Predictive Maintenance Applications for Machine Learning

Brad Cline, Parametric Technology Corporation

Radu Stefan Niculescu, Parametric Technology Corporation

Duane Huffman, Parametric Technology Corporation

Bob Deckel, Parametric Technology Corporation

Key Words: Machine Learning, Predictive Maintenance, Predicted Failure Analysis

SUMMARY & CONCLUSIONS

Machine Learning provides a complementary approach to maintenance planning by analyzing significant data sets of individual machine performance and environment variables, identifying failure signatures and profiles, and providing an actionable prediction of failure for individual parts.

1 INTRODUCTION

Systems operating in industrial settings require a high degree of Reliability and Availability. Facilities like power generation plants, manufacturing lines, and oil & gas drilling sites rely on the operational capability of complex equipment. The consequences of unscheduled maintenance include not just the costs of lost production revenue. Failures impact the bottom line through repair and clean-up costs while critical failures can potentially cause harm to people and the environment. Reducing the likelihood of failure in these environments has long been a goal of Reliability Engineering.

Standard RAMS and reliability centered maintenance analyses are good techniques for mitigating risk and reducing unexpected downtime for fielded systems. Reliability Predictions are most often based on handbook models of Reliability, Weibull works with incomplete field data and assumptions about the failure distribution, while RCM uses both of these methods coupled with rules of thumb to define maintenance intervals and procedures. While useful, these approaches look at the equipment population as a whole and do not provide insight into whether a particular part is likely to fail at a given point in time. This critical piece of information is what is required to manage preventative maintenance in a cost effective manner.

Machine Learning [1], a form of Artificial Intelligence that consists of a set of advanced algorithms to build models to predict certain outcomes based on historical data about a problem, provides a complementary approach to maintenance planning by analyzing significant data sets of individual machine performance and environment variables, identifying failure signatures and profiles, and providing an actionable prediction of failure for individual parts.

In a real world application of Machine Learning, we recently analyzed inspection data for connectors at various oil & gas drilling installations. These connectors are deployed in a high fatigue environment, subjected to 15,000 psi, proppant flowing at 40 ft/sec, and corrosive acid and inhibiters. Failure of a connector at a drilling site can create loss of revenue due to down time, significant costs for equipment replacement, clean-up and collateral damage, and expose workers and the surrounding environment to potential safety harms.

In this use-case the customer, a manufacturer of oil and gas equipment that also has a service/maintenance arm, has collected a long history of inspection data for the assets serviced (including the connectors which represented the focus of this study) and used this data to periodically supply their downstream customers with reports of potential imminent connector failures. The process currently used by the service organization identifies roughly only 1% of all connector failures. Using the same inspection data but developing a model through Machine Learning techniques we were able to identify 61% of the connector failures. This represents an enormous increase in predictability and is projected to have a substantively positive impact on field operations.

This paper will discuss Machine Learning analysis, how it will better identify failures and improve preventative maintenance while providing an example of a real world application of the process.

2 DATASET

We looked at a dataset containing a subset of maintenance inspections performed by the customer across a period of 19 years, starting in 1997 and ending in 2015. These inspections cover two families of oil and gas equipment, swivels and valves, for two downstream customers. There are 123 different products with a total of 206 different model variations, some of them serviced more often than others. For the most prevalent products, we have about 35.5 thousand tracked assets, whereas the least frequently seen product appeared only once. Each asset is tracked from the time the service contract started for a specific batch of assets, until the asset is scrapped or the contract expires. Some of the newer assets have been serviced only once, whereas at the other extreme, some older assets have seen up to 43 inspections.

Typical maintenance consists of the downstream customer / end user of the equipment sending a batch of assets

978-1-5090-5284-4/17/$31.00 ©2017 IEEE

periodically for inspection. Most of the time, inspections take place a couple of times a year, depending on the type of asset and maintenance contract. During such an event, a number of ultrasonic readings are taken over the length of the asset, complemented by a number of additional tests (e.g. Pressure test, Gauge test). If the ultrasonic readings of asset wall thickness are less than a threshold, or if certain critical tests fail, the asset is scrapped. If other less critical tests fail, an attempt is made to fix the asset and return to the customer.

