

A COMPARISON OF THREE DATA-DRIVEN TECHNIQUES FOR PROGNOSTICS

Kai Goebel^{*}, Bhaskar Saha⁺, Abhinav Saxena[#]

^{*}NASA Ames Research Center

⁺MCT, NASA Ames Research Center

[#]RIACS, NASA Ames Research Center

MS 269-4, Moffett Field, CA 94035

kai.goebel@nasa.gov, bsaha@email.arc.nasa.gov, asaxena@riacs.edu

Abstract:

In situations where the cost/benefit analysis of using physics-based damage propagation algorithms is not favorable and when sufficient test data are available that map out the damage space, one can employ data-driven approaches. In this investigation, we evaluate different algorithms for their suitability in those circumstances. We are interested in assessing the trade-off that arises from the ability to support uncertainty management, and the accuracy of the predictions. We compare here a Relevance Vector Machine (RVM), Gaussian Process Regression (GPR), and a Neural Network-based approach and employ them on relatively sparse training sets with very high noise content. Results show that while all methods can provide remaining life estimates although different damage estimates of the data (diagnostic output) changes the outcome considerably. In addition, we found that there is a need for performance metrics that provide a comprehensive and objective assessment of prognostics algorithm performance.

Key Words: Damage progression; data-driven techniques; Gaussian process regression; neural network; prediction; prognostics; relevance vector machine; remaining useful life; RUL

I. INTRODUCTION

There are different strategies for remaining useful life (RUL) estimation using data-driven methods. One strategy directly estimates RUL by applying a multivariate pattern matching process from the data to the remaining life. Another approach is to estimate RUL indirectly by first estimating damage, then performing a suitable extrapolation to the damage progression and calculate RUL from the intersection of the extrapolated damage and the failure criterion. The latter approach is more closely aligned with engineering reasoning but it requires the definition of both damage and a failure criterion which is often times very difficult to establish. Data-driven approaches rely on the availability of run-to-failure data. This kind of data is hard to come by and there are very few public repositories [1] available that allow a comparative analysis of different

prognostic algorithms. In this paper we attempt to compare three different algorithms on the same data set and the emphasis has been laid on their predictive capabilities, which in turn is a function of their capability to learn and generalize from the training data.

II. DATA-DRIVEN TECHNIQUES FOR PROGNOSTICS

Common to data-driven approaches is the modeling of desired system output (but not necessarily of the mechanics of the system) using historical data. Such approaches encompass “conventional” numerical algorithms, like linear regression or Kalman filters, as well algorithms that are commonly found in the machine learning and data mining communities. The latter algorithms include neural networks, decision trees, and Support Vector Machines. We enumerate below the most popular methods for data-driven techniques employed for prognostics. The review given in [2] provides an extensive overview over data-driven methods in the context of computational intelligence.

One of the most popular data-driven approaches to prognostics is artificial neural networks ([3]-[12]). An artificial neural network is a type of (typically non-linear) model that establishes a set of interconnected functional relationships between input stimuli and desired output where the parameters of the functional relationship need to be adjusted for optimal performance. Besides supervised networks, other types such as reinforcement learning [13] have been proposed. Some of the conventional numerical techniques used for data-driven prognostics include wavelets [7], [14], Kalman filters [4], particle filters [15],[17] regression [18], demodulation [19], and statistical methods [20]. Another popular technique that is used for prognostics is fuzzy logic [21],[22]. Fuzzy logic provides a language (with syntax and local semantics) into which one can translate qualitative knowledge about the problem to be solved. The fuzzy reasoning mechanism has powerful interpolation properties that in turn give fuzzy logic a remarkable robustness with respect to variations in the system's parameters, disturbances, etc.

A core issue encountered in making a meaningful prediction is to account for and subsequently bound various kinds of uncertainties arising from different sources like process noise, measurement noise, inaccurate process models, etc. in the whole exercise. Long-term prediction of the time to failure entails large-grain uncertainty that must be represented effectively and managed efficiently. For example, as more information about past damage propagation and about future use becomes available, means must be devised to narrow the uncertainty bounds. Prognostic performance metrics should take the width of the uncertainty bounds into account. Therefore, it is critical to choose methods that can take care of these issues in addition to providing damage trajectories. In [6], the authors introduced a confidence prediction neural network that employs confidence distribution nodes based on Parzen estimates to represent uncertainty. The learning algorithm is implemented as a lazy or Q-learning routine that improves uncertainty of online prognostics estimates over time. Not all data-driven techniques can be expected to inherently handle these issues and thus must be combined with other methods suited for uncertainty management. Some such techniques used for dealing with uncertainty include Dempster-Shafer theory [23] or using a Bayesian framework with relevance vector machines combined with particle filters [24]. In another effort to reduce uncertainty, the concept of prognostic fusion has been introduced, [25], [26] Here, similar to multiple

classifier fusion, the output from several different prognostic algorithms is fused such that the resulting output is more accurate and has tighter uncertainty bounds than on average the output of any individual algorithm alone. In this paper we have not addressed this issue directly, but our choice of RVM and GPR algorithms is driven by this requirement which have been shown to possess uncertainty handling capabilities.

