relation between feature and asset numbers in installed systems (in previous projects). Features are properties of a system or project that can be counted or measured by domain experts. For instance, in the domain of rail automation, features are the numbers of track switches, of signals of various types, of railway crossings, etc. In large systems, there are hundreds of different features and asset types, whereas the number of installed systems may be lower than hundred. In order to reduce efforts for the planers of future projects who need to supply the feature numbers, we need to find a minimal subset of features that still can predict the number of assets of a given type with sufficient accuracy.

The problem to find a model that predicts assets from a given amount of features relates to multivariate data analysis (MVA). One method amongst others in MVA that addresses similar complex settings is Partial Least Squares Regression (PLSR) (Wold et al., 2001). PLSR is a technique for collinear data that reduces the input variables (i.e., the features) to a smaller set of uncorrelated components and performs least squares regression on these components. This technique is especially useful when features are highly collinear, or when the data set reveals more features than observations (i.e., installed systems in previous projects). Furthermore, unlike ordinary multiple regression, PLSR does not suppose that the set of input features is resolved from ambiguities in data. Since PLSR performs least squares regression on components instead of original data, the original data can be measured with ambiguities. This makes the technique more robust to measurement uncertainty.

The main two reasons why standard linear regression does not perform well in our setting are the large number of input variables compared to the number of observations, and the fact that some observations (i.e., some previous projects) contain non-standard data disrupting the learned model.

2.1 Overview of PLSR

Now we give a brief overview of the PLSR approach (see, e.g., (Wold, 1980)). The input data set is available in the form of the following data matrix:

$$D = \begin{pmatrix} x_{1,1} & \cdots & x_{1,p} & y_{1,1} & \cdots & y_{1,d} \\ x_{2,1} & \cdots & x_{2,p} & y_{2,1} & \cdots & y_{2,d} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,p} & y_{n,1} & \cdots & y_{n,d} \end{pmatrix} \quad (1)$$

Each observation i $(1 \leq i \leq n)$ in D corresponds to an existing system and consists of input variable values $x_{i,1},\dots,x_{i,p}$ (features) and related output variable values $y_{i,1},\dots,y_{i,d}$ (assets). The task is to exploit the covariance relationship between input and output variables, to estimate the assets for planned systems where only the features are known. PLSR takes into account the underlying relationship, i.e., the latent structure, of features and assets. The matrices X and Y are decomposed into latent structures in an iterative process. The latent structure corresponding to the most variation of Y is extracted and explained by a latent structure of X that explains it the best.

The model of the population of D consists of a p-dimensional input variable vector X and a d-dimensional output variable vector Y. Instead of calculating the parameters β in the linear model

$$y_i = x_i^\top \beta + \varepsilon_i$$

in PLSR, we estimate the parameters γ in the so-called latent variable model

$$y_i = t_i^\top \gamma + \varepsilon_i \quad (2)$$

where the new coefficients γ are of dimension $q \leq p$, ε is an error (noise) term, i.e., the residuals, and the values t_i of the variables are put together in a $(n \times q)$ score matrix T. According to (Wold, 1980), the objective of PLSR is to estimate the scores T. Due to the dimension reduction, the regression of y on T should be more stable. The construction of T is sequentially performed for $k = 1,2,\dots,q$ through the PLS criteria

$$a_k = \underset{a}{\operatorname{argmax}}\, Cov(y, Xa)$$

under the constraints $\|a_k\| = 1$ and $Cov(Xa_k, Xa_j) = 0$ for $1 \leq j < k$. The vectors a_k with $k = 1,2,\dots,q$ are called *loadings*, and they are collected in the columns of the matrix A. The resulting score matrix is then

$$T = XA.$$

Once the loadings are computed, the regression model of (2) can be used as predictive regression model for assets.

2.2 Application of PLSR

One of the major challenges in our problem domain comes from potential outliers of asset quantities in the data set. Since the input data originate from heterogeneous data sources, anomalies may occur due to the complexity of merging data structures and due to non-standard conditions in some projects. Even if high data quality during the process of data integration is achieved by approaches such as described in (Wurl et al., 2017), anomaly detection including model validation is required during the training phase of the prediction model.

In model validation, the most common methods for training regression models are Cross-Validation (CV) and Bootstrapping. In our setting, CV is preferred because it tends to be less biased than Bootstrapping when selecting the model and it provides a realistic measurement of the prediction accuracy.

To ensure a stable prediction model, we make use of a double cross-validation strategy which is a process of two nested cross-validation loops. The inner loop is responsible for validating a stable prediction model, the outer loop measures the performance of prediction. This strategy is applied to similar approaches that have been described for optimizing the complexity of regression models in chemometrics (Filzmoser et al., 2009), for a binary classification problem in proteomics (Smit et al., 2008), and for a discrimination of human sweat samples (Dixon et al., 2007). Our approach is a formal, partly new combination of known procedures and methods, and has been implemented in a function for the programming environment R[2], as illustrated in Figure 1.

In the internal cross-validation loop CV1, we train our model with 80% of the data, using 10-fold cross-validation. In the external cross-validation CV2, we test our trained model with the 20% rest of the full data set.

2.2.1 Training

Our regression model potentially uses all input variables of all observations to predict a selected output variable. For each row $i = 1..n$ of the training set as subset of data matrix D (see Equation 1), the asset a ($a \in \{1 \ldots q\}$) is estimated by a function f_a on the input variables. The predicted values

$$\hat{y}_{i,a} = f_a(x_{i,1}, x_{i,2}, \ldots, x_{i,d})$$

should be as close as possible to the real asset values. ”As close as possible” is achieved by an iterative process of improving the prediction model step by step, which is computed by means of PLSR in combination with CV.