In addition to the information above, the dataset records the customer name, the location of the site where the equipment is used, the service facility location, serviced equipment information (item number, style, family, size, manufacturer, pressure rating, date first seen in service), and details of the service visit (job number, date performed, service level, required interval between inspections, status of the inspection, and rejection reasons if applicable), for a total of 96 variables.

For the purpose of our proof of concept engagement, to provide the most business value, we have agreed with the customer to focus on the asset that was most serviced, a particular type of sophisticated swivel. This swivel is used in two different settings: fracking or cementing, with different observed failure risk profiles.

3 ANALYTICS METHODOLOGY

To run our algorithms, we first need to frame the Predictive Maintenance problem in relation to the available data. When looking at a Predictive Analytics problem, one must first define a Goal / Outcome variable, a Cohort of examples to build and validate a model on, as well as a set of Predictive Variables. With these in place, the data needs to be transformed into a format conducive for Machine Learning. The most common such format is a data frame where rows represent examples to predict or learn from, and columns represent variables, including both Predictive Variables and Goals. This data frame can then be fed to various Machine Learning algorithms to build and validate models which can then be consumed by end users in various forms such as via daily / weekly reports, dashboards, or live, interactive applications. Below we will describe the above concepts in our context, will talk about the Machine Learning models built and how they performed, and will touch upon a few considerations related to the real world implementation of Predictive Maintenance.

3.1 Goal Variables

In our initial design sessions, we looked at the available data and came up with three candidate Goal variables. To follow the downstream customers' budget cycles, there was interest in predicting which assets are going to *Fail in the Next Year*. This would not only prevent unnecessary downtime by overusing an asset, but also allows the incorporation of a replacement budget in the overall yearly budget of the downstream customers, the field operators. Other outcome variables we considered were *Failure at the Next Inspection*, and *Remaining Life of Asset*. The first two goal variables were Boolean (True / False), whereas Remaining Life was continuous.

The ability to compute the chosen goal variable determines the size of the training dataset and therefore impacts the performance of the predictive models. A larger dataset typically results in better models owing to the ability to capture more predictive trends in the overall asset population. For example, in our case, we were unable to compute *Remaining Life* for the majority of assets due to the simple fact that a lot of the assets have not failed yet.

During early discussions with the customer, we learned that once an asset passes the inspection and is returned to the field, it is not necessarily immediately put back in production, and instead it may sit on a shelf until needed by the downstream customer. Since our customer, the service organization, did not have direct visibility into the period a specific asset sits on a shelf in the field, this variable could not be presented to our models to learn from. The lack of this information impacts much more short term predictions (e.g. *Failure at the Next Inspection*) than long term predictions.

Due to the above considerations, we narrowed the scope of our proof of concept engagement to predicting *Failure within the Next Year* after an inspection event for an asset.

3.2 Cohort Definition

With the above Goal in mind, our task is to identify / predict which assets are going to fail in the next year, based on the latest information available. Given that no additional data about the assets is collected between service inspections, it makes sense to predict failure based on the data available at the end of each inspection. Therefore, the cohort of relevant examples consists of all inspections, with multiple examples at various points in time for each asset. However, not all such inspections can be used to train a model. In order for an inspection to be useful in training, we need to be able to observe the goal variable at that point in time. Consequently, we had to eliminate the inspections that do not have a follow up *Passed* inspection a year or later after the date of service and also do not have a *Failed* inspection within a year. Note that even though these inspections cannot be used to train the model, once the model is built, it can still be applied to predict the failure of such assets.

3.3 Predictive Variables and Data Cleansing

In order to build the columns for the data frame required for predictive analytics, we started with the 96 variables available in the dataset and performed a number of data cleansing and transformation steps.

First of all, we have eliminated the examples for which the goal variable cannot be computed, as mentioned above. We then cut a number of variables that are either collinear (for example, given the City where the equipment is located, the State does not provide significant additional information), constant (for example product specifications such as size, minimum or maximum reading values), or are expected to have values that are not going to be seen in the future, when the model is to be used to predict failure. Examples of the latter include *Asset Serial Number* and *Date of Service*.

