Using Artificial Intelligence Algorithms to Identify Factors of Methane Leaks from Gas Transmission Assets

Amel Belounnas, Florent Brissaud and Elodie Rousset

July 23, 2022

Using artificial intelligence algorithms to identify factors of methane leaks from gas transmission assets

Amel Belounnas

RICE, GRTgaz, France. E-mail: amel.belounnas@grtgaz.com

Florent Brissaud

RICE, GRTgaz, France. E-mail: florent.brissaud@grtgaz.com

Elodie Rousset

RICE, GRTgaz, France. E-mail: elodie.rousset@grtgaz.com

GRTgaz has launched a major program for improving the assessment and the reduction of methane emissions due to leaks from its gas transmission network and facilities. The industrial assets of GRTgaz notably include thousands of gas delivery units, each containing pipes, gas pressure regulator(s), filter(s), shutoff valve(s), safety relief valve(s)... The leak detection campaign on the gas delivery units then requires significant resources and a lot of time. Targeting the assets that are the most likely to leak is therefore an important challenge for improving the campaign efficiency, moving forward the implementation of corrective measures, and reducing the methane emissions.

This work aims to explore XGBoost (Extreme Gradient Boosting) Cox survival regression model, associated with SHAP (SHapley Additive exPlanations) method, and Bayesian Networks using BayesiaLab software, to identify and explain the effect of different features on reliability of natural gas transmission assets, based on field feedback data.

Keywords: XGBoost, Bayesian Networks, Cox survival, Asset management, Machine learning, gas leaks

1. Introduction

GRTgaz is a major operator in the high-pressure gas transportation sector. The company has a public service mission aimed at guaranteeing the continuity of gas transmission, and a genuine commitment to promoting renewable gas and to the energy transition in the territories.

GRTgaz is committed to reduce its environmental impact and, as a priority, its direct emissions. It has launched a major program for improving the assessment and the reduction of methane emissions due to leaks from its gas transmission network and facilities. In parallel, GRTgaz joined the OilGas Methane Partnership (OGMP 2.0), a parternship launched by the United Nations Environment Program (UNEP) with the support of the European Commission, which provide a framework for methane emissions reporting and for their efforts to reduce them. The European Commission is working on a new regulation which will provide standards about measurement, reporting and immediate reduction of emission.
GRTgaz has thousands of asset which includes gas delivery units, each containing several pipes, gas pressure regulator(s), filter(s), shutoff valve(s), safety relief valve(s), manual valves... The leak detection campaign on the gas delivery units then requires significant resources and a lot of time.
To improve the campaign efficiency, field data processing methods based on AI (Artificial Intelligence) are investigated for analysing ”methane emissions” influencing factors, that is, internal and external parts of an asset acting on the rate of gas leaks Brissaud et al. (2010).

Available field data are: knowledge about the industrial assets, maintenance activities performed for repairing the “external leak” failure modes, and up-to-date results of the methane leak detection campaign. A major issue is the characteristics of the input data, notably regarding the assets. In fact, these factors can be binary (i.e.

Proceedings of the 32nd European Safety and Reliability Conference.
Edited by Maria Chiara Leva, Edoardo Patelli, Luca Podofillini and Simon Wilson
ISBN: 981-973-0000-00-0 :: doi: 10.3850/981-973-0000-00-0_output

yes or no), numerical (e.g. size, pressure…) or textual (e.g. manufacturer, position…). In addition, most of the factors are not known for all the assets (i.e. incomplete data) and erroneous values are inevitable (bad filling of the database), which limits the application of certain statistical approaches Brissaud et al. (2019). Considering only the assets for which the factors are fully known and confident would eliminate a larger part of the park, making the reduction of methane emission inefficient. It is therefore required that the proposed methods can deal efficiently with these constraints.

2. Methodology

2.1. *Data and target variables*

The used data represents around 30,000 items (regulators, filters, safety relief valves, shutoff valves and pipes), each of them has up to 30 given features, an observation period from 1 to 15 years where a certain number of failures (notably for the ”external leakage” failure mode) is observed. Each type of these items is analysed separately with two models: XGBoost Cox survival and Bayesian Network. In XGBoost Cox survival, the target variable is the ”$mean\,time\,to\,failure$” (MTTF) which is the observation time divided by the number of leaks. If an item has 0 observed leak, then the target variable is ”right truncated” (or censored) and equal to the observation time. At the opposite, the Bayesian models do not handle censored data. Therefore, in the supervised Bayesian model, the target is the ”$leak\ rate$”, which is the total number of leaks divided by the observation time. Moreover, this ”leak rate” is normalized to handle the heterogeneous scales of values.