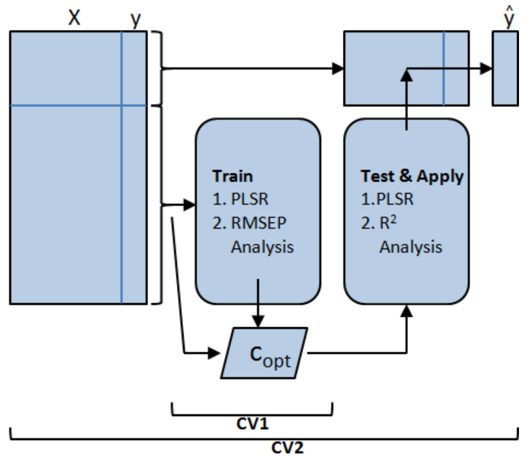


Figure 1: Double Cross-Validation: First, learn a model on training data set CV1. Second, test the model with the optimal number of components c_{opt} on test data set CV2.

In the inner CV loop, we employ 10-fold CV to train the model. By training, we aim at finding the number of components which adequately explains both predictors and response variances. This is calculated by the Root Mean Squared Error of Prediction (RMSEP) for asset a:

$$RMSEP = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\underbrace{(y_{i,a} - \hat{y}_{i,a})^2}_{\text{quadratic error}}}$$

The differences $y_{i,a} - \hat{y}_{i,a}$ are called residuals. RMSEP calculates the residual variation as a function of the number of components. The result presents for each component the estimated cross validation error and the cumulative percentage of variance explained. Aiming for an optimal number of components, the goal is to find the component with the lowest cross validation error. The minimum indicates the number of components with minimal prediction error and an optimum rank of the cumulative percentage of variance explained.

While calculating RMSEP, the estimated CV error usually decreases with an increasing number of components. In case of an error increase, it is likely that:

[2] www.r-project.org

1. Dependent and independent variables are not linearly related enough to each other.
2. The data set contains insufficient number of observations to reveal the relationship between input and output variables.
3. There is only one component needed to model the data.

In the outer loop, in addition to RMSEP we use the significance indicator R^2. R^2 is defined by

$$R^2 = 1 - \frac{\sum(y_{i,a} - \hat{y}_{i,a})^2}{\sum(y_{i,a} - \bar{y}_a)^2} \tag{3}$$

R^2 is a value between 0 and 1 and measures how close the test data are to the values predicted by the regression model. A low R^2 value (e.g., $R^2 < 0.9$) indicates that there are still outliers in the observations we used for training the model. These outliers in the data degraded the quality of our model, so we remove them and train a new model without them.

Algorithm 1 sketches the iterative process of training a model for asset prediction. Firstly, the data set is shuffled to obtain a fair distribution of observations. Secondly, a PLSR regression model is trained and validated through RMSEP. As a result, the model is trained again with the optimal number of components c_{opt}. Thirdly, this model is applied to the test data set and R^2 is computed. We use two criteria to decide whether our model is already good enough or not: (i) the RMSEP must monotonically decrease with the number of components used, and (ii) according to longitudinal studies R^2 must be at least 0.9 (Mooi et al., 2018). If the model is not yet good enough, we remove all observations from the data set that we categorized as outliers. After some experiments, the following outlier classification has turned out to be useful: Outliers are observations with residuals that are larger than 25% of the maximum asset value in the data set. After removing the noisy observations, learning starts again. We proceed with this learning cycle until our model meets the above mentioned quality criteria or no improvement could be achieved.

In the last step of Algorithm 1, the features that significantly contribute to a prediction are extracted according to their weights. Currently, this selection is done by a domain expert based on the weights of the features that dominate the prediction model.

Algorithm 1: Calculate Asset Regression Model

1. Shuffle rows in the data set to obtain a fair distribution of observations.
2. Train a PSLR regression model
   (a) Apply a 10-fold cross-validation to the training data set.
   (b) Compute RMSEP and evaluate optimal numbers of components c_{opt}.
   (c) Apply a 10-fold cross-validation to the training data set using c_{opt}.
3. Test model on test data set by computing R^2. If RMSEP does not monotonically decrease with the number of components, or if $R^2 < 0.9$
   (a) Identify outliers in observations and remove them.
   (b) Go to step 2.
4. Extract significant features to select the final prediction model.

2.3 Prediction

Our trained model is now

$$\hat{y}_a = f_a(X') \tag{4}$$

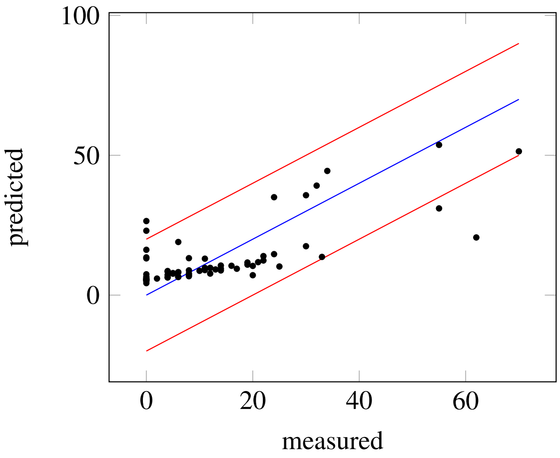
where $X' \subseteq \{x_1, \ldots, x_d\}$ is the set of significant features used for predicting assets of type a based on a model f_a. Applying this model to concrete feature values $\bar{X}'$ of features X' for a planned system will provide an expectation value for the number of assets a. In this sense, the output variable $\hat{y}_a$ could be seen as a random variable with a normal distribution - $\hat{y}_a \sim \mathcal{N}(\mu, \sigma^2)$ - where the mean value μ is $f_a(\bar{X}')$ and σ is the above described mean square error RMSEP. Therefore, a 95% confidence interval is given by $y_a = \hat{y}_a \pm 2 * RMSEP$ (Friedman et al., 2001). In the next subsection, we demonstrate this on an example.