In the next section we discuss the three techniques we chose to compare followed by the methodology specific to the dataset we used for this study.

III. METHODS

The choice for the algorithms was motivated largely by the desire to benchmark typical algorithms upon which the assessment of further algorithms will be based. Generally, the neural net approach stands out by its relative simplicity by which it can approximate coefficients of an exponential damage propagation function in response to different operational stimuli. In contrast, RVM and GPR stand out by providing uncertainty estimates with the prediction. We will briefly discuss these techniques in this paper and focus more on the application approach and results from this study. The reader is encouraged to follow references for deeper details.

Neural Networks:

For the NN-based approach, we employ here the strategy where we learn the damage state as an intermediate step. To that end, data were first transformed into log space, where damage propagation was observed to be linear [23]. Then, the rate of change for operational settings could be learned such that the states for which there were no supporting experimental data were covered by a smooth curve, employing a network with low complexity (2-4-1) to avoid overfitting [23]. The network was tasked to learn the damage propagation rate based on operational conditions which were given by two features. Data were preprocessed to remove bias. The results were smoothed to deal with large non-monotonicities.

For RUL calculation, the rate of damage change was retrieved from the NN model and damage was calculated using an exponential damage propagation equation:

$$d_{th} = \sum_{k=t+1}^{t_{th}} e^{\log(d_{k-1} + f_{NN}(c_{i,k}))} \quad (1)$$

The damage was then iteratively calculated until the damage threshold was reached and the associated time at the threshold t_{th} was recorded. RUL was calculated by subtracting the current time t_0 from t_{th} .

$$RUL = t_{th} - t_0 \quad (2)$$

Fig. 1 shows the damage rate curve that the NN learns as a function of the operational conditions.

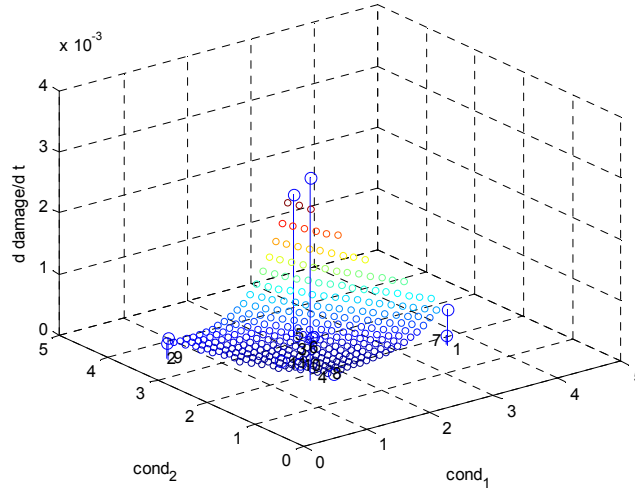


Fig. 1 - Fitted curve for damage rates as a function of operational conditions

Relevance Vector Machine

The *Relevance Vector Machine* (RVM) [27] is a Bayesian form representing a generalized linear model of identical functional form of the *Support Vector Machine* (SVM) [28]. Although, SVM is a state-of-the-art technique for classification and regression, it suffers from a number of disadvantages, one of which is the lack of probabilistic outputs that make more sense in health monitoring applications. The RVM attempts to address these very issues in a Bayesian framework. Besides the probabilistic interpretation of its output, it uses a lot fewer kernel functions for comparable generalization performance.

This type of supervised machine learning starts with a set of input vectors $\{\mathbf{x}_n\}$, $n = 1, \dots, N$, and their corresponding targets $\{\mathbf{t}_n\}$. The aim is to learn a model of the dependency of the targets on the inputs in order to make accurate predictions of \mathbf{t} for unseen values of \mathbf{x} . Typically, the predictions are based on some function $F(\mathbf{x})$ defined over the input space, and *learning* is the process of inferring the parameters of this function. The targets are assumed to be samples from the model with additive noise:

$$\mathbf{t}_n = F(\mathbf{x}_n; \mathbf{w}) + \varepsilon_n \quad (3)$$

where, ε_n are independent samples from some noise process (Gaussian with mean 0 and variance σ^2). Assuming the independence of \mathbf{t}_n , the likelihood of the complete data set can be written as:

$$p(\mathbf{t} | \mathbf{w}, \sigma^2) = (2\pi\sigma^2)^{-N/2} \exp\left\{-\frac{1}{2\sigma^2} \|\mathbf{t} - \Phi\mathbf{w}\|^2\right\} \quad (4)$$

where, $\mathbf{w} = (w_1, w_2, \dots, w_M)^T$ is a weight vector and Φ is the $N \times (N+1)$ design matrix with $\Phi = [\phi(\mathbf{t}_1), \phi(\mathbf{t}_2), \dots, \phi(\mathbf{t}_N)]^T$; in which $\phi(\mathbf{t}_N) = [1, K(\mathbf{x}_n, \mathbf{x}_1), K(\mathbf{x}_n, \mathbf{x}_2), \dots, K(\mathbf{x}_n, \mathbf{x}_N)]^T$, $K(\mathbf{x}, \mathbf{x}_i)$ being a *kernel* function.

