

# Metrics for Evaluating Performance of Prognostic Techniques

Abhinav Saxena<sup>a</sup>, Jose Celaya<sup>a</sup>, Edward Balaban<sup>b</sup>, Kai Goebel<sup>b</sup>,  
Bhaskar Saha<sup>c</sup>, Sankalita Saha<sup>a</sup>, Mark Schwabacher<sup>b</sup>

<sup>a</sup>Research Institute for Advanced Computer Science, <sup>b</sup>NASA Ames Research Center, <sup>c</sup>Mission Critical Technologies  
NASA Ames Research Center, Moffett Field, CA 94035

**Abstract** *Prognostics is an emerging concept in condition based maintenance (CBM) of critical systems. Along with developing the fundamentals of being able to confidently predict Remaining Useful Life (RUL), the technology calls for fielded applications as it inches towards maturation. This requires a stringent performance evaluation to establish the ingenuity and significance of the concept. Prognostics concepts lack standard definitions and suffer with ambiguous and incoherent interpretations. Applications vary in end-user requirements, time scales, available information, domain dynamics, etc. to name a few that have prevented from establishing any standards. The research community has used a variety of metrics based on respective requirements. Very little attention has been focused on establishing a common ground to compare different efforts. This paper presents a thorough review of various domains like medicine, weather, finance, nuclear, automotive, aerospace, electronics etc. that employ prediction related tasks and use a variety of performance metrics to evaluate these methods. Differences and similarities between these domains and health maintenance have been analyzed to help understand what performance evaluation methods may or may not be borrowed. Further, these metrics have been categorized in several ways that may be useful in deciding upon a suitable set based on the nature of a specific application. Some important prognostic concepts have been defined using a notational framework that enables interpretation of different metrics coherently. Last, but not the least, a list of metrics has been suggested to assess critical aspects of RUL predictions before they are fielded in real applications.*

## Introduction

Prognosis is emerging at the forefront of Condition Based Maintenance (CBM) of critical systems giving rise to the term Prognostic Health Management (PHM). However, there are major challenges in building a successful prognostics system that can be deployed in field applications [1]. Research efforts are focusing on developing algorithms that can provide a Remaining Useful Life (RUL) estimate, generate a confidence bound around the predictions, and can be integrated with existing diagnostic systems. A key step in successful deployment of a PHM system is prognosis verification. Since prognostics is still considered relatively immature as compared to diagnostics, more focus so far has been on developing prognostic methods rather than evaluating and comparing their performances. Tests are conducted based on specific

requirements to declare the goodness of the algorithms but little or no effort is made to generalize the performance over variety of other situations. Hence, there is no direct way of comparing different efforts if one needs to identify the most suitable algorithm from a list of several. This calls for a set of general metrics that can be used in a standardized manner. Furthermore, different users of prognosis have different requirements; hence these verification metrics should be tailored for each end user (*customer based verification*) [2]. This poses a conflicting requirement to the idea of generalization of metrics. This confusion has prevailed for sometime in the CBM/PHM community and there is a need to classify various metrics into categories catering to different requirements. In this paper we have attempted to evaluate the verification process such that it can provide a structure for how to choose performance metrics for specific tasks and also compare an algorithm with other competing ones.

In this paper we provide a concise review on a variety of domains that involve prediction tasks of some sort. All these domains have fielded prognostics/forecasting applications and have, therefore, implemented performance metrics that evaluate and compare one system with another. These metrics have been consolidated and categorized into several categories based on different criteria that may be useful to the CBM/PHM community. For the sake of consistency and clear description, a notational framework has been introduced and included along with basic prognostics related terms and definitions. The various metrics collected have been briefly explained and discussed as to how they can be of use to PHM applications. Finally, a set of metrics is suggested that may be used to evaluate and compare different algorithms in a standardized manner.

## Motivation

For end of life predictions of critical systems, it becomes imperative to establish a fair amount of faith in the prognostic systems before incorporating their predictions into the decision making process. A maintainer needs to know how good the prognostic

estimates are before he can optimize the maintenance schedule. Without any reasonable confidence bounds a prediction completely loses its significance. Confidence bounds are a function of uncertainty management capabilities of an algorithm whereas performance metrics provide a means to establish sanity of any claims regarding such confidence bounds. Therefore, these algorithms should be tested rigorously and evaluated on a variety of performance measures before they can be certified. Furthermore, metrics help establish design requirements that must be met. In the absence of standardized metrics it has been difficult to quantify acceptable performance limits and specify crisp and unambiguous requirements to the designers. Standardized metrics will provide a lexicon for a quantitative framework for requirements and specifications.

There are a number of other reasons that make the verification process important. In general three broad categories, *scientific*, *administrative*, and *economic*, have been identified for such reasons [3]. Performance evaluation allows comparing different schemes numerically and provides an objective way to measure how changes in training, equipment or prognostics models (algorithms) affect the quality of predictions. This provides a deeper understanding from the research point of view and yields valuable feedback for further improvements. One can identify bottlenecks in the performance and guide research and development efforts in the required direction. As these methods are further refined, quantitatively measuring improvement in predictions generates scores that can be used to justify for research funding in areas where either PHM has not yet picked up or where better equipment and facilities are needed. These scores can also be translated into costs and benefits to calculate Return-on-Investment (ROI) type indexes to justify their fielded applications. Therefore, it is essential to devise metrics that can measure performance of various algorithms before any implementation can be fielded successfully.

Performance evaluation is usually the foremost step once a new technique is developed. In many cases benchmark datasets or models are used to evaluate such techniques on a common ground so they can be fairly compared. Prognostics, in most cases, has neither of these options. Various research teams have shown how to evaluate their algorithms using a set of performance metrics but there have been inconsistencies in the choice of such metrics. This makes it incredibly difficult to compare various algorithms even if they have been declared successful based on their respective evaluations. It is an accepted fact that prognosis methods are application oriented and that it is difficult to develop a generic algorithm useful in every situation. Likewise, the methods to evaluate such algorithms are

expected to be different. Furthermore, there has been an inconsistency in terminology used in different applications that leads to confusion in even the basic definitions. So far very little has been done to identify a common ground to test and compare different algorithms. In a survey on data driven methods for prognostics [4], it can be easily seen that there is a lack of standardized methodology for performance evaluation and in many cases performance evaluation is not even formally addressed. Even the ISO standard [5] for prognostics in condition monitoring and diagnostics of machines lacks a firm definition of such metrics. However, there must be a way to establish a common ground that can give a fair idea of how an algorithm fares w.r.t. others. Therefore, in this paper we have attempted to review various domains where prognostics type applications exist and have matured to a point of being fielded. We have also reviewed the state-of-the-art in PHM technology and tried to structure the verification methods in a logical fashion.