2.4 Example and Experimental Results

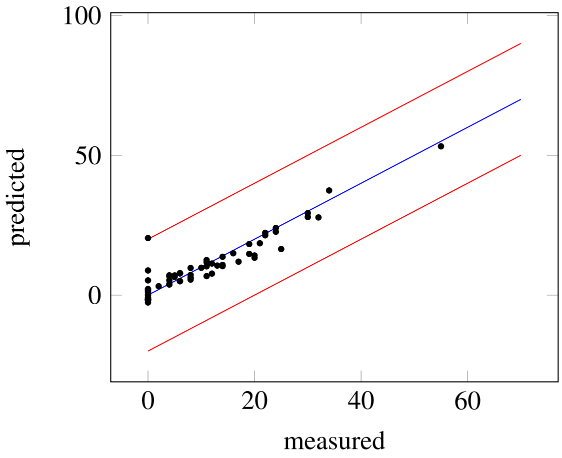
We tested validity of our method on a data set from the railway safety domain with data collected over about a decade. Our data set contained ca. 140 features (input variables), ca. 300 assets (potential output variables), and ca. 70 observations (installed systems). We chose the concrete asset type $A41$ - a hardware module that controls track switches - to demonstrate training of the regression model and prediction. We implemented this example and the regression training algorithm in R, making use of the plsr() function in the pls library[3].

Before starting any calculation, we preprocessed the data set by removing all assets except $A41$ and shuffled the data set in order to have a fair distribution

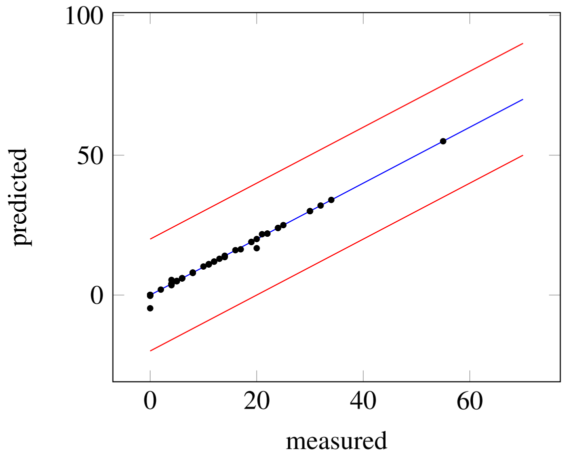
[3] http://mevik.net/work/software/pls.html



(a) Cycle 1: RMSEP=14.11, R^2=0.35



(b) Cycle 2: RMSEP=2.77, R^2=0.87



(c) Cycle 3: RMSEP=1.74, R^2=0.94

Figure 2: Three cycles for training a regression model for asset $A41$.

of the training data set and the test data set. Following Algorithm 1, we built the model by specifying $A41$ as the output variable and all features as input variables, and performed 10-fold cross-validation based on the NIPALS algorithm (Wold, 1966; Wold, 1980).

Figure 2 shows the three training cycles for $A41$. After the first cycle, four observations were identified as outliers and removed from the training set. After the second cycle, another two observations were removed. After the third cycle we ended up in a regression model of high precision.

The features are the input variables of the regression model. Their values must be provided by domain experts. To compute/count/measure these feature values (e.g., counting the signals of various types in a railway station, or measuring the distances between signals on the tracks) is often a time-consuming and expensive task. The reduction of the feature set to a small subset of significant ones with satisfying predictive power will save time and costs, especially in the usually very stressful phase of proposal preparation.

Therefore we identified the most significant input variables by analyzing the coefficients of the input variables in the model. Figure 3 shows the first 10 positive coefficients sorted by their values. It can easily be seen that feature F31 (corresponding to the number of track switches of the railway station) and feature F32 (corresponding to the number of track key-locks) have much more impact on asset $A41$ than all the other features. We learned a final model based on these two input variables only. This final model, as shown in Figure 4, is used for predicting the asset values in the test data set and does not show noticeable differences to the results of the last iteration (see Figure 2 (c)), although it is based on two features only.

We tested the described method on other asset types, e.g. on hardware modules for controlling track-side signals, and it shows a similar behavior – after a few cycles of outlier removal the learned regression models have high prediction quality.

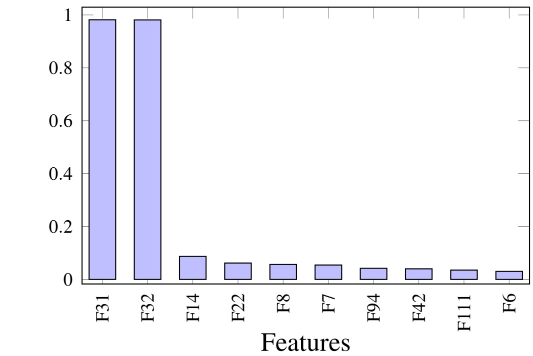


Figure 3: Importance Analysis of Features.