To prevent over-fitting a preference for smoother functions is encoded by choosing a zero-mean Gaussian prior distribution over \mathbf{w} parameterized by the *hyperparameter*

vector η . To complete the specification of this hierarchical prior, the *hyperpriors* over η and the noise variance σ^2 are approximated as delta functions at their most probable values η_{MP} and σ_{MP}^2 . Predictions for new data are then made according to:

$$p(\mathbf{t}^* | \mathbf{t}) = \int p(\mathbf{t}^* | \mathbf{w}, \sigma_{MP}^2) p(\mathbf{w} | \mathbf{t}, \eta_{MP}, \sigma_{MP}^2) d\mathbf{w}. \quad (5)$$

Gaussian Process Regression

A Gaussian Process (GP) is a collection of random variables any finite number of which have a joint Gaussian distribution. A real GP $f(x)$ is completely specified by its mean function $m(x)$ and co-variance function $k(x, x')$ defined as:

$$\begin{aligned} m(x) &= E[f(x)], \\ k(x, x') &= E[(f(x) - m(x))(f(x') - m(x')))], \text{ and} \\ f(x) &\sim GP(m(x), k(x, x')). \end{aligned} \quad (6)$$

The index set $X \in \mathfrak{R}$ is the set of possible inputs, which need not necessarily be a time vector. Given prior information about the GP and a set of training points $\{(x_i, f_i) | i = 1, \dots, n\}$, the posterior distribution over functions is derived by imposing a restriction on prior joint distribution to contain only those functions that agree with the observed data points [29]. These functions can be assumed to be noisy as in real world situations we have access to only noisy observations rather than exact function values, i.e. $y_i = f(x) + \varepsilon$, where ε is additive IID $N(0, \sigma_n^2)$. Once we have a posterior distribution it can be used to assess predictive values for the test data points. Following equations describe the predictive distribution for GPR [30].

$$\mathbf{Prior} \quad \begin{bmatrix} y \\ f_{test} \end{bmatrix} \sim N \left(\mathbf{0}, \begin{bmatrix} K(X, X) + \sigma_n^2 I & K(X, X_{test}) \\ K(X_{test}, X) & K(X_{test}, X_{test}) \end{bmatrix} \right) \quad (7)$$

$$f_{test} | X, y, X_{test} \sim N(\tilde{f}_{test}, \text{cov}(f_{test})), \text{ where}$$

$$\mathbf{Posterior} \quad \tilde{f}_{test} \equiv E[f_{test} | X, y, X_{test}] = K(X, X_{test}) [K(X, X) + \sigma_n^2 I]^{-1} y, \quad (8)$$

$$\text{cov}(f_{test}) = K(X_{test}, X_{test}) - K(X_{test}, X) [K(X, X) + \sigma_n^2 I]^{-1} K(X, X_{test}).$$

A crucial ingredient in a Gaussian process predictor is the covariance function ($K(X, X')$) that encodes the assumptions about the functions to be learnt by defining the relationship between data points. GPR requires a prior knowledge about the form of covariance function, which must be derived from the context if possible. Furthermore, covariance functions consist of various hyper-parameters that define their properties. Setting right values of such hyper-parameters is yet another challenge in learning the desired functions. Although the choice of covariance function must be specified by the user, corresponding hyper-parameters can be learned from the training data using a gradient based optimizer such as maximizing the marginal likelihood of the observed data with respect to hyper-parameters [31]. After a simple description of GPR we now describe the methodology we followed for the challenge dataset.

IV. DATA

The challenge dataset contains a set of time series data from experiments running from no fault to some time after the fault. The data was obtained on a test stand involving rotating equipment in an aerospace setting. Several but not all experiments trip the failure

threshold (set here at 45 units). In some experiments the equipment keeps operating after the failure criterion has been reached. Damage was measured for each run several times, once shortly after fault initiation and several times afterwards. There are only few measurements because it was very costly and impractical to obtain measurements. Different but constant, during a run, operational and environmental conditions were used for the training sets, except where the experiments were interrupted for taking ground truth measurement. In contrast, the set used for testing was subjected to varying conditions (cyclic loading).

This work does not consider anomaly detection or diagnostics and instead focuses on the prognostic aspects. The data also includes a diagnostics flag that indicates absence or presence of the fault. Perfect diagnostics is assumed and is used to trigger prognostics whenever diagnostic flag turns true. While this is an unrealistic assumption, it does not significantly affect this study. The primary challenges encountered arise from training with sparse damage measurements. Interpolation between the measurements or a curve fit performed on the set of measurements does not take into account that damage propagation is not necessarily a smooth process and can occur in non-linear increments. Another major issue is the extremely noisy nature of the data.