## Prognostics Terms and Definitions

In this section we describe some commonly used terms in prognostics. Similar terms have been used interchangeably by different researchers and in some cases the same term has been used to represent different notions. This list is provided to reduce ambiguities that may arise by such non-standardized use.

### Assumptions:

- Here *prognostics* is considered to be the remaining useful life estimation based on the current state assessment and expected future operational conditions of the system.
- It is possible to estimate a health index as an aggregate of features and conditions
- RUL estimation is a prediction/ forecasting/ extrapolation process.
- Algorithms under consideration are capable of generating a single RUL value for each prediction. E.g., algorithms that produce RUL distributions can be adapted to compress the distribution to a single estimated number for comparison purposes.
- All systems are under continuous monitoring and have the measurement capability that can acquire data as fault evolves.

### Glossary

RUL: Remaining Useful Life

UUT: Unit Under Test

$i$ : Index for time instant  $t_i$

EOL: End-of-Life - Time index of actual end of life

EOP: End-of-Prediction – earliest time index,  $i$ , when prediction has crossed the failure threshold

$0$ : Time index for time of the birth of the system,  $t_0$

$F$ : Time index for the time when fault occurs,  $t_F$

$D$ : Time index (for time  $t_D$ ) at which the fault is detected by diagnostic system

$P$ : Time index (for time  $t_P$ ) at which the first prediction is made by the prognostic system

$f_n^l(i)$ : Value of the  $n^{\text{th}}$  Feature for the  $l^{\text{th}}$  UUT at time index  $i$

$c_n^l(i)$ : Value of the  $n^{\text{th}}$  operational condition for the  $l^{\text{th}}$  UUT at time index  $i$

$r^l(i)$ : RUL Estimation at time  $t_i$  given that data is available up to time  $t_i$  for the  $l^{\text{th}}$  UUT

$\pi^l(i|j)$ : Prediction at time index  $i$  given data up to time  $t_j$  for the  $l^{\text{th}}$  UUT. Prediction may be made in any domain, e.g. feature, health, etc.

$\Pi^l(i)$ : Trajectory of predictions at time index  $i$  for the  $l^{\text{th}}$  UUT

$h^l(i)$ : Health of system for the  $l^{\text{th}}$  UUT

$h_e^l(i)$ : Measured (ground truth value) health at time index  $i$  for the  $l^{\text{th}}$  UUT. It is usually available from post failure analysis

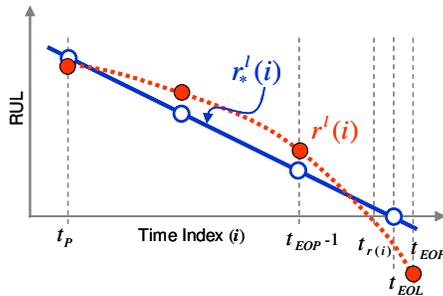


Figure 1: Illustration depicting some important prognostic time definitions and prediction concepts.

**Definition 1 - Time Index:** The time in a prognostics application can be discrete or continuous. We will use a time index  $i$  instead of the actual time, e.g.,  $i=10$  means  $t_{10}$ . This takes care of cases where sampling time is not uniform. Furthermore, time indexes are invariant to time-scales.

**Definition 2 - Time of Detection of Fault:** Let  $D$  be the time index ( $t_D$ ) at which the diagnostic or fault detection algorithm detected the fault. This process will trigger the prognostics algorithm which should start making RUL predictions shortly after the fault was detected as

soon as enough data has been collected. For some applications, there may not be an explicit declaration of fault detection, e.g., applications like battery health management, where prognosis is carried out on decay process. For such applications  $t_D$  can be considered equal to  $t_0$  (time of birth) i.e., we expect to trigger prognosis as soon as enough data has been collected and not wait for an explicit diagnostic flag (Figure 2).

**Definition 3 - Time to Start Prediction:** We will differentiate between the time when a fault is detected ( $t_D$ ) and the time when the system starts predicting ( $t_P$ ). For certain algorithms  $t_P = t_D$  but in general  $t_P > t_D$  as these algorithms need some time to tune with additional fault progression data before they can start making predictions (Figure 2). In cases where a data collection system is continuously collecting the data even before a fault is detected, enough data is already available to start making predictions right away and hence  $t_P = t_D$ .

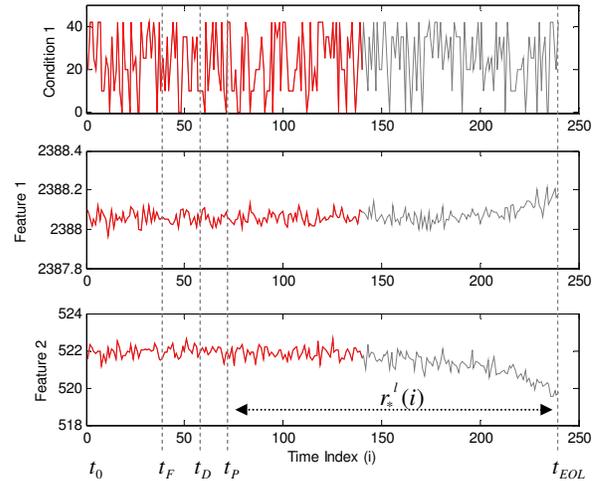


Figure 2: Features and conditions for  $l^{\text{th}}$  UUT.

**Definition 4 - Prognostics Features:** Let  $f_n^l(i)$  be a feature at time index  $i$ , where  $n = 1, 2, \dots, N$  is the feature number, and  $l = 1, 2, \dots, L$  is the UUT index (an index identifying the different units under test). In prognostics, irrespective of the analysis domain, i.e., time, frequency, wavelet, etc., features take the form of time series and they can be physical variables, system parameters or any other measurable quantity on the system that provides/aides the prognosis. The features can be also referred to as a feature vector  $F^l(i)$  of the  $l^{\text{th}}$  UUT at time index  $i$ .

**Definition 5 - Operational Conditions:** Let  $c_m^l(i)$  be an operational condition at time index  $i$ , where  $m = 1, 2, \dots, M$  is the condition number, and  $l = 1, 2, \dots, L$  is the UUT index. The operational conditions describe how the system is being operated and is sometimes

referred to as the load on the system. The conditions can also be referred to as a vector  $C^l(i)$  of the  $l^{th}$  UUT at time index  $i$ .

**Definition 6 - Health Index:** Let  $h^l(i)$  be a health index at time index  $i$  for UUT  $l = 1, 2, \dots, L$ .  $h$  can be considered a normalized aggregate of health indicators (relevant features) and operational conditions.