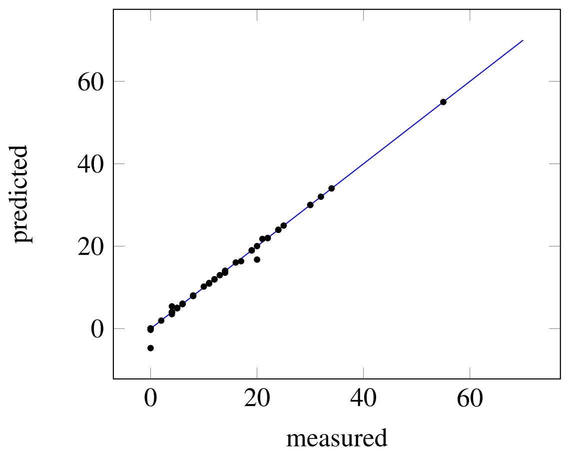


Figure 4: Final linear regression model for asset $A41$ based on features $F31$ and $F32$ only.

3 Asset Prediction Model

Building upon the regression model of the previous section, in this section we present a stochastic model for answering the following question: *How many assets of type A are needed in the next n years?* The input parameters of this problem are:

- An asset type A. Associated with an asset type is a failure rate, like MTBF (mean time between failure). MTBF is the expected time between failures of the asset. We only consider failures that cause the replacement of the asset.[4] The failures can be seen as random samples of a non-repairable population and the failure times follow a distribution with some probability density function (PDF).
- A scope $S = \{s_1, \ldots, s_n\}$ is a set of asset groups. Basically, each asset group corresponds to all assets of type A of an installed or planned system that must be taken into account for the forecast. Each member of an asset group must have the same installation date. As we will see later, this is important for computing the renewal numbers (older assets are more likely to fail than newer ones). Therefore, an installed system may be represented by more than one asset group: one group for all assets initially installed and still alive, and the other groups for the already necessary asset replacements.
  Each $s \in S$ has the following properties:
  - M_s is a random variable representing the number of assets in the asset group $s \in S$; $\Pr(M_s = n), n \in \mathbb{N}$, is the probability that s contains n elements.
  - $tbos_s$: Begin of service time of all assets in $s \in S$. At this point in time the assets has been or will be installed in the field.[5]
  - $teos_s$: End of service time of all assets in $s \in S$. This is the time where the service contract ends. After this point in time the assets need no longer be replaced when failed.
  - A probability $q_s \in [0, 1]$ representing the likelihood that the system containing $s \in S$ will be ordered. Trivially, for already installed assets $q_s = 1$.
- A point in time $t_{target} \in \mathbb{N}$ up till that the prediction should be made. All assets in the scope whose service times overlap the period $[t_{now}, t_{target}]$ are to be taken into account. The service time of interest for each asset group $s \in S$ is then
  $\tau_s = min(teos_s, t_{target}) - tbos_s$.

Please note that the computation of the forecast model is always done for assets of a given type A, so for the sake of simplifying the notation we omit to subscript all variables with A in this section.

The prediction of needed assets must take into account both renewal of already installed assets in case of non-repairable failures and new assets needed for planned systems. We will present a stochastic model that combines both cases. Algorithm 2 summarizes the steps how to compute such a prediction model. In step 1, the user has to provide an asset type, a

[4] A detailed differentiation of MTBF and MTTF (Mean Time To Failure) is beyond the scope of this paper.

[5] We use a simple representation of time: years started from some absolute zero time point. This simplifies arithmetic operations on time variables.

Algorithm 2: Total Asset Prediction

1. Input: (i) asset type A (ii) scope (iii) target time t_{target}
2. Collect asset groups of given asset type and given scope: $S = \{s_1, \ldots, s_n\}$
   (a) Existing assets from installed systems
   (b) Asset predictions for planned systems
3. Perform renewal asset analysis: $\hat{R}_s$
4. Compute estimator for each asset group: $\hat{N}_s$
5. Sum up asset group estimators: $\hat{N}_{total} = \sum_{s \in S} \hat{N}_s$

scope, and a target point in time. In step 2, the asset groups from the installed base (existing projects) and the sales database (future projects) are collected. Then a stochastic model that represents needed renewal assets is computed based on failure rates/failure distributions of the assets and the service time periods of the asset groups. In step 4, an estimator is computed for each asset group using project order probability, the probability of the number of assets in the group, and the renewal model. Finally, all these estimators are summed up to an overall asset estimator $\hat{N}_{total}$. We present the details in the next subsections.

3.1 Renewal Processes

To estimate the number of assets needed for replacement of failing assets we resort to the theory of renewal processes, see, e.g., (Grimmett and Stirzaker, 2001; Randomservices.org, 2017). In this subsection, we recap the main definitions of renewal processes and show how we apply it to predict the number of assets that must be replaced because of failures. A *renewal process* is a stochastic model for renewal events that occur randomly in time. Let X_1 be the random variable representing the time between 0 and the first necessary renewal of an asset. The random variables $X_2, X_3, \ldots$ are the subsequent renewals. These variables X_i are called inter-arrival times. In our case, their distribution is directly connected to the failure rates or MTBF of the asset type. So $(X_1, X_2, X_3, \ldots)$ is a sequence of independent, identically distributed random variables representing the time periods between renewals. X_i takes values from $[0, \infty)$, and $\Pr(X_i > 0) > 0$. Let $f_X(t)$ be the PDF and $F_X(t) = \Pr(X \leq t)$ be the distribution function of the variables X_i.