We are posed with two requirements before we can make predictions. First, we must estimate the current state of the system and second, we need to estimate the damage accumulation from there on till the failure condition is met. Features are expected to be good indicators of the damage level. The operational conditions (e.g. system loading) are expected to affect the extent of damage accumulation. Keeping these requirements in mind for the challenge data set, we used our algorithms to learn two relationships. We made an assumption about the form of the damage growth model being exponential in nature i.e., $D = \exp\{\lambda \cdot t + C\}$. First, we chose to exclude cases for training where ground truth data was either missing, consisted of less than 3 data points, or did not follow monotonically growing characteristics. We, then fit an exponential curve to the damage ground truth data for that subset of cases, and assess the values of parameters λ and C for each case. This provides a regression model to compute damage progression rate for any set of operational conditions. Next, we established a relationship between the feature values and the extent of damage based on all ground truth data available from the training set. The model thus learnt was used to estimate the current state of the damage based on feature values available at the time. Since the feature data was extremely noisy we used a simple moving average filter with window size ten to smooth any sharp variations.

After describing the data preprocessing and application approach we report our findings from this study in the following section.

V. RESULTS

The results of the algorithm indicate that all algorithms can in principle come up with remaining life estimation although the actual remaining life estimates vary considerably. Figure 2 shows prediction trajectories obtained from all the three algorithms. As can be seen in the figure, all the algorithms start predictions from different damage levels. This is because we let these algorithms use their own respective estimates of the current

damage level at the time of prediction. One can observe similar trends for all algorithms with some variation in the local slopes. Two of the algorithms come up with late predictions as the time approaches closer to failure whereas the RVM does not produce late predictions. It must be noted that from safety point of view making conservative predictions is often times preferred over making late predictions. The superimposed prediction of the three algorithms is shown at times $t=3750$, $t=4250$, $t=4750$, and $t=5250$ that have a true remaining life of 1637, 1137, 637, and 137 time units. The numerical results are summarized in Table I.

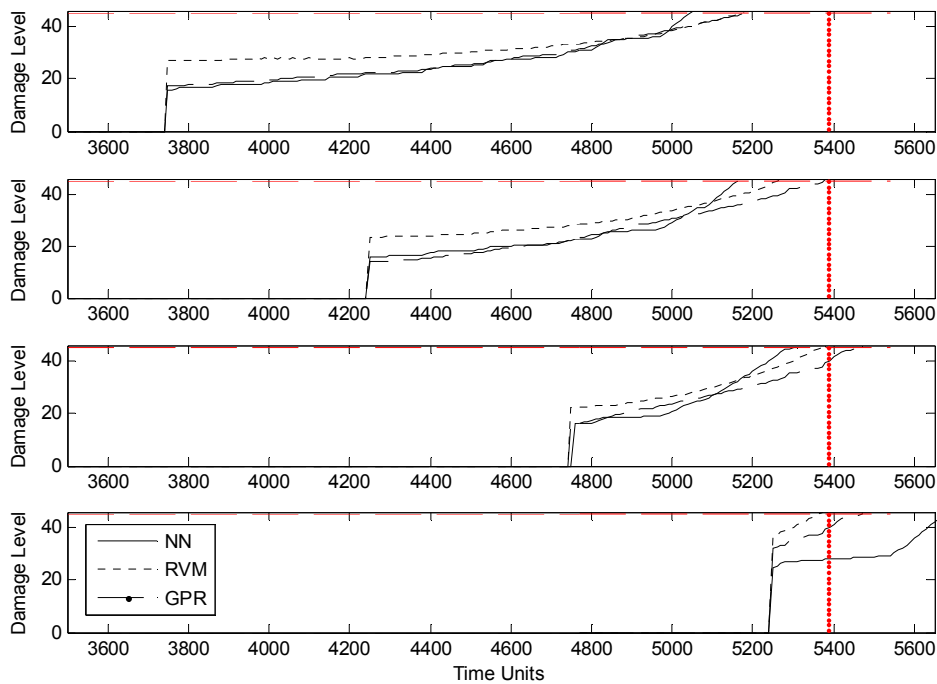


Figure 2 - Damage prediction trajectory of the 3 algorithms at different times using a algorithm specific damage estimates

Table I – Results with different damage estimation

RUL	NN Error	RVM Error	GPR Error
1637	337	207	201
1137	227	117	17
637	77	17	-83
137	-283	17	-83

Current state estimation accuracy is a function of diagnostic capability of an algorithm. In this work we did not focus on optimizing the damage estimation nor did we evaluate which algorithm provides the best damage level estimates. Instead, the focus of this paper is to evaluate the prognostic capabilities of these algorithms. To provide a better comparison of the algorithms, we deployed them using the same starting damage levels.