**Definition 7 - Ground Truth:** Let  $h_s^l(i)$  be the computed health (ground truth) at time index  $i$  for UUT  $l = 1, 2, \dots, L$  after a run to failure test. This health index represents an aggregate of information provided by features and operational conditions up to time index  $i$

**Definition 8 - Point Prediction:** Let  $\pi^l(i|i)$  be a point prediction at time index  $i$  given information up to time  $t_j$ .  $\pi^l(i|i)$  for  $i = EOL$  represents the critical threshold for a given health indicator. Predictions can be made in any domain, features or health. E.g. in some cases it is useful to extrapolate features and then aggregate them to compute health and in other cases features are aggregated to a health and then extrapolated to estimate RUL.

**Definition 9 - Trajectory Prediction:** Let  $\Pi^l(i)$  be the trajectory of predictions at time index  $i$  such that

$$\underline{\Pi}^l(i) = \{\pi^l(i|i), \pi^l(i+1|i), \dots, \pi^l(EOL|i)\}$$

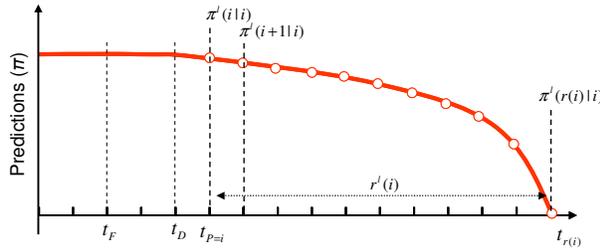


Figure 3: Illustration showing a trajectory prediction. Predictions may modify every time instant and hence the corresponding RUL estimate.

Trajectory prediction may be carried out in any domain, e.g. feature or health index. A general schematic has been shown in Figure 4.

**Definition 10 - RUL Estimation:** Let  $r^l(i)$  be the remaining useful life estimation at time index  $i$  given that the information (features and conditions) up to time index  $i$  and an expected operational profile for the future are available. As shown in Figure 4, prediction is made at time  $t_i$  and it predicts the RUL given information up to time  $i$  for the UUT  $l = 1, 2, \dots, L$ . RUL will be estimated as

$$r^l(i) = \arg\{h(z) = 0\} - i$$

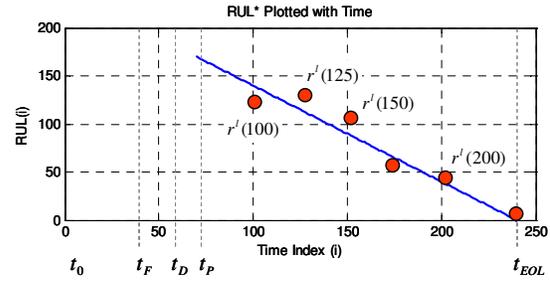


Figure 4: Comparing RUL predictions from ground truth ( $t_p \in [70,240]$ ,  $t_{EOL} = 240$ ,  $t_{EOP} > 240$ ).

### Forecasting Application Classification

Based on our survey of several forecasting application domains, we identified two major classes of forecasting applications (Figure 5).

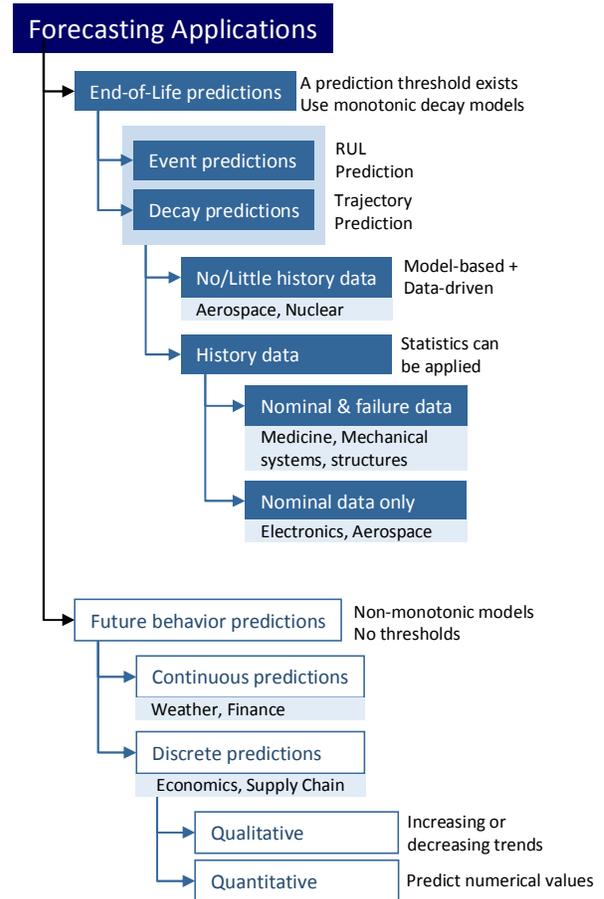


Figure 5. Different categories of the forecasting applications

In one class of applications a prediction is made on a continuous basis, and the trend of data is generally non-monotonic. These predictions may be discrete (e.g. forecasting market demand for a particular

month) or continuous (e.g. variation of temperature over the period of next week). These predictions can be quantitative (e.g. prediction exact numbers) or qualitative (e.g. high or low demands) in nature. These applications like weather and finance have been in existence since a long time and have matured to a good extent. The intent is to learn from such domains and adapt methods to our needs. The other class of applications involve where a critical threshold exists such that the system under test is declared to have died if it crosses the threshold. These applications usually can be modeled using decay models. Here the task of prognostics is to predict a Remaining Useful Life (RUL) estimate. In some cases, where enough history data exists (e.g. medicine) or can be experimentally generated (e.g. mechanical systems) for nominal and failure conditions, a variety of data-driven or statistical techniques can be applied. In such situations it is also relatively easy to evaluate the performance by comparing the prediction *a posteriori*. However, there are critical applications where run-to-failure experiments can not be afforded and very little failure history data is available (e.g. aerospace). In such cases a variety of methods based on data-driven and model-based techniques have been proposed. It becomes extremely tricky and difficult to assess the performance in such cases due to absence of knowledge about the future outcomes. Methods are tested on experimental or simulated data and are expected to perform on real systems. Unfortunately algorithm performance does not always translate meaningfully from one dataset to another or one domain to another. Therefore, a standard set of metrics independent of application domain would be very desirable.

## Forecasting Domains Reviewed

In this section we provide a concise assessment of prediction performance assessment methods in various domains. Specific relevant performance metrics have been listed in the next section.