A number of data inconsistencies have been observed in the dataset. Working with the customer's subject matter experts, we dealt with these issues either by removing assets where a reconciliation could not occur, or by correcting the data where possible. We excluded assets where there was an inspection with *Asset Status* of Unknown or Pending, or where the *Asset Status* was Passed, even though some tests were failed, or critical readings were missing. In other cases, we were able to fix the data in place. Examples include some assets with a *Born on Date* later than the *Service Date* (the date could be reconciled using the *Serial Number*), assets with multiple inspection entries on the same date (the multiple entries were collapsed into one entry using rules such as taking the minimum reading for numeric values), or assets with readings outside possible values (for example swivel wall thickness readings greater than the manufacturing specification were lowered to the specification).

After cleansing the data, for the swivel which constitutes the focus of the Predictive Maintenance proof of concept, we were left with 30,265 examples, out of which 7078 (23.4%) were failures.

Next we enhanced the data frame with a number of Derived Variables. While the original variables were asset, location, and inspection specific, there was a lot of information to be captured from past inspections of the same asset. The intuition is that more maintenance may result in lower failure rates over a period of time, or longer remaining life. The additional variables included:

- Captured *Counts of Prior Inspections* and *Specific Tests* (not all tests were performed at every inspection and not all assets were seen per the ideal maintenance schedule). These counts are captured both over the lifetime of the asset so far, as well as over the year prior to the current inspection.
- *Wear Index* (percent of wear between manufacturing wall thickness spec to failure threshold) at various positions on the asset. A Wear Index of 0 corresponds to a new asset, whereas a Wear Index of 100% corresponds to an asset that needs to be scrapped. According to maintenance specifications, 7 readings are taken at various points along the swivel. Once the Wear Indices were created, the collinear original Readings were excluded as they do not provide additional information.
- *Min, Max, Average Wear Index* across all swivel at a given inspection.
- Corresponding variables normalized by the life of asset until the Service Date.
- *Speed of Wear over Time* and *Difference in Wear since Last Inspection*.
- *Smoothed Readings* and corresponding *Smoothed Wear Index* companion variables. Due to slight variations in the position where readings are taken at different inspections and also because of manual entry of readings in earlier years, the wall thickness readings for the same asset at a certain point sometimes were not continuously decreasing over time, as one would expect (think of the example of a car tire with more tread depth at the next annual inspection). The companion variables are smoothing the Reading curves at each reading point (out of the 7 measurements) over time by keeping previous values when the Reading is increasing (see Figure 1). Once Smoothed Wear Indices are computed, the collinear Smoothed Readings are excluded.

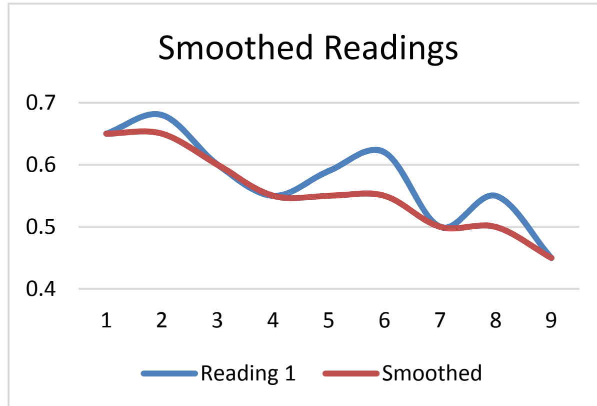

Figure 1 - Smoothed Reading Companion Variable

We mentioned before how the dataset has about 19 years' worth of data. During this time, a series of enhancements were made to both the software used, as well as the measurement procedures and tools, resulting in an improvement in data quality over time. For example, ultrasonic readings are now recorded automatically, eliminating the error prone step of entering these fractional numbers in the system by hand. However, data errors may still happen due to lack of calibration of the reading device. As of 2010, if an increase in one of the wall thickness readings is noticed when compared to previous inspections, the software alerts the service technician who can either validate or correct it. The increase in data quality over time also translates in better models as we will see below.