We chose the damage level estimates provided by GPR algorithm as a common initial point. We chose Matérn class covariance function with ν parameter $3/2$ which translates into a product of an exponential and a first order polynomial covariance function. The results have been summarized in Figure 3 and Table II. In this case the RVM algorithm also results in late predictions.

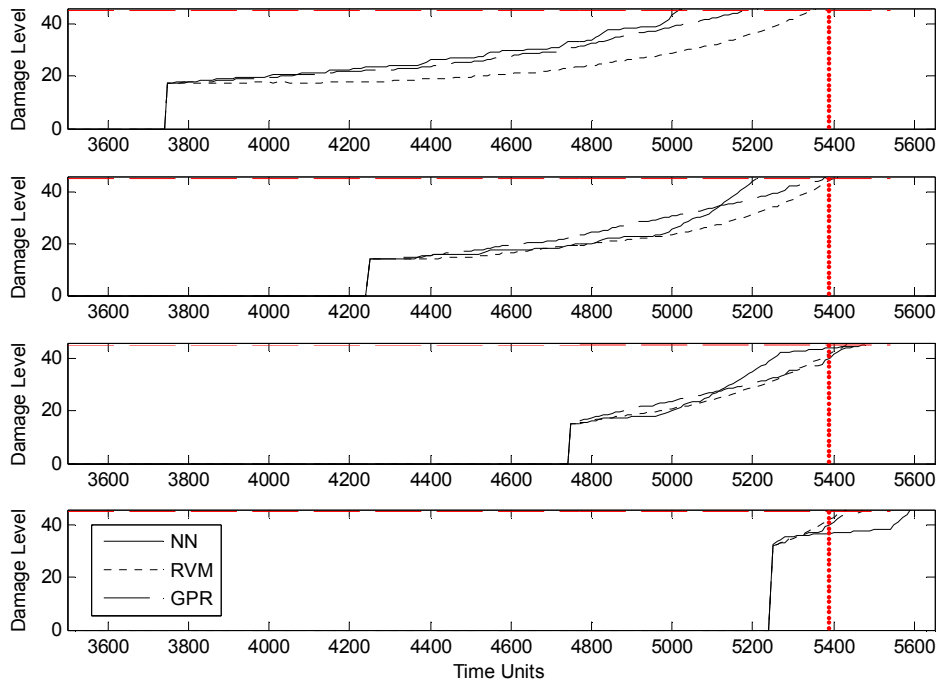


Figure 3 – Damage prediction trajectory of the 3 algorithms at different times using a common damage estimate

Table II- Results with common damage estimation

RUL	NN Error	RVM Error	GPR Error
1637	367	37	201
1137	177	-13	17
637	-103	-43	-83
137	-203	-43	-83

One issue encountered when the estimated time of failure is later than the actual time of failure is what operational conditions to use since only the operational conditions up to actual failure exist. We assumed here that the conditions would be repeated using the same cycles as up to failure. Clearly, the error will change based on what operational conditions are being chosen.

VI. DISCUSSIONS

Generally, the prediction accuracy seems acceptable. The (somewhat arbitrary) metric that the accuracy of the prediction performed halfway between first fault detection and actual failure should be within 20% of the actual remaining life is met by all three algorithms. However, this metric does not account the prediction accuracy at later times. Indeed, performance at other times varies considerably. Generally, one would expect that the prediction error becomes smaller the closer one gets to the actual end-of-life. Yet, that is only true for the RVM when it uses its own customized damage estimates. Moreover, all three algorithms predict late as the as the remaining useful life gets smaller when using the common damage estimation. This is clearly in part a function of the damage state estimation. Figure 4 shows the predictions using the GPR as an example of how the damage level estimates impacts the prediction quality. What is apparent is that the damage degree estimate does not monotonically increase, which accounts for a large degree of the variation of the remaining life estimation. Superimposed are also the ground truth measurements which one would not typically have in a fielded system. The possible explanation that the damage progression does not follow the same model as during the earlier time should not distract from the lack of a metric that (besides accuracy) quantifies the prediction qualities over time. Indeed, while data-driven techniques may generally be considered an attractive alternative for prognostics in situations where models are hard to come by, unstable prediction results can occur due to sensitivity to state estimation (for the NN-based approach) or due to sensitivity to training data coherence (for the RVM-based approach).

The intrinsic ability of RVM and GPR to fit probability distribution functions (pdfs) to the data is desirable for prognostics where uncertainty management is of paramount importance. What remains is a validation that the uncertainty estimates are in fact reasonable. A more formal approach for uncertainty management that gives an upper bound for the confidence would be desirable. In addition, we note here again the need for a metric that describes the quality of the uncertainty properties.