**Aerospace:** The aerospace industry is likely the field with the most vibrant research and development activity in prognostics today. This happened for a good reason – systems health inspections on spacecraft and aircraft are often difficult and costly, and sometimes impossible. The consequences of a premature failure can, however, be dire.

Prognostic algorithms are beginning to be applied to monitoring condition of aircraft structures, avionics, wiring, control actuators, power supplies, and propulsion systems. Prognostic functionality is being incorporated into the health management system of the latest military aircraft (Joint Strike Fighter) and civilian (Boeing 747) aircrafts, in order to reduce the overall

lifecycle cost and increase flight readiness. Companies such as Boeing, Moog, Impact Technologies, and Ridgetop Corporation, among others, have established dedicated prognostics research groups. Active work on aerospace prognostics is also being conducted by government and academic organizations - NASA, Air Force Research Laboratory, DARPA, Georgia Institute of Technology, and Vanderbilt University, among others.

The aerospace industry has also led in developing the metrics to evaluate prognostic algorithms. Most of the metrics have, historically, focused on the technical merits of prognostic techniques, such as accuracy and reliability, although in the recent years more attention have been given to those accessing the business merits (ROI, Total Value, and others). As the prognostic systems make their way into the commercial aerospace sector, they are expected to help with maintenance scheduling, optimal operating mode determination, and asset purchasing decisions.

**Electronics:** Prognostics for electronics is currently less advanced than prognostics for mechanical systems. Many researchers in electronics prognostics therefore take their inspiration from previous work in mechanical prognostics, and use similar algorithms and similar metrics, including the usual accuracy metrics [6-8]. Some of the work in electronics prognostics emphasizes the potential cost savings provided by prognostics, and therefore relies on cost/benefit metrics such as ROI [9-11], life cycle cost [12], and MTBF/MTBUR ratio [13]. Methods used for data collection include measuring the temperatures of components [14, 15], installing "canaries" (electronic devices that are designed to fail before the operational devices do) [15], collecting data about operational conditions such as vibration [7], usage hours [15], or ambient temperature, using strain gauges to measure the strain on solder joints [16], and detecting when the performance of a system degrades (for example, when more correctable errors begin to occur) [14].

**Medicine:** Medicine is a field where diagnostics and prognostics have a long tradition. Indeed, medicine has a large body of tests and indicators that are used commonly to aid in decision-making such as blood pressure and cholesterol levels. The field has come to trust these prognostic indicators when they have been subject to the double blind clinical trial. While this test is not perfect, it provides a metric against which other results can be compared. Although prognostics is a common tool in medicine, the most significant constraint is the way how prognostic results are measured. Typically, survival rates are quantized into increments such that the problem boils down to a classification problem. For example, one would

typically measure the number of cancer survivors past, say, 10 years, and then assess whether the prediction was correct or not. Despite that constraint, there are a number of ancillary metrics that have been in use which quantify the quality of a prediction in the context of a regression problem.

**Nuclear:** With increasing energy demands nuclear power plants play an important role in the energy sector. Average life of a nuclear reactor being 20-30 years, efforts are underway to extend the life of these reactors using advanced monitoring and maintenance techniques. Where advanced diagnostics has been implemented in the US and Europe, prognostics is still at conceptual levels. Most metrics developed so far have been to establish a profitable business case rather than maturing prognostics itself [17, 18]. Data records like overall plant operating efficiency and maintenance, machinery repair records, etc. are used to derive cost-benefit analysis for prognosis. For instance, improved thermal efficiency is translated into gas cost savings and increase in available capacity translated into savings from not using the spare unit, etc. However, the lack of prognosis deployment has resulted in very little research in improving the prognosis itself and hence not many verification schemes.

**Finance:** In the area of finance and economics, we encountered various metrics that are used to evaluate future predictions. In most cases these metrics are sophisticated estimates of accuracy and precision. Metrics like bias, standard error, and variance have been further modified into estimates like Mean Squared Error (MSE), Average Percentage Error (APE), Mean Absolute Percentage Error (MAPE), MAD, ADE, etc. on the other hand we also encountered various tests that establish the trust in predictions. Tests like Henriksson and Merton test, Chi-squared test, Timmerman's test, Theil's U-statistic, etc. are some to name a few. In general a trajectory is usually predicted and the performance is assessed on a continuous basis as actual results become available.

**Weather:** Forecasting weather patterns has probably been one of man's earliest attempts at modeling and prediction, and continues to be just as significant today as it was before. Various modeling and forecasting methodologies have evolved from the study of weather as well as a variety of metrics to compare these techniques. However, the essence of the widely used metrics can be grouped in two categories: those that measure bias or error with respect to a baseline, and those that measure resolution or the ability of the forecast to distinguish between different outcomes. The baselines to be used as a basis of comparison can also vary between aggregate weather history (over the last 10 years, for example), current measurements or even

reference forecasts. This kind of approach is well suited to a field where measurements have improved in accuracy but our understanding of weather patterns is still evolving.

**Automotive:** Fault prognostics have recently become a vital part of on-board diagnostics (OBD) of the latest vehicles. "The goal of this technology is to continually evaluate the diagnostics information over time in order to identify any significant potential degradation of vehicle subsystems that may cause a fault, to predict the remaining useful life of the particular component or subsystem and to alert the driver before such a fault occurs." Mostly, the approach consists of trending of residuals extracted from diagnostic information. The metrics used are mainly accuracy measures like MSE or Gaussian pdf overlaps. The overall methodology is data-driven and suitable where extensive baseline data is available.

## Prognostics Metrics Classifications

A variety of prognostics metrics are used in the domains reviewed above. Depending on the end use of the prognostic information, basic accuracy and precision based metrics are transformed into more sophisticated measures. Several factors were identified that classify these metrics into different classes. In this section we attempt to enumerate some of these classifications.

### Functional Classification

The most important classification is based on the information these metrics provide to fulfill specific functions. In general we identified three major categories, namely: (1) Algorithm performance metrics, (2) Computational performance metrics, and (3) Cost-benefit metrics. As evident from their names these metrics measure success based on entirely different criteria. As shown in Figure 6, the algorithmic performance metrics can be further classified into four major subcategories.

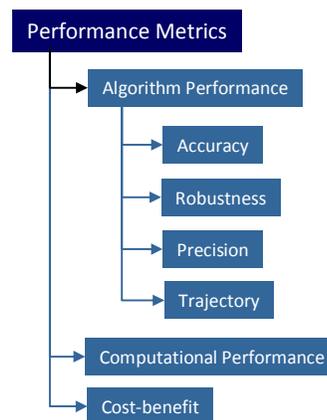


Figure 6. Functional classification of prognostics metrics.