3.4 Analytical Models

With the data frame prepared as described in the previous step, we began building a number of Predictive Models to identify assets that are going to fail within a year of the maintenance inspection. For this purpose, we employed ThingWorx [2], a platform chosen by the customer for its ability to perform both Advanced Analytics, as well as quickly build web based interactive applications that allow end users to evaluate and consume the results of such Analytics. Another reason for using the technology was the request to have explainable predictions, which ThingWorx is tackling by producing *Important Variables* for each individual prediction.

As a first step in our modelling, we ran a signal detection function that stack ranks the predictive variables based on metrics such as Mutual Information and Correlation against the Goal variable. We chose to proceed with the Mutual Information metric owing to its ability to measure both numerical and categorical variables. This metric can be used to control how complex ThingWorx predictive models are, by applying a threshold for feature inclusion.

We then experimented with a number of predictive analytics algorithms available in the platform, including Linear Regression, Logistic Regression, Neural Networks, Decision Trees, Random Forests, Gradient Boosting Machines, as well as with a number of Ensembles / Combinations of the above. The platform provides options to both run all these models in an automated fashion with predefined parameters, or to tweak parameters of the models. For example, one can specify the number of layers of a Neural Network, or a Mutual Information threshold that will determine how many important variables are considered when building the model.

There are two ways of modelling the Goal variable, *Failure within the Next Year*. We can either consider it a Boolean (True or Fail / False or Pass) variable or a probabilistic Risk of Failure. While in our dataset we know which assets failed with 100% confidence, for a future asset, it is more appropriate to predict a Risk of Failure as same asset may or may not fail depending on how the downstream customer uses it (e.g. if it sits on a shelf vs in production when returned from the inspection). That is why in our experiments we chose to model the Goal as a continuous risk variable with values between 0% and 100%. Therefore, the models built this way will predict a probability of failure for examples not previously seen in the dataset. These probabilities can be either aggregated to obtain an expected number of failures over the next year, for part replacement budgeting purposes, or can be used to identify assets that may need proactive maintenance or replacement (e.g. all assets with a Risk Score of 0.8 or higher).

To evaluate the performance of the models, the metric that was most important to our customer was the *Failure Capture Rate*, or, in Machine Learning terms, the *True Positive Rate*. However, a model can achieve a 100% True Positive Rate if all assets are predicted as failures, but that would not help narrow down the riskier assets. In practice, an additional metric we need to look at is the True Negative Rate (assets that will not fail and the model will predict them correctly). With this in mind, once we built a model we looked for a risk threshold that maximizes the Failure Capture Rate without misclassifying too many non-failures. In our experiments we computed both the Failure Capture Rate for the threshold that achieves 90% True Negative Rate, as well as the ROC/AUC score [3], which is an aggregated measure of the True Positive and True Negative Rates over all possible risk thresholds between 0% and 100%. In the Machine Learning community an ROC over 0.70 is considered good, and an ROC over 0.80 is very good.

When evaluating the models, we typically randomly split the dataset into a training set and a held out validation set. We use the model built from the training set to predict the examples in the held out validation set and compute the metrics described above. However, with small datasets, we are exposed to possibly large variance in the random selection of the validation set. For this reason, we employed a method called 10 fold Cross-Validation that is estimating the performance of the final model built on a full dataset by splitting the dataset into 10 folds, then building a model on 9 folds and predicting the 10^{th}, then rotating the folds in such a way that we end up with 10 submodels where each example in the original set appeared in exactly one validation set for a submodel. At this point we have collected one prediction for each example, which will ensure that our computation of the performance metrics has lower variance and is more robust than one based on a random sample of the data alone.

Next we will review the results of our experiments, as well as discuss the application prototype we created for end users to interact with the Analytical models.