The random variable T_n for some number $n \in \mathbb{N}$ represents the so-called arrival time; $F_{T_n}(t) = \Pr(T_n \leq t)$ represents the probability of n renewals up to time t. T_n is simply the sum of the inter-arrival variables X_i:

$$T_n = \sum_{i=1}^{n} X_i$$

The PDF of T_n is therefore the convolution of its constituents:

$$f_{T_n} = f_X^{*n} = f_X * f_X * \ldots * f_X$$

Remark. To add two random variables one has to apply the convolution operator on the their PDFs. The convolution operator $*$ for two functions f and g is defined as

$$(f * g)(t) = \int f(t')g(t - t')\,\mathrm{d}t' \text{ , continuous case}$$

$$(f * g)(n) = \sum_{m=-\infty}^{\infty} f(m)g(n-m) \text{ , discrete case}$$

The expression f^{*n} stands for applying the convolution operator on a function f n times. For many probability distribution families, summing up two random variables and therefore computing the convolution of their PDFs is easy. For instance, the sum of two Poisson distributed variables with parameters λ_1 and λ_2 is: $Poi[\lambda_1] + Poi[\lambda_2] = Poi[\lambda_1 + \lambda 2]$. (End of remark.)

In the renewal process, the arrival time variables T_n are used to create a random variable N_t that counts the number of expected renewals in the time period $[0, t]$. It is defined in the following way:

$$N_t = |\{n \in \mathbb{N} : T_n \leq t\}| \text{ for } t \geq 0$$

The arrival time process T_n and the counting process N_t are kind of inverse to each other, where the probability distribution of N_t can be derived from the probability distribution of T_n. Thus, the theory of renewal processes gives us a tool for deriving the counting variable N_t from a given failure distribution X of an asset. For complicated distributions of X, the derivation of N_t could get elaborate and could reasonably be done by numeric methods only, but it is straight-forward for some prominent distribution families. The most important case is the Poisson process, where X has an exponential distribution with parameter λ. In this case, the n-th arrival time variable T_n has a Gamma distribution with shape parameter n and rate parameter λ, and the counting variable N_t has a Poisson distribution with parameter λt.

3.2 Prediction Model

In this subsection we describe how to combine the number of assets in each asset group M_s ($s \in S$), the above described renewal counting variable N_{τ_s}, and the probability q_s representing the likelihood that the assets will be ordered to an estimator, i.e., a probability mass function[6](PMF), reflecting the number of needed assets in a given time period.

Definition 1 (Renewal Estimator $\hat{R}_s$)**.** *Let M_s be a random variable representing the number of assets in the asset group $s \in S$. Let N_{τ_s} be the renewal counter for the service time period τ_s of asset group s.The* renewal estimator *$\hat{R}_s$ is a random variable, where $Pr(\hat{R}_s = n), n \in \mathbb{N}$, is the probability that exactly n renewal assets are needed in total for the asset group $s \in S$. It's PMF $f_{\hat{R}_s}$ is defined in the following way:*

$$f_{\hat{R}_s}(n) := \sum_{k=0}^{\infty} f_{M_s}(k) f_{N_{\tau_s}}^{*k}(n) \tag{5}$$

Definition 2 (Asset Group Estimator $\hat{N}_s$)**.** *Let M_s be a random variable representing the number of assets in the asset group $s \in S$. Let q_s be the probability that the project containing the assets of s are ordered at all. Let $\hat{R}_s$ be the renewal estimator as defined above.$\hat{N}_s$ is a random variable with $Pr(\hat{N}_s = n), n \in \mathbb{N}$, representing the probability that the number of assets needed for s is exactly n. It is the sum of the number of assets and the number of renewals scaled by the order probability. Its PMF is:*

$$f_{\hat{N}_s}(n) := q_s(f_{M_s}(n) * f_{\hat{R}_s}(n)) + (1 - q_s)\delta_0(n) \tag{6}$$

Remark. The delta function $\delta_k(x)$ (also called unit impulse) is 1 at $x = k$ and otherwise 0. We use the delta function here to express certainty of zero assets in the case that the project will not be ordered. As we use later, the convolution with the delta function can be used for shifting: $f(x) * \delta_k(x) = f(x - k)$. (End of remark.)

Definition 3 (Total Asset Estimator $\hat{N}_{total}$)**.** *The total asset estimator $\hat{N}_{total}$ is a random variable with $Pr(\hat{N}_{total} = n), n \in \mathbb{N}$, representing the probability that the total number of assets needed for all asset groups $s \in S$ until t_{target} is exactly n. It is the sum of the asset group estimators $\hat{N}_s$.*

$$\hat{N}_{total} := \sum_{s \in S}^{n} \hat{N}_s \tag{7}$$

Its PMF is the convolution of the PMFs of the asset group estimators $\hat{N}_s$:

$$f_{\hat{N}_{total}}(n) := \ast_{s \in S} f_{\hat{N}_s}(n) \tag{8}$$

The probability distribution $f_{\hat{N}_{total}}$ can now be used for calculating its expectation value, its variance or standard deviation, but also the number of needed assets with a guaranteed probability that the number will be high enough, i.e., compute the smallest n with $\Pr(\hat{N}_{total} <= n)$ is greater or equal some given probability, like 0.75 or 0.95, depending on the certainty the forecast should provide.

The design of the variables in this framework is both suited for installed and planned systems. For a planned system, the probability that the system will be ordered is an information provided by sales experts. The estimation of how many assets will be needed can be predicted by the regression model described in Section 2. In this case, the estimator M_s corresponds to a random variable derived from the regression model $\hat{y} = f(\bar{x})$, where $\bar{x}$ is the feature vector of the planned system. See Equation 4 in Section 2.3.