**End User based classification**

Prognostics information may be used by different people for entirely different purposes. In general, end users of prognosis may be classified in the following five categories as shown in Table 1.

Table 1. Classification of prognostic metrics based on end user requirements.

End User	Goals	Metrics
Program Manager	Assess the economic viability of prognosis technology for specific applications before it can approved and funded	Cost-benefit type metrics that translate prognostics performance in terms of tangible and intangible cost savings
Plant Manager	Resource allocation and mission planning based on available prognostic information	Accuracy and precision based metrics that compute RUL estimates for specific UUTs. Such predictions are based on degradation or damage accumulation models.
Operator	Take appropriate action and carry out re-planning in the event of contingency during mission	Accuracy and precision based metrics that compute RUL estimates for specific UUTs. these predictions are based on fault growth models for critical failures.
Maintainer	Plan maintenance in advance to reduce UUT downtime and maximize availability	Accuracy and precision based metrics that compute RUL estimates based on damage accumulation models.
Designer	Implement the prognostic system within the constraints of user specifications. Improve performance by modifying design.	Reliability based metrics to evaluate a design and identify performance bottlenecks. Computational performance metrics to meet resource constraints.

**Classification Based on Predicted Entity**

Within PHM applications, we identified three major classes of the forms of prediction outputs and hence the corresponding metrics. Prognostics performance can be established based on different forms of the prediction outputs, e.g. future health index trajectory at  $t_p$ , an RUL estimate at  $t_p$ , or a on a RUL trajectory as it evolves with time. Some algorithms provide a distribution over predicted entities to establish confidence in predictions. Metrics to evaluate such outputs differ in form than those required for single value predictions. In other cases such a distribution is obtained from multiple UUTs, e.g. from fleet applications. The basic form of the metrics used for various categories may be similar, but the underlying information conveyed is usually

different in a statistical sense. Figures 7-9 illustrate some representative examples.

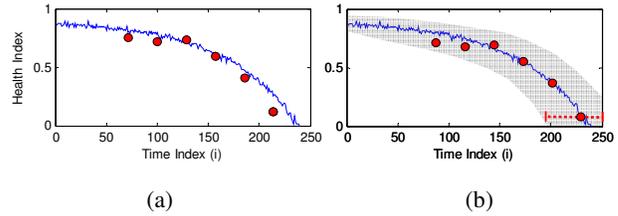


Figure 7(a) Predictions are made in the health domain for a single UUT. A health trajectory is predicted to consider evolution of fault in the system. (b) Predictions can be in the form of distributions with associated confidence bounds.

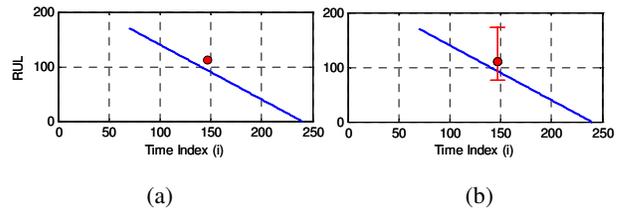


Figure 8 (a) Each prediction in health domain appears as a point prediction in the RUL domain, which then may be compared with ground truth (b) RUL predictions may be obtained with corresponding confidence limits.

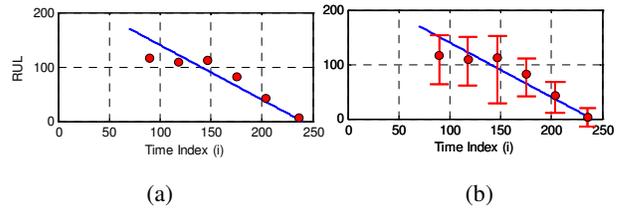


Figure 9 (a) A further assessment can be made on how well an algorithm’s RUL estimate evolves over time and converges to the true value as more data becomes available. (b) Such RUL trajectories may be accompanied by corresponding error bars as well.

**Prognostics Metrics**

**Computational Metrics**

Most of the publications in the area of prognostic algorithms for aerospace make no mention of computational performance. Many authors have been able to avoid the question of computational performance so far because they have not yet deployed their systems. We feel that assessing the computational performance of prognostic algorithms is very important, especially for applications that intend to monitor real-time data to make safety-critical decisions, such as deciding when it is necessary to shut down an engine or to land an aircraft to perform critical maintenance. In this section, we suggest several metrics that could be used to measure computational performance of prognostic algorithms,

all of which are already widely used to measure the computational performance of other types of algorithms.

In theoretical computer science, the computational complexity of algorithms is usually described using “Big O” notation [19]. This notation describes the amount of time needed for the algorithm to run, as function of the size of the input, and does so asymptotically, ignoring constant factors. For example, if the time performance of an algorithm is  $O(n^2)$ , then the time needed to run the algorithm increases quadratically with the size of the input. Big O notation allows the comparison of different algorithms to be independent from the particular software implementations and from the hardware on which the algorithms are run.

To measure the combined performance of an algorithm, its software implementation, and the hardware on which it is run, one can measure either central processing unit (CPU) time or elapsed time. CPU time measures the amount of time that the CPU spends executing the software, and does not include the time that the CPU spends running other software (in a time-shared system), or the time that the CPU spends waiting for input or output (I/O). The advantage of measuring CPU time instead of elapsed time is that it is more repeatable. Elapsed time (also known as “wall-clock time”) simply measures the amount of time that it takes for an algorithm to run, including I/O time. It is not appropriate to use elapsed time as a metric on a time-shared (multi-user) system, since in that situation the activities of other users can affect the elapsed time. CPU time and elapsed time are both appropriate for applications in which the prognostic algorithm is run in “batch mode” on recorded data. They can answer the question of whether the software will run fast enough to produce results within a reasonable amount of time.

For applications in which the data is processed in real-time, the more relevant question is whether or not the software can keep up with the real-time data stream. A metric that can be used to answer this question is how many samples per second the software (running on a particular hardware configuration) can handle. For example, an application may require the software to be able to process real-time sensor data at 100 samples per second (100 Hz).

Besides time, the other major consideration in computational performance is space. Often it makes sense to separately measure the amount of main memory [such as dynamic random access memory (DRAM)] used, and the amount of storage (such as disk space or flash RAM) used. In both cases, one can either report the asymptotic space complexity using Big O notation, or the number of bytes used by a particular

implementation. Space usage is particularly important in embedded applications, such as when the algorithm is run on the flight computer of an aircraft or spacecraft, since these on-board computers usually have very limited space available.

## Recommendations

A survey of wide variety of domains reveals that some metrics are common to most applications whereas some are very domain specific. In this section we pick metrics that we consider very relevant to prognostics and also make a recommendation for few new ones that evaluate several key aspects of prognostics.