The limitations of the NN are rooted in the tradeoff between providing a smooth curve for damage rate parameters that can be obtained from the training data: If the training data, as was the case here, exhibit trajectories that do not support that model, it is hard to eliminate those trajectories when only few training data exist and without using some knowledge about the underlying physics. Consequently, the NN performance varies primarily with the choice of training data and of course also with the design of its architecture.

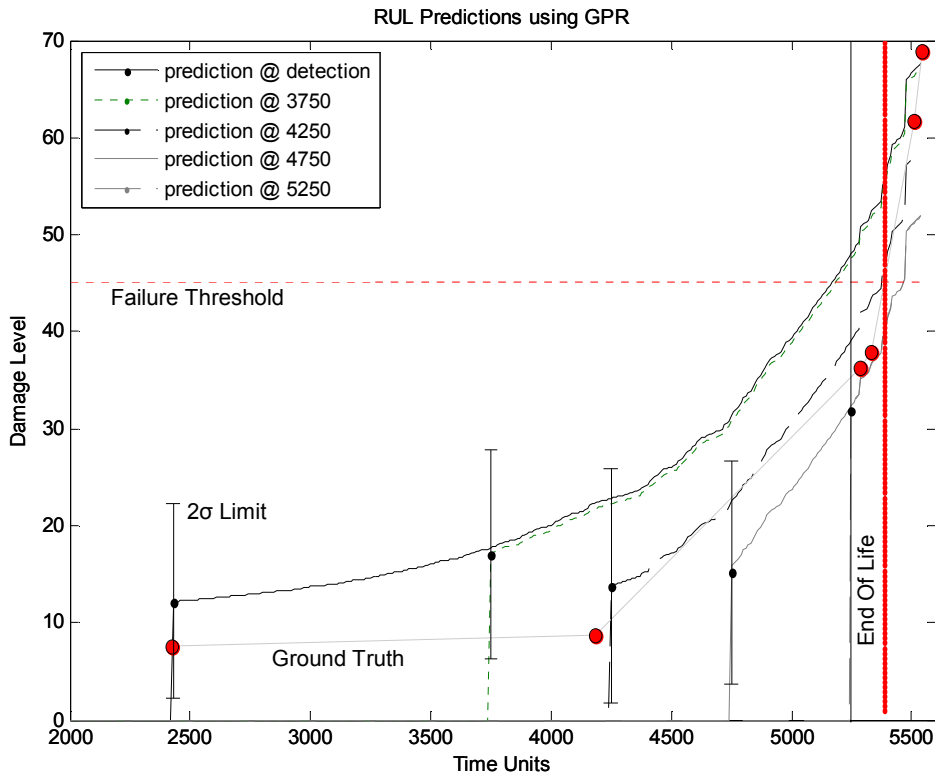


Figure 4 – RUL predictions using GPR

While GPR provides a theoretically sound framework for prediction tasks it has some limitations in its use as well. As mentioned earlier, choosing a correct covariance function is critical because it encodes our assumption of inter-relationships within data. While there are several covariance functions available from the literature [29], it is sometimes difficult to make a choice in the absence of any knowledge about the actual process that governs the system. Although methods have been suggested to evaluate various covariance functions based on likelihood values, the task is reduced to picking the best out of available ones but still does not guarantee that our assumptions about the process were correct.

GPR provides variance around its mean predictions. The premise is that it computes the posterior by constraining the prior to fit the available training data. Therefore, any prediction points lying close to training data in the input space are often predicted fairly accurately and with high confidence (small variance). For the regions where training data was not sufficiently available GPR may still predict the mean functions fairly well, assuming a suitable covariance function was identified and hyper-parameters were reasonably set. However, the confidence bounds it provides tend to be extremely conservative (large variance). Whereas this may not be very counter-intuitive for predictions involving a long time horizon these bounds get unmanageable unless somehow contained.

Another limitation arises from the fact that GPR scales typically as $O(n^3)$ with the increasing number of training examples. In our application this did not pose a problem as we had a small training data set however, it may be a limitation in terms of computational time and power in an online prognosis type of application. Various methods have been suggested for approximating the computations to reduce the problem but it can get tricky as data size increases and prediction horizon shrinks.

In the case of the RVM, its power to detect underlying trends in noisy data lies in its ability to use probabilistic kernels to account for the inherent uncertainties in the application domain. However, this advantage can also be a drawback if there are insufficient points in the training dataset or if the test dataset is unknown or significantly different such that a validation of the RVM performance on the training data has little bearing on its performance in the test case. Figure 5 shows difference in performance of the RVM in estimating the damage level from the feature values in the test dataset, having trained on selected datasets with different kernel widths. The plot on the left shows that all the widths do well on the training sets while their performances differ widely on the test case, with 7 being the optimal width value. Thus, it is difficult to come up with a strategy to select kernel width without any knowledge of the test case data.