4 RESULTS

Before delving into the actual predictive modeling results, we will take a moment to establish a *Baseline Comparison Model*, based on the existing reporting employed by the customer. At the time of the engagement, the customer was periodically providing its downstream customers (the oil field operators) with a report meant to identify high risk assets. An asset was placed on the report if any of the ultrasonic wall thickness readings was within 10% of the minimum passing threshold. These reports were used by the customer to make a decision on how long to put the asset back in production, or on whether they should replace it proactively. As defined by the inclusion criterion, the report, which is in itself a basic predictive failure model, does not take into account any historical inspection information, or more complex relationships between variables.

Owing to ThingWorx API automation, hundreds of models have been tried in a relatively short time span, to explore the full predictive potential of the data. Some models spanned the entire 19 years of data in the dataset. However, due to improvements in software, and automation, the quality of data has improved over the years. In general, more data results in better models, but if old data is not as reliable as the newer data, if may be more beneficial to train models on the smaller, more recent data. This hypothesis is further supported by the fact that the maintenance specification changed sometime in 2008-2009. Among the modifications, the minimum wall thickness that will allow an asset to pass inspection has slightly increased, making inspections more conservative. Therefore, some assets that would have passed inspection before would fail at that point, resulting in the Goal variable not being consistently defined for all 19 years of data. A third argument for training models on the recent data is the fact that the risk profile of the assets has changed over time, with the advent of fracking that adds additional stress on the equipment compared to previous uses.

In our experiments we investigated a variety of model techniques (Regression, Decision Trees, Neural Networks, Gradient Boosting, Random Forests, and Ensembles), each with different combinations of model parameters, over the full data set, then chose a champion model, which we then applied to progressively narrower time bound datasets. To choose the champion model, we looked at the Captured Asset Failure

Rate (True Positive Rate) of the models at the risk threshold corresponding to a True Negative Rate of 90% (we are willing to misclassify only 10% of the assets that are not going to fail). The Champion Model was an Artificial Neural Network with 3 neuron layers, which looked at all variables in the data frame (in other words we did not perform any Mutual Information feature selection as mentioned in the previous section). This model achieved a 46% Captured Asset Failure Rate, slightly above the 45% recorded for a 10 Decision Tree Ensemble, which used feature selection at a 0.01 Mutual Information threshold. Even though we did not perform explicit feature selection for the Champion Model, the Artificial Neural Networks (see [1]) perform feature abstraction (a more general feature selection) at the hidden layer levels, where intermediate neurons compose sub-features based on the initial feature sets.

Table 1 - Model Performance and Failure Rates over Time.

Metric \ Year	All	2011 +	2012 +	2013 +	
Captured Failure Rate	46%	57%	59%	61%	
ROC Score	0.77	0.81	0.82	0.83	
Average Failure Rate	23%	25%	28%	38%	

Table 1 shows the change in the performance of the Champion Model when the same configuration was trained using more recent data. We only report the results on datasets after the automated software checks and new service specifications were introduced, as mentioned above. As suspected, the models trained on the more recent data perform better in terms of both Capture Failure Rate and ROC score, despite the fact that they are trained on significantly less examples. Also note that significant increase in observed failure rate in the later years, which we attribute to the increase in popularity of fracking.

Table 2 - Champion Machine Learning vs Baseline Models.

Metric \ Year	All	2013+
Total Failures	7078	2622
Captured Failure Rate of Champion Model	46%	61%
Captured Failure Rate of Baseline Comparison Model	2%	1%
Accuracy of Predicted Failures for Champion Model	58%	79%
Accuracy of Predicted Failures for the Baseline Comparison Model	33%	69%

The ROC score of the Champion Model built on the recent years of data is over 0.80, which corresponds to a very good model. The lower performance of the model on the whole dataset is, as expected, due to the lower quality of the data in the earlier years, as detailed in 3.3. We assessed our model against the *Baseline Comparison Model* on a couple of dimensions and time periods. As seen in *Table 2*, the Machine Learning models dramatically outperform the incumbent wall thickness report in terms of ability to recognize true future failures (61% captured on newer data vs only 1% identified by the report). Another metric to look at is the Accuracy of the model when it predicts a failure. The Machine Learning model performs better here as well, though the difference is less pronounced. This improvement in performance is expected as Neural Networks have the ability to uncover very complex relationships between variables.

