RELIABILITY PAPER

Designing predictive maintenance systems using decision tree-based machine learning techniques

Shashidhar Kaparthi and Daniel Bumblauskas
Department of Management, University of Northern Iowa, Cedar Falls, Iowa, USA

Received 25 April 2019
Revised 18 August 2019
27 October 2019
20 December 2019
Accepted 28 December 2019

Abstract

Purpose – The after-sale service industry is estimated to contribute over 8 percent to the US GDP. For use in this considerably large service management industry, this article provides verification in the application of decision tree-based machine learning algorithms for optimal maintenance decision-making. The motivation for this research arose from discussions held with a large agricultural equipment manufacturing company interested in increasing the uptime of their expensive machinery and in helping their dealer network.

Design/methodology/approach – We propose a general strategy for the design of predictive maintenance systems using machine learning techniques. Then, we present a case study where multiple machine learning algorithms are applied to a particular example situation for an illustration of the proposed strategy and evaluation of its performance.

Findings – We found progressive improvements using such machine learning techniques in terms of accuracy in predictions of failure, demonstrating that the proposed strategy is successful.

Research limitations/implications – This approach is scalable to a wide variety of applications to aid in failure prediction. These approaches are generalizable to many systems irrespective of the underlying physics. Even though we focus on decision tree-based machine learning techniques in this study, the general design strategy proposed can be used with all other supervised learning techniques like neural networks, boosting algorithms, support vector machines, and statistical methods.

Practical implications – This approach is applicable to many different types of systems that require maintenance and repair decision-making. A case is provided for a cloud data storage provider. The methods described in the case can be used in any number of systems and industrial applications, making this a very scalable case for industry practitioners. This scalability is possible as the machine learning techniques learn the correspondence between machine conditions and outcome state irrespective of the underlying physics governing the systems.

Social implications – Sustainable systems and operations require allocating and utilizing resources efficiently and effectively. This approach can help asset managers decide how to sustainably allocate resources by increasing uptime and utilization for expensive equipment.

Originality/value – This is a novel application and case study for decision tree-based machine learning that will aid researchers in developing tools and techniques in this area as well as those working in the artificial intelligence and service management space.

Keywords Classification techniques, Machine learning, Maintenance, Reliability, Predictive maintenance systems

Paper type Research paper

Introduction and motivation

According to the "State of Service Management - 2015" report by the Pinder and Aberdeen Group (2015), after-sale service industry contributes over 8 percent to the US GDP. The most frequently cited KPIs of service success are customer satisfaction, service profitability, first-time fix rate, service revenue, SLA/contract compliance rate, service costs, customer retention, and serviceable asset uptime/availability. We are at the precipice of a monumental shift in the way we think about performing service and maintenance. Sophisticated maintenance decision-making models are now making use of machine learning (ML) and

International Journal of Quality & Reliability Management
Vol. 37 No. 4, 2020
pp. 659-686

0265-671X
DOI 10.1108/IJQRM-04-2019-0131

artificial intelligence (AI) design methods. The abundance of big data sources available for analysis is unprecedented in human history, often leading to the use of cyber-physical modeling (Bumblauskas *et al.*, 2017a) and paralysis in decision-making (Bumblauskas *et al.*, 2017b). Concurrently, analytical techniques making use of ML, AI, and more traditional operations research & industrial engineering techniques are being widely adopted to solve problems across various industries.

The motivation for this research arose from discussions held with a large agricultural equipment manufacturing company interested in increasing the uptime of their expensive equipment in the field. In this article, we provide a framework for using ML to predict failure in such industrial systems. A case is provided for a cloud data storage provider; however, the methods used in the case can be applied to any number of systems and industrial applications, making this a very scalable case for industry practitioners.

More specifically, contemporary industrial production systems have made use of maintenance practices that have evolved over time. Time-based maintenance was one of the first methodologies used during and post-industrial revolution, followed by condition-based maintenance in which analog signals were used to assess the operating conditions to more accurately predict maintenance intervals and frequency. With the advent of sensory technologies, we can now collect data on parameters such as tool wear, temperature, and remaining useful life, which drive condition and reliability-centered maintenance programs (Moubray, 1997; Schlabbach and Berka, 2001).

One of the overarching concepts inherent to all of these maintenance programs is preventive or preventative versus predictive maintenance decision-making. The theory is that we should be performing maintenance preventatively to avoid unplanned outages or downtime. The drawback to this can be less than optimal operations & maintenance (O&M) budgeting and spending as we are trying to prevent a potential failure more proactively, which might not be required and could make the operating condition worse; thus, if it is not broke, don't fix it, and so on. This also conflicts with the desire to operate in a sustainable fashion by overusing resources. This somewhat flawed logic can lead to O&M decision-making that is not in the best interest of the asset owner. To be fair, in mission-critical applications where lives and injuries are at stake, this might need to be deployed (e.g., commercial aviation). However, it also leads to less than optimal maintenance and supply chain management decision-making by *over-maintaining* assets.