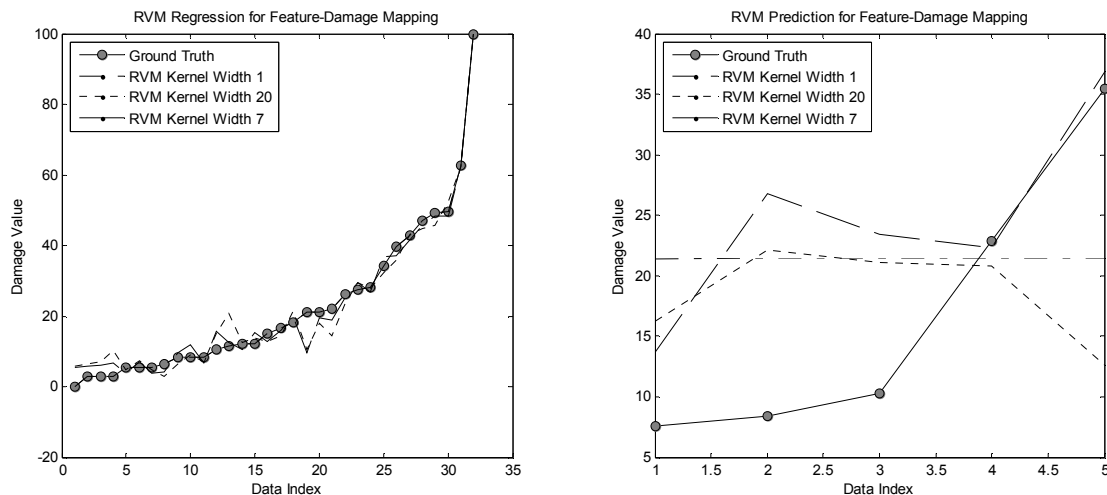


Figure 5 – Impact of kernel width on damage mapping for regression and prediction

VII. CONCLUSIONS

In this paper we have compared three regression techniques used as data driven prognostic algorithms. We have shown that while these algorithms can learn the dynamics of the process from sparse and noisy data fairly well, the RUL estimates depend significantly on the current state estimation. Each of the algorithms came up with its own estimates which were not close to each other. Clearly, the methods suffered from the low signal to noise ratio as well as the small number of training data. In particular the latter is often times a constraint that will be experienced by many systems since run-to-failure data – in particular for new systems - is hard to come by. Future work should investigate methods for dealing with sparse time series data sets, research formal methods for validation of data-driven approaches, and investigate fusion of prognostic estimates,

including concepts of estimator diversity. Additionally, as part of our future investigations, we would like to employ techniques better suited for state estimation and then use those estimates as initial points for such prediction algorithms. In addition, this work made clear that there is a need for prognostic metrics that can comprehensively quantify performance beyond accuracy and precision. Specifically, metrics that take into account the prediction horizon length, sensitivity to damage state estimation, modality of confidence distribution, preference distribution around actual time of failure, and stability/robustness of the prediction (among others) would be desirable.

REFERENCES

- [1] NASA Ames Research Center, "Prognostics Center of Excellence Data Repository web site", <http://ic.arc.nasa.gov/tech/groups/index.php?gid=53&ta=4>.
- [2] M. Schwabacher and K. Goebel, "A Survey of Artificial Intelligence for Prognostics", Working Notes of 2007 AAAI Fall Symposium: AI for Prognostics, 2007.
- [3] P. Bonissone and K. Goebel, "When will it break? A Hybrid Soft Computing Model to Predict Time-to-break Margins in Paper Machines", Proceedings of SPIE 47th Annual Meeting, International Symposium on Optical Science and Technology, Vol. #4787, pp. 53-64, 2002.
- [4] C. Byington, M. Watson, and D. Edwards, "Data-Driven Neural Network Methodology to Remaining Life Predictions for Aircraft Actuator Components", Proceedings of the IEEE Aerospace Conference, New York: IEEE, 2004.
- [5] R. Chinnam and P. Baruah, "A Neuro-Fuzzy Approach For Estimating Mean Residual Life in Condition-Based Maintenance Systems", International Journal of Materials and Product Technology, vol. 20, 2003.
- [6] T. Khawaja, G. Vachtsevanos and B. Wu, "Reasoning about Uncertainty in Prognosis: A Confidence Prediction Neural Network Approach", Proceedings of the Annual Meeting of the North American Fuzzy Information Processing Society, 2005.
- [7] M. Roemer, J. Ge, A. Liberson, G. Tandon, and R. Kim, "Autonomous Impact Damage Detection and Isolation Prediction for Aerospace Structures", Proceedings of the IEEE Aerospace Conference, New York: IEEE, 2005.
- [8] Y. Shao, and K. Nezu, "Prognosis Of Remaining Bearing Life Using Neural Networks", Proceedings of the Institute of Mechanical Engineer, Part I, Journal of Systems and Control Engineering, vol. 214, no. 3, 2000.
- [9] V. Stone and M. Jamshidi, "Neural Net Based Prognostics for an Industrial Semiconductor Fabrication System", Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, New York: IEEE, 2005.
- [10] L. Studer and F. Masulli, "On The Structure Of A Neuro-Fuzzy System To Forecast Chaotic Time Series", Proceedings of the International Symposium on Neuro-Fuzzy Systems, pp. 103-110, 1996.
- [11] A. Weigend and N. Gershenfeld, (eds.), Time Series Prediction: Forecasting the Future and Understanding the Past, Reading, MA: Addison-Wesley, 1993.
- [12] P. Werbos, "Generalization of Back Propagation with Application to Recurrent Gas Market Model", Neural Networks, vol. 1, pp. 339-356, 1993.
- [13] J. Bock, T. Brotherton, and D. Gass, "Ontogenetic Reasoning System for Autonomic Logistics", Proceedings of the IEEE Aerospace Conference, New York: IEEE, 2005.
- [14] J. Sheldon, H. Lee, M. Watson, C. Byington, and E. Carney "Detection of Incipient Bearing Faults in a Gas Turbine Engine Using Integrated Signal Processing Techniques", Proceedings of the American Helicopter Society Annual Forum, Alexandria, VA: AHS, 2007.
- [15] B. S. Bhangu, P. Bentley, D. A. Stone, and C. M. Bingham, "Nonlinear Observers for Predicting State-of-Charge and State-of-Health of Lead-Acid Batteries for Hybrid-Electric Vehicles", IEEE Transactions on Vehicular Technology, vol. 54, no. 3, pp. 783-794, 2005.
- [16] M. Orchard B. Wu, and G. Vachtsevanos, "A Particle Filtering Framework for Failure Prognosis", Proceedings of the World Tribology Congress, 2005.