Due to the imbalance of the positive class (Failure Rate being 23% over the whole population), one may wonder how our models compare to some naïve classifiers. First, a model that always predicts that there is no failure, while 77% accurate, will be useless for predictive maintenance purposes since it does not identify any true failures. A model that always predicts failure, while capturing all true failures, will be impractical as it is not possible to proactively replace or pay additional attention to the whole stock of swivels. Finally, a model that randomly predicts failure in the proportion seen in the overall population (23%), while it will estimate very well the overall aggregated failure rate, it will only capture about 5% (23%*23%) or the individual failures, a factor of 9 worse than the Champion Model above. Also note that both the Capture Rate and the ROC score above are unbiased performance estimators, since each model that was part of the cross validation was evaluated on a dataset not seen in its training.

When we looked at the features which are predictive of failure within the year after inspection, the *Location* of equipment and service facility, the *Average Wear Index* (how worn is the asset overall), the *Max Wear Index* (think of this as the reading at the most fragile point across the swivel, possibly a different point for each swivel) popped to the top of the list. Interestingly, they were followed by the *Wear Index* corresponding to the 3rd reading across the swivel, which may indicate useful information to feed back to the design teams for future iterations of the product. In our investigation we also looked at what is driving the remaining asset life. In this case, while the location was still at the top, the following variables were related to the inspection history (counts of prior inspections and tests of various kinds).

4.1 Results Visualization

As a deliverable of our engagement, we used a drag and drop ThingWorx platform tool to quickly prototype a web application that allows End Users (service mangers) to meaningfully consume the analytics results. The resulting application is both prospective and retrospective in nature.

On one side, the tool is employing the predictive models detailed above to predict failure for previously unseen assets or assets for which we have new inspections. The assets are categorized as Very High, High, Medium, Low, or Very Low risk and sorted in reverse order of risk. An embedded Google Maps widget allows a service manager to see where these assets are located, thus enabling new service delivery models that may be location and risk driven. The same information can be used to assist customers in their budgeting cycle, by providing them with better reports on how many assets may need replacement in the next year, and potentially negotiating discounts if the customers proactively purchases based on the predicted needs.

Another aspect of the application is retrospective in nature, whereby the service manager can visualize the variables that drive failure, as well as automatically identify sub-segments / profiles of assets with failure rates much higher than the average. We call these *Hot Spots*. Their counterparts, *Cold Spots*, are equally important as they may be indicative of good use practice, which can hopefully be replicated to reduce the prevalence of Hot Spots. In 3.4 we talked about how the ThingWorx platform identifies most relevant features using Mutual Information or Correlation. Similarly, there is an API call that applies BEAM search to automatically look for combination of variables that describe a Hot or Cold Spot. Once the most important such segment is identified, the algorithm iteratively looks at the remaining asset population until all significant segments have been identified. In traditional Business Intelligence applications, this is a very time consuming task, where an end user is trying various drill downs on many dimensions, ending with potentially overlapping segments. To give an example, the application identified a segment of assets that are located in the same place and exhibit the same readings in the middle 3 readings across the swivel, and have a failure rate twice the average failure rate in the overall population. The retrospective aspects of the application allow the end users and the R&D teams who created the product to perform root cause analyses and iteratively improve the design to tackle the automatically detected problem spots.

Note that these proof of concept models and application have been developed based on a static dataset. When put into production, one needs to consider the dynamic nature of the problem, where new assets are serviced every day. That requires to automatically connect and ingest data into the application, as well as produce scores on a frequent (daily or near real-time) basis. Also, with new data, we will require the ability to retrain the predictive analytics models either on a schedule or on demand, perhaps when certain performance metrics degrade or when we have reason to believe that training on newer data will result in a better model, as we have seen in our results.

5 SUMMARY AND FUTURE WORK

In this paper we demonstrated the potential of Machine Learning techniques on enhancing the operations of an Oil and Gas equipment service department. Our approach resulted in major improvements in the customer's ability to identify risky assets up to one year in advance. While we only proved our concept on the most frequent asset seen by the service organization, the framework presented is easily extensible to predict other assets or specific failure modes such as *Erosion*, *End Splits*, or *Pitting*.