As far as algorithmic performance metrics are concerned, the metrics based on accuracy and precision dominate the list. They provide the simplest assessment of prediction assessment. Of course, simple error measures can then be combined into more sophisticated ones to cater specific needs. Whereas in diagnostics the aim is to classify a fault or precursor of a fault, a prognostics problem tries to make a judgment about the remaining life of a component. Starting with the assumption that remaining life estimates will essentially never be completely on the mark, and using the fact that this is not required in most cases, the metric takes advantage of the acceptable tolerance around the actual remaining life. Here, one needs to keep in mind that the utility of the error is most often not symmetric with respect to zero (where the error is defined as the difference between actual remaining life and estimated remaining life). For instance, if the prediction is too early, the resulting early alarm forces more lead-time than needed to verify the potential for failure, monitor the various process variables, and perform a corrective action. On the other hand, if the failure is predicted too late, it means that this error reduces the time available to assess the situation and take a corrective action. The situation deteriorates completely when the failure occurs before a prediction is made that advises of critical system state. Therefore, given the same error size, it is in most situations preferable to have a positive bias (early prediction), rather than a negative one (late prediction). Of course, one needs to define a limit on how early a prediction can be and still be useful. Therefore, two different boundaries for the maximum acceptable late prediction and the maximum acceptable early one can be established. Any prediction outside of the boundaries will be considered either a false positive or a false negative. One can define the prediction error [20] as the difference between actual time to failure and predicted time to failure Figure 10.

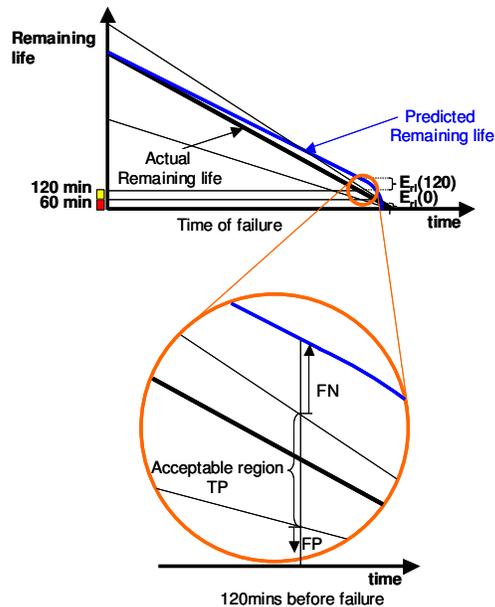


Figure 10. Conceptual prediction results and error assessment  
 In particular, focus will be on two instances of the error:

1.  $E(t_c)$  - prediction error at the time  $t_c$  when the critical zone (for example, within the next mission) is reached, and
2.  $E(t_{EOL})$  - prediction error at the time when the failure occurs.

Incorrect classifications are typically classified as false negatives (FN) and false positive (FP). In the context of late or early predictions, these categorizations are based on the magnitude of deviation from true time of failure. Therefore, one can define the following limits as the maximum allowed deviations from the origin:

**False Negatives** A prediction is considered a false negative if one fails to correctly predict a failure more than  $t_{fn}$  time units later than the actual time to failure, i.e.,  $E(t_c) < -t_{fn}$  time units. Note that a prediction that is late more than  $t_c$  time units is equivalent to not making any prediction and having the failure occurring.

**False Positives** A prediction is considered a false positive if we fail to correctly predict a failure if the prediction is more than  $t_{fp}$  time units earlier than the actual time to failure, i.e.,  $E(t_c) > t_{fp}$  time units. This is considered to be excessive lead time, which may lead to unnecessary corrections.

## Conclusions

In this paper we have provided a concise review of several domains and collected a variety of commonly used metrics to evaluate prediction performance. A list of concepts specific to CBM/PHM requirements has

been compiled and put these concepts into a notational framework to facilitate unambiguous descriptions. Several possible categorizations of these metrics have been formulated and provided to enhance the understanding of commonalities and differences between varied usages of similar methods. Towards the end some new metrics have been suggested that specifically cater to the PHM requirements. Although an effort has been made to cover most requirements, a further refinement in concepts and definitions is expected as prognostics matures. With this effort a discussion has been opened within the research community to standardize the performance evaluation of the prognostic systems.

## References

- [1] G. Vachtsevanos, F. L. Lewis, M. Roemer, A. Hess, and B. Wu, *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*, 1st ed. Hoboken, New Jersey: John Wiley & Sons, Inc., 2006.
- [2] I. T. Jolliffe and D. B. Stephenson, *Forecast Verification: A Practitioner's guide in Atmospheric Science*: Wiley, 2003.
- [3] G. W. Brier and R. A. Allen, "Verification of Weather Forecasts," in *Compendium of Meteorology*, T. F. Malone, Ed. Boston: American Meteorological Society, pp. 841-848.
- [4] M. Shchwabacher and K. Goebel, "A Survey of Data-Driven Prognostics," in *AAAI Fall Symposium*, Arlington, VA, 2007.
- [5] ISO, "Condition Monitoring and Diagnostics of Machines - Prognostics part 1: General Guidelines," in *ISO13381-1:2004(E)*. vol. ISO/IEC Directives Part 2, I. O. f. S. (ISO), Ed.: ISO, 2004, p. 14.
- [6] D. W. Brown, P. W. Kalgren, C. S. Byington, and R. F. Orsagh, "Electronic Prognostics - A Case Study Using Global Positioning System (GPS)," in *IEEE Autotestcon* Orlando, FL, 2005.
- [7] J. Gu, D. Barker, and M. Pecht, "Prognostics implementation of electronics under vibration loading," *Microelectronics Reliability*, vol. 47, 2007.
- [8] J. P. Hofmeister, P. L. T. Walter, D. Goodman, E. G. Ortiz, M. G. P. Adams, and T. A. Tracy, "Ball Grid Array (BGA) Solder Joint Intermittency: Detection: SJ BIST," in *IEEE Aerospace Conference* Big Sky, Montana, 2008.
- [9] D. L. Goodman, S. Wood, and A. Turner, "Return-on-Investment (ROI) for Electronic Prognostics in Mil/Aero Systems," in *IEEE Autotestcon* Orlando, FL, 2005.
- [10] S. Wood and D. Goodman, "Return-on-Investment (ROI) for Electronic Prognostics in High Reliability Telecom Applications," in *Annual International Telecommunications Energy Conference*, 2006.