For asset groups of installed systems, the order probability q_s is simply 1, and the estimator variable M_s for the asset count is the delta function $\delta_k(n)$ with k being the actual number of installed assets. In this case the asset group estimator $\hat{N}_s$ is simply the shifted renewal estimator $\hat{R}_s$ with PDF $f_{\hat{R}_s}(n - k)$. It should be noted that $\hat{N}_{total}$ not only contains predicted assets, but also all already installed assets. These could be simply subtracted from $\hat{N}_{total}$, if only the number of assets to be ordered in the future are needed.

3.3 Example

Figure 5 shows a small example that shall demonstrate the combination of statistical methods of our asset prediction framework.

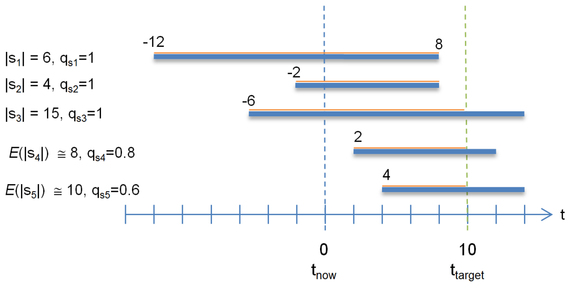
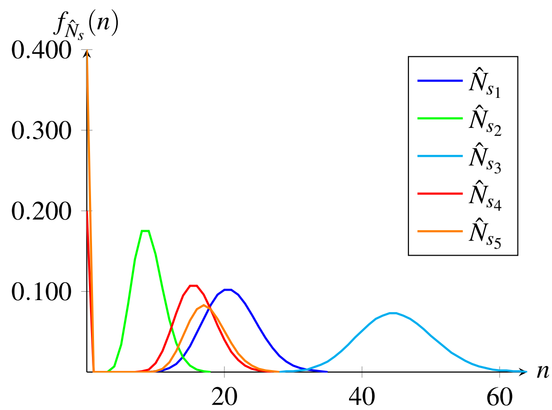


Figure 5: Example of an asset prediction problem, containing 3 asset groups of existing projects (s_1, s_2, s_3) and 2 future projects with asset groups s_4 and s_5.

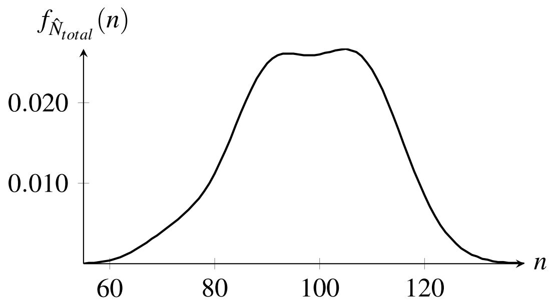
We use an exponential distribution of the failure rates of the assets with $\lambda = 0.125$ (i.e., 0.125 failures per year expected), corresponding to a MTBF of 8 years. So the renewal counting variables N_τ are Poisson distributed: $N_\tau \sim Poi(\lambda\tau)$. The number of assets of the existing systems 1 to 3 are 6, 4, and 15, so $M_{s_1} \sim \delta_6$, $M_{s_2} \sim \delta_4$, and $M_{s_3} \sim \delta_{15}$. The number estimators M_{s_4} and M_{s_5} of the two planned systems are:

[6]A probability mass function (PMF) is the discrete counterpart of a probability distribution function (PDF). A PMF $f(n)$ corresponding to a random variable X is $\Pr(X = n), n \in \mathbb{N}$.

$M_{s_4}: \Pr(M=7)=0.2, \Pr(M=8)=0.6, \Pr(M=9)=0.2$; $M_{s_5}: \Pr(M=8)=0.05, \Pr(M=9)=0.1, \Pr(M=10)=0.7$, Pr(M=11)=0.1, $\Pr(M=12)=0.05$. The resulting asset group estimators and the total asset estimator are depicted in Fig. 6. Some interesting results are shown in Table 1. The first row shows the resulting expectation values of the number of assets. The other rows in Table 1 show the variances and standard deviations of our estimator probability functions, along with 3 examples of asset estimations with specified likelihood, i.e., $F^{-1}(p)$ stands for the number of assets n with $\Pr(F \leq n) >= p$. While usually only this expectation value is used for asset prediction, our approach provides valuable additional information, like the standard deviation, and the possibility to find an asset number estimation with high reliability, like $F^{-1}(0.95)$.



(a) Asset Group Estimators $\hat{N}_s$



(b) Total Asset Estimator $\hat{N}_{total}$

Figure 6: Probability mass functions of predicted assets for asset groups and total asset estimator for the example scenario depicted in Fig. 5.

Figure 7 shows the resulting total asset estimator of a real world example. We applied our method to the data set used in Section 2.4 and computed a predictor for the number of assets of type $A41$ for the next 10 years. About 40 installed systems with 580 instances of asset $A41$ and 7 future systems have been taken into account. Subtracting the already installed 580 assets, we get an expectation value of 395 additional assets, covering the assets for the 7 new systems and asset renewals. A prediction with 95% likelihood results in 438 new assets.

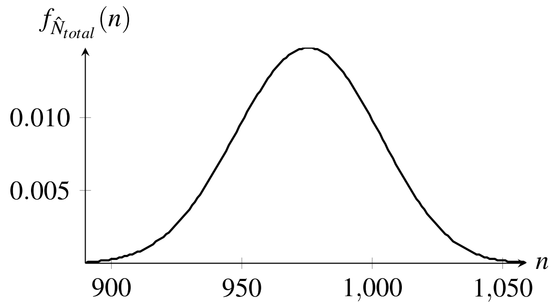


Figure 7: Probability mass function for an estimator of asset $A41$. 580 assets are already installed. The total expectation value $\mathbb{E}$ is 975 (corresponds to 395 new assets), the standard deviation σ is 26.30, $F^{-1}(0.75) = 993$ (corresponds to 413 new assets), $F^{-1}(0.95) = 1018$ (corresponds to 438 new assets).