2.2. *XGBoost Cox Survival*

Cox proportional hazards model Cox (1972), is essentially a regression model commonly used in medical research for investigating the association between the survival time of patients and one or more predictor variables. The purpose of the model is to evaluate simultaneously the effect of several factors on survival time. In other words, it allows us to examine how specified factors influence the rate of a particular event happening (e.g. death, failure...) at a particular point in time. This rate is commonly referred to as the hazard rate (in our case, this is a ”leakage rate”). Predictor variables (or features) are usually termed covariates in the survival-analysis literature.

The Cox model is expressed by the hazard function denoted by $h(t)$. Briefly, the hazard function can be interpreted as the risk of ”failing” (in our case, the risk of leaking) at time t. It can be expressed as follow:

$$h(t) = h_0(t) \times exp(b_1x_1 + b_2x_2 + ... + b_px_p) \quad (1)$$

Where:

- t represents the survival time (in our case, the ”time to failure”, where the failure is a leakage)
- $h(t)$ is the hazard function determined by a set of p covariates $(x_1, x_2, ...x_p)$
- the coefficients $(b_1, b_2, ..., b_p)$ measure the impact of covariates.
- $h_0(t)$ is called the baseline hazard. It corresponds to the value of the hazard at time t if all the x_i are equal to zero.

XGBoost[a] is a scalable machine learning system for tree boosting, available as an open source package. Its impact has been widely recognized in a number of machine learning and data mining challenges Chen and Guestrin (2016). XGBoost handles efficiently the missing data, hence no imputation method is needed. It has many learning objectives, here we use ”Cox survival regression”. For right censored survival time data (i.e. when no leak is observed during the observation time), the MTTF is ”coded” by a negative value. The predictions are the $\mathbf{Risk\ score}$ which is the hazard ratio: $HR = exp(marginal\ prediction)$ in the proportional hazard function $h(t) = h_0(t) \times HR$. The $\mathbf{Concordance\ Index}$ or $\mathbf{C-index}$ is used as an evaluation metric for this model. It is a generalization of the area under the ROC curve (AUC) that can take into account censored data. It represents the global assessment of the

[a] https://xgboost.ai/

model discrimination power which is the model's ability to correctly provide a reliable ranking of the survival times based on the individual risk scores Uno et al. (2011).The $\mathbf{C-index}$ is not implemented in XGBoost, it is computed with the following formula:

$$\text{C-index} = \frac{\sum_{i,j} 1_{T_j<T_i} \cdot 1_{\eta_j>\eta_i} \cdot \delta_j}{\sum_{i,j} 1_{T_j<T_i} \cdot \delta_j} \quad (2)$$

With:

- η_i, the risk score of a unit i
- $1_{T_j<T_i} = 1$ if $T_j < T_i$ else 0
- $1_{\eta_j>\eta_i} = 1$ if $\eta_j > \eta_i$ else 0
- δ_j = 0,1 is a binary event indicator

Similarly to the AUC, C-index = 1 corresponds to the best model prediction, and C-index = 0.5 represents a random prediction.

For the interpretability of the results, a Shapley additive explanations (SHAP)[b] library is used as a complement of XGBoost Scott and Su-In (2017). Starting from a mean value, the SHAP value represents the positive or negative effect of each factor (given the value of this factor). Dedicated graphs show the average impact of each factor on the target, and the impact of each value of the factors.

2.3. *Bayesian Network*

Bayesian networks Jouffe and Conrady (2020) are implemented by BayesiaLab, a commercial software tool. The 10th version, issued in 2021, is used for the present study. Both discrete (including binary and textual) and continuous values are handled. However, continuous values need to be discretized. A genetic algorithm is used to automatically perform this task (nine other algorithms are also available, plus a manual mode). Moreover, when discrete values are numerous for a given factor, they need to be aggregated into smaller numbers of "sets" (e.g. about five values, depending on the quantity of data). Missing values are inferred by a structural Expectation Maximization (EM) algorithm, the set of observations is supplemented with one weighted observation per combination of the states of the jointly unobserved variables. Each weight equals the posterior joint probability of the corresponding state combination. (other algorithms, including entropy-based, static or dynamic imputations are also available).