In this article, we have utilized an ML approach to help improve decision-making of possible failure events and associated maintenance actions. We have illustrated in our test case that using machine-learning techniques can improve the accuracy of event decision-making, that is, using machine-learning techniques can help us figure out what is the likelihood of a failure event leading to a maintenance decision. The goal of this work is to continue the contribution toward better predictive maintenance practices in industrial systems. The types of questions that this work can help answer include:

(1) When should I perform maintenance (time-based, condition-based, or predictive)?

(2) What type of maintenance should I do (e.g., minor, major)?

(3) Can I predict when maintenance should be performed using statistics and mathematical modeling?

This article provides a review of the literature and provides a system-theoretic-based framework for understanding past research in predictive maintenance. Following the literature review, a brief overview of decision tree-based ML techniques is presented. A design and operation strategy for such systems is presented next, followed by the application of this strategy as illustrated using a case study with data from a cloud storage provider. Finally, a final discussion and conclusions are provided.

Literature review and framework

Madu (2000) outlines the importance of businesses and organizations creating a competitive advantage by effective maintenance practices. Others have detailed the importance of using statistical analysis to determine maintenance decisions (Lawrence *et al.*, 1995), and numerous industry applications exist for systems such as repairable and non-repairable systems (Singh Jolly and Jit Singh, 2014; Settanni *et al.*, 2016), jet engines (Settanni *et al.*, 2015), electrical equipment (Bumblauskas *et al.*, 2012, 2017a; Bumblauskas, 2015; Bumblauskas *et al.*, 2015; Chan and Asgarpoor, 2006; Yam *et al.*, 2001), gas distribution equipment (Pievatolo and Ruggeri, 2004), agricultural equipment (Bumblauskas *et al.*, 2015), healthcare (Eker *et al.*, 2012), and building systems (Wong and Li, 2008). Today, post-industrial revolution, we seemingly use terms such as industry 4.0 and industry 5.0 interchangeably to denote the use of cyber-physical systems (CPS), which is well documented (Bumblauskas *et al.*, 2017a; Herterich *et al.*, 2015; Lee *et al.*, 2015a, 2015b), industrial CPS (ICPS), safety (Wang *et al.*, 2016), ML, and AI to represent work such as that detailed in this article.

The use of decision support systems, including the use of artificial neural networks (Yam *et al.*, 2001) and analytical hierarchy process models (Bumblauskas *et al.*, 2017a; Wong and Li, 2008), for maintenance applications has also been well documented. Condition-based maintenance programs have also been used to predict failure and optimization of maintenance decision-making (Yang, 2003; Lu *et al.*, 2007). Based on the foundational research and models, a paradigm-shifting new framework can be deployed.

We first propose a framework for describing a Predictive Maintenance System (PdMS) using a systems theory perspective. Then, we use this framework to categorize previous research in the area. A PdMS is a set of interrelated components that measure one or several input variables and process and transform these measurements to provide insights into the likelihood of future states of the system occuring. This is useful to provide identification of corrective actions that may be taken in case an undesirable future state is foreseen. After the corrective action is taken, the input measurements get influenced and the process gets repeated. Figure 1 summarizes this framework. As also seen in Figure 1, this system is woven together by a supporting infrastructure within the context of an enveloping environment. Both the system design and the system operation are of interest.

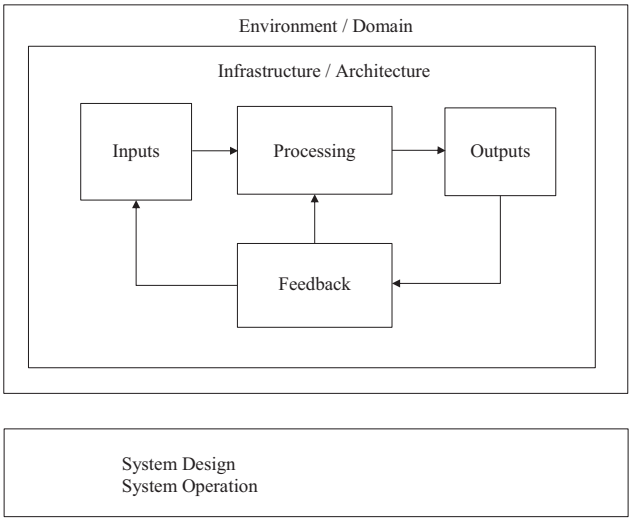


Figure 1.
A systems-theoretic framework for predictive maintenance

A search was conducted in the ABI/INFORM database of periodicals for all articles with titles that include the term "predictive maintenance." The search resulted in 26 articles that were published during the period 1992–2017. The journals that published research in this area included *Computers & Industrial Engineering, International Journal of Production Economics, International Journal of Production Research, and Journal of Intelligent Manufacturing* among others. Table I summarizes the results of this search.