4 Related Work

In industry, asset management is defined as the management of physical, as opposed to financial, assets (Amadi-Echendu et al., 2010). Managing assets may comprise a broad range of different potentially overlapping objectives such as planning, manufacturing, and service. Depending on the scope of asset management, the data included determine the possibilities of predictive data analytics (Li et al., 2015). An essential part of asset management, especially in manufacturing, is to estimate the condition (health) of an asset.

In the last decade, an engineering discipline, called *prognostics*, has evolved aiming to predict the remaining useful life (RUL), i.e., the time at which a system or a component (asset) will no longer perform its intended function (Vachtsevanos et al., 2006). Since the reason for non-performance is most often a failure, data-driven prognostics approaches consist of modeling the health of assets and learning the RUL from available data (Mosallam et al., 2015; Mosallam et al., 2016). (Bagheri et al., 2015) present a stochastic method for data-driven RUL prediction of a complex engineering system. Based on the health value of assets, logistic regression and the assessment output is used in a Monte Carlo simulation to estimate the remaining useful life of the desired system. (Lee et al., 2017) calculate the weighted RUL of assets to optimize the productivity in cyber-physical manufactur-

Table 1: Some resulting probability values for the example scenario depicted in Fig. 5.

	s_1	s_2	s_3	s_4	s_5	**Total**
$\mathbb{E}$	21	9	45	13	11	99
var	14.91	4.97	29.68	47.95	78.47	175.58
σ	3.86	2.23	5.45	6.92	8.86	13.25
$F^{-1}(0.50)$	21	9	45	15	15	99
$F^{-1}(0.75)$	24	10	49	17	18	108
$F^{-1}(0.95)$	28	13	54	21	22	119

ing system by applying Principal Component Analysis (PCA) in combination with Restricted Boltzmann Machine (RBM).

Other predictive methods aim for predicting the *obsolescence* of assets based on sales data. (Sandborn et al., 2007) propose a data-mining approach including linear regression to estimate when an asset becomes obsolete. Based on this approach, (Ma and Kim, 2017) use time series models instead of linear regression models and state that prediction over years may be inaccurate. (Jenab et al., 2014) examine the detection of obsolescence in a railway signaling system with a Markovian model and mention that forecasting obsolescence of assets may be inexact. (Thaduri et al., 2015) draft the potential application of predictive asset management in the railway domain by incorporating heterogeneous data sources such as construction and sales data. One of the main challenges is predictive asset management for long-term maintenance which coincides with our context.

One possible starting point to address the prediction of *long-term* asset management is formulating the product life cycle with evolutionary parametric drivers that describe an asset type whose performance or characteristics evolve over time (Solomon et al., 2000). Since evolutionary parametric drivers cannot be found in our database, procurement life modeling may be used (Sandborn et al., 2011). The most prominent model of the mean procurement life, which is analogous to the mean-time-to-failure (MTTF), is the bathtub curve (Klutke et al., 2003), usually modeled by Weibull distributions. This model represents three regimes in the lifetime of a hardware module: the first phase shows high but decreasing failure rates ("infant mortality"), the main, middle phase shows low, constant failure rates, and the third phase represents "wear out failures" and therefore increasing failure rates.

A prominent model in *preventive maintenance* (see, e.g., (Gertsbakh, 2013)) represents the replacement of an asset after a constant time before the failure probability gets too high. This can be modeled by a distribution function F that consists of two parts: the first part is some "conventional" distribution function, but from a defined replacement time t_r on, the probability of replacement is just 1.

Our input data is taken from the installed base (database of currently installed systems) and the sales database containing forecasts of expected future projects (planned systems). Unlike previous predictive obsolescence approaches where the main goal is to estimate the date when particular assets become obsolete, the combination of our data sets enables us to represent not only the numbers of renewal assets needed for n years but also the uncertainty of the estimation for long-term maintenance.

5 Conclusion and Future Work

In this paper, we proposed a predictive asset management method for hardware assets. The proposed method consists of two phases: In the first phase, a regression model is learned from installed systems to predict assets for planned systems. In the second phase, a stochastic model is used for summing up all assets needed in the next n years for existing and also future projects, taking renewals of failing components into account. In experiments we validated our method in the domain of railway safety systems.

There are several ideas for future work: Currently, when calculating an asset regression model we apply PLSR in an iterative way. In future work, we aim at a broader evaluation by comparing additional methods such as *Sparse partial robust M-regression* (Hoffmann et al., 2015), or *Robust and sparse estimation methods for high dimensional linear and logistic regression* (Kurnaz et al., 2017).

In our approach we train the model for each asset type separately. This means that the set of features for each regression model is minimized for each asset separately. An important problem for future work is to find a minimal subset of features that is significant for predicting all assets. Furthermore, it would be interesting to take costs of feature measurement into account. E.g., in the railway domain it is easier to count signals and track switches than insulated rail joints of a railroad region. An optimal subset of features is one with minimal total measurement costs.

Presently, we use a given MTBF per asset type in our method. We assume that prediction accuracy can be further improved by applying advanced prognostics techniques for customizing the MTBF to the respective system context.

Although we tested our method only on data from rail automation, we suppose that applying it to hardware components in other domains will produce similar results of prediction. This needs to be evaluated in future work.