- [11] S. Vohnout, "Electronic Prognostics System Implementation on Power Actuator," in *IEEE Aerospace Conference Big Sky*, Montana, 2008.
- [12] C. Wilkinson, D. Humphrey, B. Vermeire, and J. Houston, "Prognostic and health management for avionics," in *IEEE Aerospace Conference*, 2004.
- [13] C. Teal and B. Larsen, "Technology Update II: Wire Systems Diagnostics & Prognostics," in *Digital Avionics Systems Conference*, 2003.
- [14] N. Vichare, P. Rodgers, V. Eveloy, and M. Pecht, "Environment and Usage Monitoring of Electronic Products for Health Assessment and Product Design," *Quality Technology & Quantitative Management*, vol. 4, pp. 235-250, 2007.
- [15] N. Vichare and M. Pecht, "Enabling Electronic Prognostics Using Thermal Data," in *12th International Workshop on Thermal investigations of ICs*, 2006.
- [16] J. W. Simons and D. A. Shockey, "Prognostics Modeling of Solder Joints in Electronic Components," in *IEEE Aerospace Conference*, 2006.
- [17] L. J. Bond, S. R. Doctor, D. B. Jarrell, and J. W. D. Bond, "Improved Economics of Nuclear Plant Life Management," in *Second International Symposium on Nuclear Power Plant Life Management* Shanghai, China: IAEA, 2007, p. 26.
- [18] D. B. Jarrell, "Completing the Last Step in O&M Cost reduction," Pacific Northwest National Laboratory, p. 8.
- [19] A. V. Aho, J. D. Ullman, and J. E. Hopcroft, *Data Structures and Algorithms*: Addison Wesley, 1983.
- [20] K. Goebel and P. Bonissone, "Prognostic Information Fusion for Constant Load Systems," in *Proceedings of the 7th Annual Conference on Information Fusion*. vol. 2, 2005, pp. 1247 – 1255.
- [21] B. Ebert, et.al., "Forecast Verification - Issues, Methods and FAQ."
- [22] T. N. Palmer, A. Alessandri, U. Andersen, P. Cantelaube, M. Davey, P. De le cluse, M. De qu, E. Diez, F. J. Doblas-Reyes, H. Feddersen, R. Graham, S. Gualdi, J. F. Gueremy, R. Hagedorn, M. Hoshen, N. Keenlyside, M. Latif, A. Lazar, E. Maisonnave, V. Marletto, A. P. Morse, B. Orfila, P. Rogel, J. M. Terres, and M. C. Thomson, "Development of A European Multimodel System for Seasonal-to-Interannual Prediction (Demeter)," *Bulletin of the American Meteorological Society*, vol. 85, pp. 853-872, June 01, 2004 2004.

Table 2. List of metrics for algorithm performance evaluation

Metric Name	Definition	Description	Range of Values	Selected References
<b>Accuracy</b>				
Error	$\Delta^l(i) = r_*^l(i) - r^l(i)$			
Exponentially weighted error	$A(i) = \frac{1}{L} \sum_{t=1}^L e^{-\left  \frac{\Delta^l(i)}{D_0} \right }$	Weighs exponentially the errors in RUL predictions and averages over several UUTs; where, $D_0$ is a normalizing constant whose value depends on the magnitudes in the application.	[0,1]	[1]
Average Bias for $l^{\text{th}}$ UUT	$B_l = \frac{\sum_{i=P}^{EOP} \{\Delta^l(i)\}}{(EOP - P + 1)}$	Averages the errors in predictions made at all subsequent times after prediction starts. This metric can be further averages bias over all UUTs to establish overall bias.	[0,∞]	[1]
Timeliness	$A(i) = \frac{1}{L} \sum_{t=1}^L \phi\{\Delta^l(i)\}$ where, $\phi(x) = \begin{cases} \exp\{x/a_1\} - 1, & \text{if } x < 0 \\ \exp\{x/a_2\} - 1, & \text{if } x \geq 0 \end{cases}$ and $a_1 > a_2$	Exponentially weighs RUL prediction errors through an asymmetric weighing function. Penalizes the late predictions more than early predictions.	[0,∞]	[1]
Anomaly correlation coefficient (ACC)	$ACC = \frac{\sum (Z_F - Z_C)(Z_V - Z_C)}{\sqrt{\sum (Z_F - Z_C)^2 \sum (Z_V - Z_C)^2}}$ where, $Z$ is a weather variable (e.g. rainfall) and the subscripts $F$ , $V$ and $C$ denote forecast (prediction), validation dataset (ground truth) and climate (history data) respectively.	Measures correspondence or phase difference between forecast and observations, subtracting out the climatological mean at each point, rather than the sample mean values. The anomaly correlation is frequently used to verify output from numerical weather prediction (NWP) models. ACC is not sensitive to forecast bias, so a good anomaly correlation does not guarantee accurate forecasts. In the PHM context, it can be used to correct long term predictions using autocorrelation regression over a few time steps after prediction. However, the method requires computing a baseline from history data.	[-1 1] Perfect score = 1	[21]
False Positives	$FP(r_*^l(i)) = \begin{cases} 1 & \text{if } r_*^l(i) - r^l(i) > t_{FP} \\ 0 & \text{otherwise} \end{cases}$ where $t_{FP}$ = user defined acceptable early prediction	Assesses unacceptable early prediction of the predictor at specified time instances. User must set acceptable range for prediction	[0,∞]	[20]
False Negatives	$FN(r_*^l(i)) = \begin{cases} 1 & \text{if } r^l(i) - r_*^l(i) > t_{FN} \\ 0 & \text{otherwise} \end{cases}$ where $t_{FN}$ = user defined acceptable late prediction	Assesses unacceptable late prediction of the predictor at specified time instances. User must set acceptable range for prediction	[0,∞]	[20]

**Robustness**

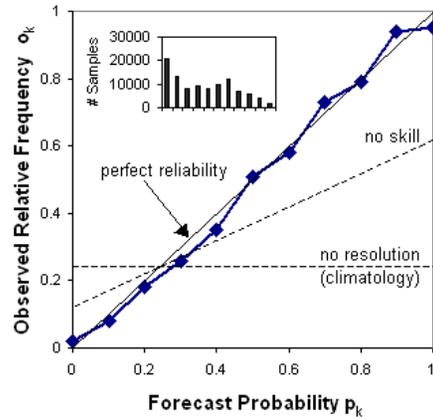
Sensitivity

$$S(i) = \frac{1}{L} \sum_{i=1}^L \left\{ \frac{\Delta M^l(i)}{\Delta_{input}} \right\}$$

Measures how sensitive a prognostic algorithm is to changes in input changes or external disturbances. Can be assessed against any performance metric of interest.  $\Delta M$  is the distance measure between two successive outputs for metric M's value and  $\Delta_{input}$  is distance between two successive inputs.