First, unsupervised structural learning is performed, using the maximum spanning tree algorithm. It is by far the quickest Unsupervised Structural Learning Algorithm. It only relies on two passes: The first pass consists of computing the a priory weight of all binary relationships between all variables. The second pass constructs the Maximum Weight Spanning Tree of those relationships.

Five other algorithms are also available. This tool is very convenient for investigating the relationships between factors. Each factor is depicted by a node and it is linked to the "most dependant" other factors by arrows. An "automatic mapping" allows drawing a planar network where the size of each node is proportional to its force (i.e. degree of dependency with linked nodes). In addition, the variable clustering can group the factors in "classes", which constitutes kinds of "families" where factors are strongly dependant.

Second, supervised learning is performed, using the naive Bayes algorithm (seven other algorithms are also available). A Naive Bayes network has a predefined structure in which the "target node" is the parent of all the other nodes. This structure implies that the target node is the cause of all the other nodes and that the knowledge of its value makes each node independent of the others. Meaning that this approach models the relationship of each factor with a selected "target" which is, in our case, the $\mathbf{leak\ rate}$. Then, it is possible to depict the total or direct (i.e. marginal) effect of the factors on the "target", using various illustrations: networks, curves, histograms, graphs... Finally, the inference is used to estimate the leak rate based on the factors. Despite these strong assumptions, which are unjustifiable in most cases, the small number of probabilities to estimate makes this structure very robust, with a short learning time.

[b] https://shap.readthedocs.io/en/latest/index.html

3. Results and Analysis

3.1. *XGBoost Cox Survival*

In this section we present the main results only for gas pressure regulators. Starting with XGBoost Cox survival, Fig.1 represents the risk score of each item as function of the MTTF. Negative values represent right censored data (items that did not leak during the observation time), the risk score of these items is close to zero as expected. On the other hand, the risk score of the items with MTTF > 0 is an exponential inversely proportional to the MTTF. The $\mathbf{C} - \mathbf{index}$ of this model is 0.93.

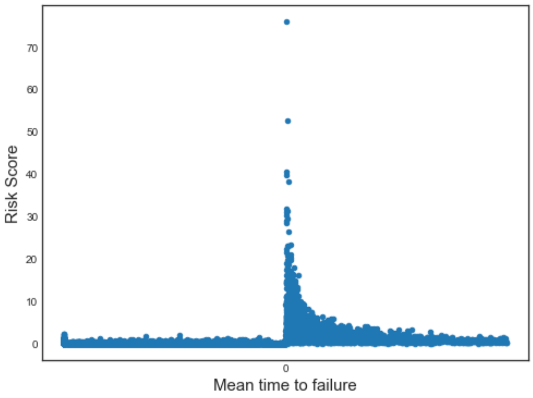

Fig. 1. Risk factor calculated by XGBoost for each item as function of the MTTF (negative values represent censored data).

We use SHAP to better understand the results of the model. Fig.2 shows the global feature importance plot, where the global importance of each feature is taken to be the mean absolute value for that feature over all the given samples. Note that features starting with ”S_” are the ones relating to the ”site” where the asset is installed, and not to the asset itself. For the first four most important features we show the individual SHAP dependence plots in the figures below. They illustrate the effect that a single feature has on the predictions made by the model, every dot is an item, and the vertical dispersion shows that the same value for a feature can have a different impact on the model’s output for different items. This means there are non-linear interaction effects in the model between that feature and the other features. As mentioned before, XGBoost handles missing values, they are represented in SHAP by grey ticks attached to the y-axis. Fig.3 shows how the model output varies by the regulator position value. The position R1L1 represents the highest risk, this is expected since the first regulators (R1) of the first line (L1) are the ones that operate the most.