This data set was not developed with the goal of using it for Machine Learning in mind. There are several ways to enhance the data that will make the Machine Learning analysis even more impactful. While the maintenance data is discrete in nature, real-time sensors (or downloaded data from assets) recording environmental and operating conditions will provide more complete data for overall better models, as well as monitoring capability with equipment failure alerts. Similarly, RFID, QR, or Barcode asset tagging can supply the service organization with additional information about when an asset is in production vs on a shelf. The predictive models presented in this paper are expected to improve when learned from data coming from more downstream customers and geographic locations.

Predictive failure identification and maintenance through Machine Learning techniques will provide additional benefits beyond the obvious improvements in risk mitigation and maximizing system uptime. Integration to a Parts Management system will reduce service parts inventory and increase fill rates. Additional opportunities include product design improvements, maintenance plans tailored to an asset's context, and performance-based service contracts ("drilling by the hour").

REFERENCES

1. T. Mitchell, *Machine Learning*, McGraw Hill, 1997.
2. ThingWorx Platform, https://www.thingworx.com, PTC.
3. J. A. Swets, *Signal detection theory and ROC analysis in psychology and diagnostics: collected papers,* Lawrence Erlbaum Associates, Mahwah, NJ, 1996.

BIOGRAPHIES

Brad Cline, MBA
41 W. Otterman St.
Greensburg, PA, 15601, USA

e-mail: bcline@ptc.com

Brad Cline is a services manager and global quality subject matter expert at PTC. He has managed quality and reliability software implementation, consulting and training activities for hundreds of customers in the aerospace, defense, electronics, medical device, consumer product and industrial product industries. Additionally, Brad has taken on the role of Business Development Director for the ThingWorx Analytics practice in Global Services which focuses on implementing Machine Learning technology to address many business needs including predictive failure and maintenance applications. In addition to being a certified reliability engineer (CRE), he has a BS in Applied Mathematics (Operations Research) and an MBA.

Radu Stefan Niculescu, PhD
Parametric Technology Corporation
550 E Swedesford Rd #260
Wayne, PA, 19087, USA

e-mail: sniculescu@ptc.com

Radu Stefan Niculescu leads the Data Science Consulting team at Parametric Technology Corporation, delivering engagements that drive better business outcomes in a variety of verticals such as Healthcare, Manufacturing, Retail, Financial and Insurance, and Media. He holds a PhD in Machine Learning, 9 patents, as well as recognized works in a number of top journals and conferences in Machine Learning / Predictive Analytics, Healthcare, and Mathematics. His experience also includes Product Management, Strategic Planning, and R&D Management.

Duane Huffman
Parametric Technology Corporation
550 E Swedesford Rd #260
Wayne, PA, 19087, USA

Duane Huffman is a Technical Sales Manager and a global quality, risk and reliability expert at PTC. Throughout his career at PTC he has partnered with customers to provide reliability engineering consulting services, evaluate their quality and reliability processes, and implement software solutions. Additionally, Duane works with the software development organization to provide industry insight and direction for adding new quality and reliability capabilities. Prior to joining PTC through the acquisition of Relex Software Corporation in 2009, his work at an industry-leading consumer electronics manufacturer included responsibility for quality and reliability of high-end televisions, both in manufacturing and in the field. In this role Mr. Huffman was responsible for closing the loop between the market, vendors, quality assurance, manufacturing, and design. In addition to being an ASQ Certified Reliability Engineer, he has a BS in Electrical Engineering from Pennsylvania State University.

Robert Deckel
Parametric Technology Corporation
550 E Swedesford Rd #260
Wayne, PA, 19087, USA

Robert Deckel is a Principal Software Engineer at PTC. Including working at PTC, Robert worked in other industries including Pharmaceuticals and Healthcare. He has managed various software platforms ranging from User Interfaces for Machine Learning, Image Claims Processing, Workflow Management and designing and implementing web sites.