- [17] B. Saha, K. Goebel, S. Poll, and J. Christopherson, "An Integrated Approach to Battery Health Monitoring using Bayesian Regression, Classification and State Estimation", Proceedings of IEEE Autotestcon, New York: IEEE, 2007.
- [18] D. Brown, P. Kalgren, M. Roemer, and T. Dabney, "Electronic Prognostics – A Case Study Using Switched-Mode Power Supplies (SMPS)", Proceedings of the IEEE Systems Readiness Technology Conference, New York: IEEE, 2006.
- [19] M. Roemer and C. Byington, "Prognostics and Health Management Software for Gas Turbine Engine Bearings", Proceedings of the ASME Turbo Expo, New York: ASME, 2007.
- [20] M. Watson, C. Byington, D. Edwards, and S. Amin, "Dynamic Modeling and Wear-Based Remaining Useful Life Prediction of High Power Clutch Systems", Proceedings of the ASME/STLE Intl Joint Tribology Conference, New York: ASME, 2004.
- [21] C. Frelicot, "A Fuzzy-Based Prognostic Adaptive System", RAIRO-APII-JESA, Journal Europeen des Systemes Automatises, vol.30, no.2-3, p.281-99, 1996.
- [22] A. Volponi, "Data Fusion for Enhanced Aircraft Engine Prognostics and Health Management", NASA Contractor Report CR-2005-214055, 2005.
- [23] K. Goebel, N. Eklund, and P. Bonanni, "Fusing Competing Prediction Algorithms for Prognostics", Proceedings of 2006 IEEE Aerospace Conference, New York: IEEE, 2006.
- [24] B. Saha, K. Goebel, S. Poll, and J. Christopherson, "An Integrated Approach to Battery Health Monitoring using Bayesian Regression, Classification and State Estimation", Proceedings of IEEE Autotestcon, New York: IEEE, 2007.
- [25] K. Goebel and N. Eklund, "Prognostic Fusion for Uncertainty Reduction", Proceedings of AIAA@Infotech Aerospace Conference, Reston, VA: American Institute for Aeronautics and Astronautics, Inc., 2007.
- [26] F. Xue, K. Goebel, P. Bonissone, and W. Yan, "An Instance-Based Method for Remaining Useful Life Estimation for Aircraft Engines", Proceedings of MFPT, 2007.
- [27] M. E. Tipping, "The Relevance Vector Machine", Advances in Neural Information Processing Systems, vol. 12, pp. 652-658, Cambridge MIT Press, 2000.
- [28] V. N. Vapnik, The Nature of Statistical Learning, Springer, Berlin, 1995.
- [29] C. E. Rasmussen and C. K. I. Williams, Gaussian Processes for Machine Learning, The MIT Press, 2006.
- [30] C. K. I. Williams and C. E. Rasmussen, "Gaussian Processes for Regression", D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo (eds.), Advances in Neural Information Processing Systems, vol. 8, pp. 514-520, The MIT Press, Cambridge, MA, 1996.
- [31] K. V. Mardia and R. J. Marshall, "Maximum Likelihood Estimation for Models of Residual Covariance in Spatial Regression", Biometrika, vol. 71, no. 1, pp. 135-146, 1984.