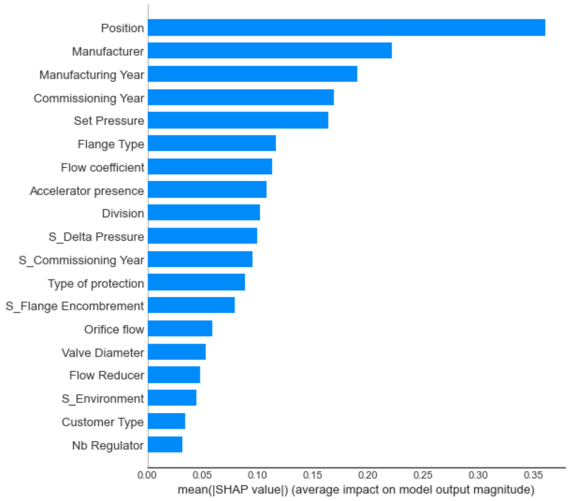

Fig. 2. Feature importance SHAP bar plot.

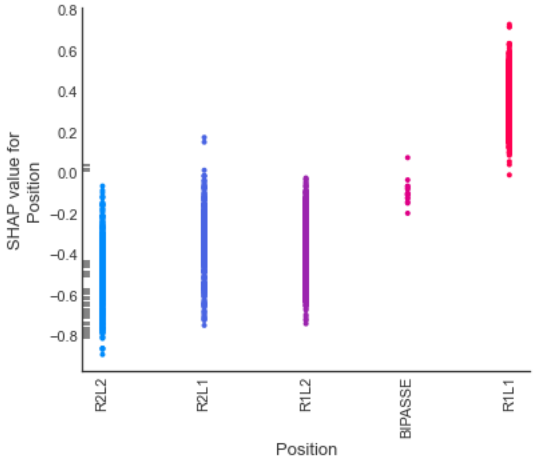

Fig. 3. SHAP dependence plot, showing how the model output varies by the position value.

Fig.4 shows how the model output varies by the manufacturer value. It is observed that man-

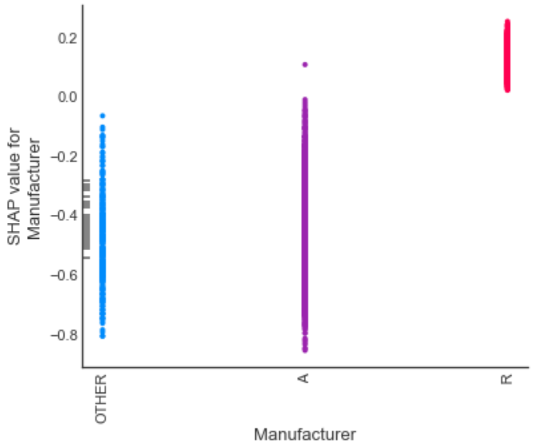

Fig. 4. SHAP dependence plot, showing how the model output varies by the manufacturer value.

ufacturer ”R” is more subject to leaks than other manufacturers. Fig 5 and Fig 6 show how the model output varies by the manufacturing year and the commissioning year of each item respectively. Younger regulators have a lower leak risk, however the risk is plateaued when they are older then thirty years.

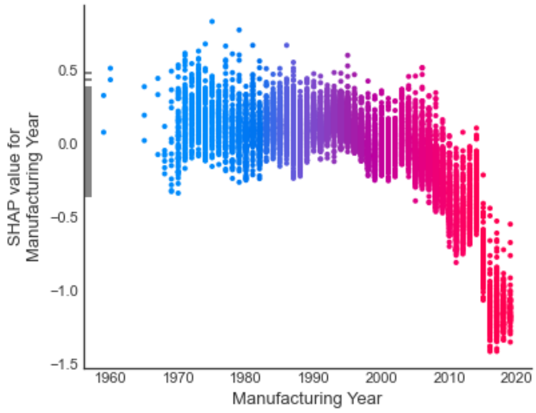

Fig. 5. SHAP dependence plot, showing how the model output varies by the manufacturing year value.

3.2. *Bayesian Networks*

The Bayesian network models bring another insight to the problem. Starting with the unsupervised structural learning (i.e, no selected target) which can find any kind and any number of probabilistic relationships between variables in a data set. Fig.7 shows the result of the ”Maximum Weight Spanning Tree” learning algorithm which consists of computing the *prior weight* of all binary relationships between all variables, then constructs the Maximum Weight Spanning Tree of those relationships. It is a practical basis for performing variable clustering. As we can see, seven different variable clusters were formed, shown by different colors. The size of each node is proportional to its force. We observe ”families” of correlated features, such as those relating to the dimensions (in yellow), to the age of assets (in purple), and to the architecture of the unit (in pink).