Acknowledgements

This work is funded by the Austrian Research Promotion Agency (FFG) under grant 852658 (CODA). We thank Walter Obenaus (Siemens Rail Automation) for supplying us with test data.

REFERENCES

Amadi-Echendu, J. E., Willett, R., Brown, K., Hope, T., Lee, J., Mathew, J., Vyas, N., and Yang, B.-S. (2010). *What Is Engineering Asset Management?*, pages 3–16. Springer London, London.

Bagheri, B., Siegel, D., Zhao, W., and Lee, J. (2015). A stochastic asset life prediction method for large fleet datasets in big data environment. In *ASME 2015 International Mechanical Engineering Congress and Exposition*, pages V014T06A010–V014T06A010. American Society of Mechanical Engineers.

Dixon, S. J., Xu, Y., Brereton, R. G., Soini, H. A., Novotny, M. V., Oberzaucher, E., Grammer, K., and Penn, D. J. (2007). Pattern recognition of gas chromatography mass spectrometry of human volatiles in sweat to distinguish the sex of subjects and determine potential discriminatory marker peaks. *Chemometrics and Intelligent Laboratory Systems*, 87(2):161–172.

Filzmoser, P., Liebmann, B., and Varmuza, K. (2009). Repeated double cross validation. *Journal of Chemometrics*, 23(4):160–171.

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics New York.

Gertsbakh, I. (2013). *Reliability theory: with applications to preventive maintenance*. Springer.

Grimmett, G. and Stirzaker, D. (2001). *Probability and random processes*. Oxford university press.

Hoffmann, I., Serneels, S., Filzmoser, P., and Croux, C. (2015). Sparse partial robust m regression. *Chemometrics and Intelligent Laboratory Systems*, 149:50–59.

Jenab, K., Noori, K., and Weinsier, P. D. (2014). Obsolescence management in rail signalling systems: concept and markovian modelling. *International Journal of Productivity and Quality Management*, 14(1):21–35.

Klutke, G., Kiessler, P. C., and Wortman, M. A. (2003). A critical look at the bathtub curve. *IEEE Trans. Reliability*, 52(1):125–129.

Kurnaz, F. S., Hoffmann, I., and Filzmoser, P. (2017). Robust and sparse estimation methods for high-dimensional linear and logistic regression. *Chemometrics and Intelligent Laboratory Systems*.

Lee, J., Jin, C., and Bagheri, B. (2017). Cyber physical systems for predictive production systems. *Production Engineering*, 11(2):155–165.

Li, J., Tao, F., Cheng, Y., and Zhao, L. (2015). Big data in product lifecycle management. *The International Journal of Advanced Manufacturing Technology*, 81(1-4):667–684.

Ma, J. and Kim, N. (2017). Electronic part obsolescence forecasting based on time series modeling. *International Journal of Precision Engineering and Manufacturing*, 18(5):771–777.

Mooi, E., Sarstedt, M., and Mooi-Reci, I. (2018). *Market Research*. Springer.

Mosallam, A., Medjaher, K., and Zerhouni, N. (2015). Component based data-driven prognostics for complex systems: Methodology and applications. In *Reliability Systems Engineering (ICRSE), 2015 First International Conference on*, pages 1–7. IEEE.

Mosallam, A., Medjaher, K., and Zerhouni, N. (2016). Data-driven prognostic method based on bayesian approaches for direct remaining useful life prediction. *Journal of Intelligent Manufacturing*, 27(5):1037–1048.

Randomservices.org (2017). Renewal processes. `http://www.randomservices.org/random/renewal/index.html` [Online; accessed 3-February-2018].

Sandborn, P., Prabhakar, V., and Ahmad, O. (2011). Forecasting electronic part procurement lifetimes to enable the management of DMSMS obsolescence. *Microelectronics Reliability*, 51(2):392–399.

Sandborn, P. A., Mauro, F., and Knox, R. (2007). A data mining based approach to electronic part obsolescence forecasting. *IEEE Transactions on Components and Packaging Technologies*, 30(3):397–401.

Smit, S., Hoefsloot, H. C., and Smilde, A. K. (2008). Statistical data processing in clinical proteomics. *Journal of Chromatography B*, 866(1-2):77–88.

Solomon, R., Sandborn, P. A., and Pecht, M. G. (2000). Electronic part life cycle concepts and obsolescence forecasting. *IEEE Transactions on Components and Packaging Technologies*, 23(4):707–717.

Thaduri, A., Galar, D., and Kumar, U. (2015). Railway assets: A potential domain for big data analytics. *Procedia Computer Science*, 53:457–467.

Vachtsevanos, G. J., Lewis, F., Hess, A., and Wu, B. (2006). *Intelligent fault diagnosis and prognosis for engineering systems*. Wiley Online Library.

Wold, H. (1966). Nonlinear Estimation by Iterative Least Squares Procedures in: David, FN (Hrsg.), Festschrift for J. *Neyman: Research Papers in Statistics, London*.

Wold, H. (1980). Model construction and evaluation when theoretical knowledge is scarce: Theory and application of partial least squares. In *Evaluation of econometric models*, pages 47–74. Elsevier.

Wold, S., Sjöström, M., and Eriksson, L. (2001). Pls-regression: a basic tool of chemometrics. *Chemometrics and intelligent laboratory systems*, 58(2):109–130.

Wurl, A., Falkner, A., Haselböck, A., and Mazak, A. (2017). Using signifiers for data integration in rail automation. In *Proceedings of the 6th International Conference on Data Science, Technology and Applications*, volume 1, pages 172–179.