[0,1]

Reliability diagram

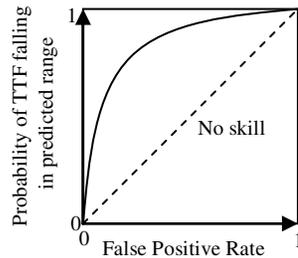


$\sum_k |o_k - p_k|$  is a measure of the deviation from the diagonal and can be used as a score.

The reliability diagram plots the observed frequency against the forecast probability, where the range of forecast probabilities is divided into K bins (for example, 0-5%, 5-15%, 15-25%, etc.). The sample size in each bin is often included as a histogram or values beside the data points. Reliability is indicated by the proximity of the plotted curve to the diagonal. The deviation from the diagonal gives the conditional bias. If the curve lies below the line, this indicates overforecasting (probabilities too high); points above the line indicate underforecasting (probabilities too low). The flatter the curve in the reliability diagram, the less resolution it has. This metric is useful in comparing RUL predictions made for a batch of systems based on the aggregate health indicators for the group.

[0 0.5]  
Perfect score = 0 [21]

Relative Operating Characteristic (ROC)



The area under the ROC curve can be used as a score.

ROC measures the ability of the forecast to discriminate between two alternative outcomes, thus measuring resolution. It is not sensitive to bias in the forecast, so says nothing about reliability. A biased forecast may still have good resolution and produce a good ROC curve, which means that it may be possible to improve the forecast through calibration. The ROC can thus be considered as a measure of potential usefulness.

[0 1]  
Perfect score = 1 [21, 22]

Table 3. List of metrics based on economic aspect of prognosis

Metric Name	Definition	Description	Range of	Selected
-------------	------------	-------------	----------	----------

			Values	References
<b>Cost/Benefit</b>				
Return on Investment (ROI)	gain/investment	An investment in prognostics is expected to save money on maintenance and possibly prevention of downtime or lost hardware over the life of the system. The <i>gain</i> is the amount of money saved as a result of using prognostics, and the <i>investment</i> is the cost of developing and installing the prognostic system. The ROI (which is usually annualized) can be seen as the interest rate that a bond would have to pay to provide the same financial return. An investment should only be made if its ROI is at least as high as those of other potential investments with similar risk.	$[-\infty, \infty]$	[9-11]
Life-cycle cost	acquisition cost + operations cost	As a metric, compare the life cycle cost of the system (which includes the cost of building it or acquiring it and the cost of operating it) with and without prognostics. Total Value is the change in life cycle cost. ROI will be positive if adding prognostics reduces life cycle cost.	$[0, \infty]$	[12]
MTBF/MTBUR ratio (mean time between failure / mean time between unit replacement)	MTBF/MTBUR	This metric measures the ratio between how long a component lasts and how long it is used before replacing it. Prognostics should enable the reduction of this ratio by allowing components to be used longer, until they are closer to failure, which would save money.	$[0, \infty]$	[13]

Table 4. New performance metrics suggested for prognostics in CBM/PHM domain

Metric Name	Definition	Description	Range of Values	Related References
<b>New Metrics for Prognostics</b>				
Prognostic Horizon	$H(i)=j-i$	This metric is mentioned in the “Electronics Prognostics R&D Needs Definition” presentation, but not explicitly defined. We suggest the following definition: Prognostic Horizon is the difference between the current time index $i$ and the largest time index, $j$ , that the algorithm can make a prediction for, provided the data accumulated up to the time index $i$ .		
Feature Set Sensitivity	$FSS(i, f') = \left  \frac{M(i, f) - M(i, f')}{M(i, f)} \right ,$ where $f' \subset f$ - a subset of the original feature set $f$ and $M$ is a performance metric of interest	Calculates the effect of an arbitrarily reduced feature set on a metric $M$ .		

<p>Sampling Rate Sensitivity</p>	$SRS(\omega_{reference}, \omega) = \frac{1}{L} \sum_{i=1}^L \left\{ \frac{\min(M(i, \omega_{reference}), M(i, \omega))}{\max(M(i, \omega_{reference}), M(i, \omega))} \right\}$	<p>The metric estimates sensitivity of another metric (accuracy, timeliness, etc) to the changes in the data set sampling frequency. The estimate is done using a reference frequency that can, for example, be the recommended design frequency for the particular algorithm</p> <p>[0,1]</p>
<p>Data Set Equivalency</p>	$DSE(A, B) = \frac{1}{N} \left( 1 - \frac{ f_A \setminus f_B }{M} \right) \left( \sum_{n=1}^N \delta L \cdot \delta \omega \cdot \delta_h \cdot \delta_l \cdot \delta_{SD} \right)$ <p>where,</p> $M = \max( f_A ,  f_B )$ $N = \min( f_A ,  f_B )$ $\delta L = \frac{ L_{An} - L_{Bn} }{\max(L_{An}, L_{Bn})} \text{ - Feature length difference coefficient}$ $\delta \omega = \frac{ \omega_{An} - \omega_{Bn} }{\max(\omega_{An}, \omega_{Bn})} \text{ - Sampling rate difference coefficient}$ $\delta_{highest} = \frac{ \max(f_{An}) - \max(f_{Bn}) }{\max(\max(f_{An}), \max(f_{Bn}))} \text{ - Highest value difference coefficient}$ $\delta_{lowest} = \frac{ \min(f_{An}) - \min(f_{Bn}) }{\min(\min(f_{An}), \min(f_{Bn}))} \text{ - Lowest value difference coefficient}$ $\delta_{SD} = \frac{ SD(f_{An}) - SD(f_{Bn}) }{\max(SD(f_{An}), SD(f_{Bn}))} \text{ - Standard deviation difference coefficient}$	<p>This metric is suggested for use when two or more different datasets are employed to train or evaluate the prognostic system and the equivalency of these sets needs to be estimated. This may become relevant when the data is obtained on differently configured equipment or from related, but sufficiently distinct fields (e.g. the same engine type deployed on a military transport versus a civilian airliner).</p> <p>Feature length, sampling rate, standard deviation, and min/max values are the suggested components for this metric; this can, however, be tailored to the specific application.</p> <p>This metric can be used as a part of the more encompassing Experiment Equivalency Metric or in conjunction with other metrics in this table - to provide a quantitative assessment of the data sources used to calculate them.</p>
<p>History Size Required</p>	$HS=n; n \in (1, \infty)$	<p>Indicates how many consecutive sets of feature values are required to be known at any given time for the algorithm to function properly</p> <p>[1,∞]</p>