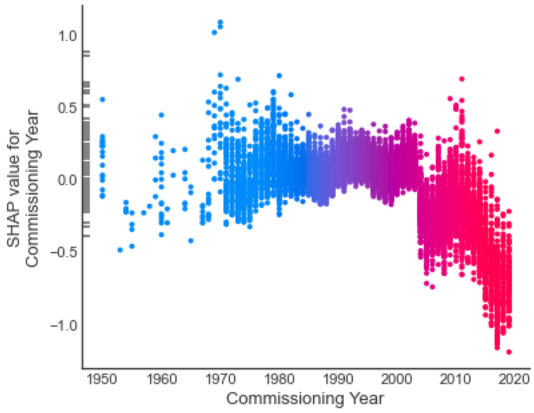

Fig. 6. SHAP dependence plot, showing how the model output varies by the commissioning year value.

Then with supervised learning, we attempt to find the best probabilistic characterization of the Target Node (leak rate), (i.e, producing a useful predictive model). We used the Naive Bayes network since it has a predefined structure in which the Target Node is the parent of all the other nodes. This structure implies that the Target Node is the cause of all the other nodes and that the knowledge of its value makes each node independent of the others. Fig.8 shows the mapping of this model, the variables are ordered by their direct effect on the target.

To check if the results of the Naive Bayes model

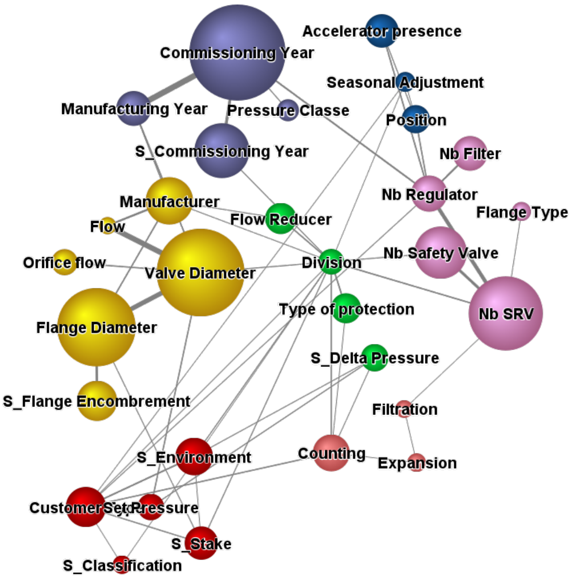

Fig. 7. Unsupervised Bayesian mapping showing the relation between variables.

are compatible with the XGBoost Cox Survival model, we check the direct effect of the position and manufacturer variables on the leak rate. First we fix all the other variables probabilities to the mean, then we select the position and the manufacturer values with the highest risk in XGBoost model: R1L1 and R (see Fig.9 and Fig.10 left), the rate value shows an increase of 0.856 and 0.393 respectively. We do the same for the lowest risk values (R2L2 and Other) (see Fig.9 and Fig.10 right). The rate value shows a decrease of 1.799 and 0.793 respectively. These results are compatible with XGBoost Survival model, since the features values follows the same order (from the least to the most leaking).

BayesiaLab curve analysis shows the correlation between the target and each variable. Fig.11 shows the correlation of the commissioning and manufacturing year with the leak rate, older regulators have a higher leak rate, which is compatible with the XGBoost Survival Cox model result, however, the plateau effect is only observed when considering a ”mean effect” between the commissioning and manufacturing years.

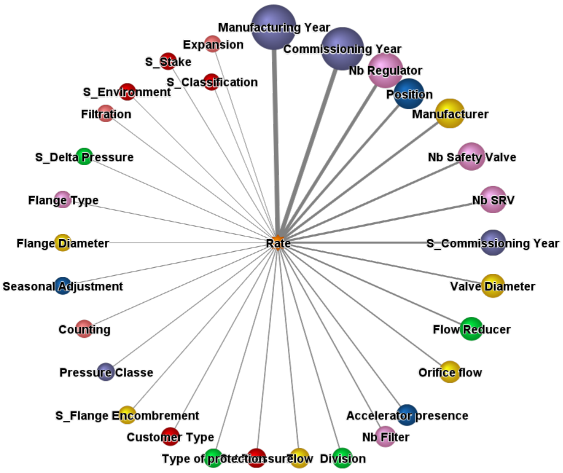

Fig. 8. Supervised Bayesian mapping showing the effect of variables on the target.

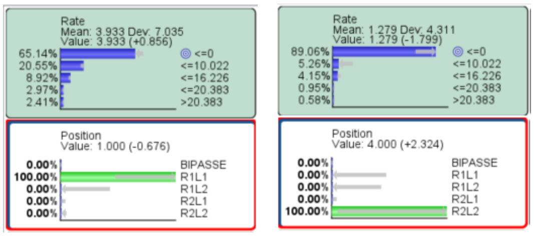

Fig. 9. Naive Bayes direct effect of position on the leak rate (rates have been normalized).

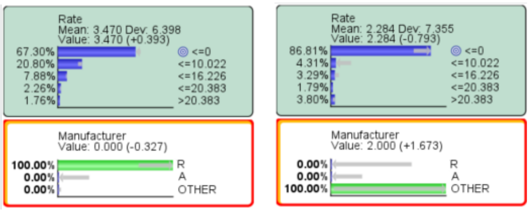

Fig. 10. Naive Bayes direct effect of manufacturer on the leak rate (rates have been normalized).

4. Conclusion

In this paper we have tested the use of two machine learning models on field data, to analyse the influencing factors of leak rate. The result of the investigation of the two models shows that both are efficient to analyse the leak factors of the assets, and to identify those that are most likely to

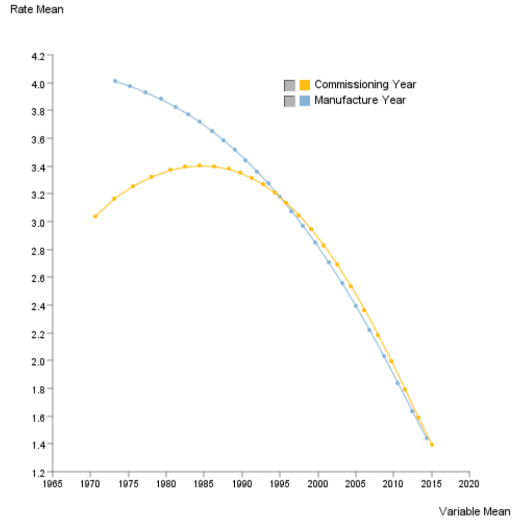

Fig. 11. Naive Bayes model correlation between Commissioning and Manufacturing year with the leak rate.

leak, even with data of different nature (discrete, continuous. . .) and missing values.

Bayesian networks implemented by BayesiaLab are very convenient because of a dedicated software tool that can perform all the suitable analysis and provide illustrations of the results. However, it is a commercial tool. XGBoost is also powerful. It is an open-source library, but it requires more experience in data handling and programming. To get illustrations of the results, a SHAP library is required.

Considering the identification of the assets that are most likely to leak, the two methods do not provide the same results. However, because of a ”risk-based” approach, these results should be only considered as indicators for optimizing a policy of methane emission reduction. Therefore, the results of both methods are used for further campaigns of gas leak detection. The feedback collected in the following months will then be used for evaluating the actual ”success rate” of each method for the identification of ”leaky assets”.

References

Brissaud, F., D. Charpentier, M. Fouladirad, A. Barros, and C. Bérenguer (2010). Failure rate evaluation with influencing factors. *Journal of Loss Prevention in the Process Industries 23*, 187–193.

Brissaud, F., L. Marle, and D. Faure (2019). Reliability factors analyses for gas transmission items. *Proceedings of the 29th European Safety and Reliability Conference*.

Chen, T. and C. Guestrin (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological) vol. 34*, 187–220.

Jouffe, L. and S. Conrady (2020). Bayesialab.

Scott, L. and L. Su-In (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems 30*, 4765–4774.

Uno, H., T. Cai, M. Pencina, R. D’Agostino, and L. Wei (2011). On the c-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. *Stat Med 30*, 1105–1117.