As seen in Table I, the researched systems covered applications in manufacturing, construction, electrical, graphic design, and healthcare industries, as well as general models applicable to many types of scenarios. The supporting infrastructure is discussed in terms of the requirements for the database design, the software necessary, the integration with enterprise information systems, and the advantages of implementing these systems using cloud computing services. Several best practices and strategies for designing and evaluating PdMS are also discussed. The inputs could be just one or many variables, discrete or continuous measures, slow-moving or fast-moving values, real-time or lagged measurements, and mechanical or physical-chemical parameters. The outputs included diagnosis of mechanical health and degradation, the optimal time for maintenance, the replacement decision, reliability, quality, availability or downtime, and cost implications. The processing was in terms of simulation modeling, forecasting techniques, root cause analysis, cost minimization, spectral analysis, solutions to integral equations, factor analysis, analytic hierarchy process, if-then rules, Markov chain models, Bayesian network models, quality deviation index, and stochastic linear degradation models.

Decision tree-based classification techniques

ML for classification involves processing data samples or observations from the past, to induce the correspondence between one or more input variables and an output or target variable. Typically, the target variable is discrete and is either two or more values. The input variables could be discrete or continuous. Decision tree-based classification techniques are a subset of such methods. In a seminal article titled "Induction of Decision Trees," Quinlan (1986) traces the history of decision tree-based classification techniques and describes the ID3 algorithm that has led to the development of its modern versions. Quinlan (1986) uses Carbonell *et al.'s* (1983) framework for classifying machine-learning techniques based on three principal dimensions:

(1) The underlying learning strategies used;

(2) The representation of knowledge acquired by the system; and

(3) The application domain of the system.

Decision tree-based classification techniques represent the underlying knowledge induced, by learning from examples, in the form of decision trees. An example of a decision tree is shown in Figure 2. It shows the knowledge induced by learning the relationship between the input variables: the presence of an international plan and the number of customer support calls made, and the output variable: the customer churn, in a cellular service provider.

Decision tree induction algorithms first partition the entire data set into two or more sets based on one of the input variables and its values by optimizing a metric that minimizes within-partition target value differences and maximizes between-partition target value differences. The metric measures the impurity of the identification done by the decision tree, and the methods seek to minimize the impurity. In the example illustrated in Figure 2, the best partitioning is done based on the values of the international variable. The first subset consists of all the observations where the value of international variable is "yes," and the second subset consists of all observations where the value of the international variable is "no." The

#	Authors	Inputs	Processing	Outputs	Infrastructure/Architecture	Environment/Domain
1	Okogbaa *et al.* (1992)	–	• Simulation modeling • Opportunistic	–	• Database design	• General
2	Pitt, T.J. (1997)	–	• Forecasting techniques	–	• Information technology	• General
3	Edwards *et al.* (1998)	• Machine conditions	• Root cause analysis	• Mechanical health	–	• Construction industry
4	Chu *et al.* (1998)	• One variable system	• Cost minimization	• Replacement decision	–	• General
5	Parrondo *et al.* (1998)	• Fluid-dynamic conditions	• Spectral analysis	• Diagnosis	–	• Centrifugal pump
6	Allele *et al.* (2002)	• Reliability function	• Bayesian discriminant analysis	• Optimal time for maintenance	–	• HV electrical circuit breakers
7	McKone, K.E. and Weiss, E.N. (2002)	–	• Minimize costs • Renewal theory • Solutions to integral equations	–	–	• Manufacturers
8	Carnero Moya, M.C. (2004)	–	–	–	• Predictive maintenance program strategy	• Industrial plants
9	Carnero, M.C. (2005)	• Lubrication • Vibration	• Several diagnostic techniques • AHP • Factor analysis	–	–	• Screw compressors
10	Raouf, A. (2005)	–	–	–	–	• Book review
11	Huang *et al.* (2005)	• Factory floor	• Watchdog agent	• Near-zero downtime	• Information technology • Enterprise systems • D2B (device-to-business)	• Case study
12	Flores-Colen *et al.* (2006)	• Several parameters	• In situ tests	• Degradation state	–	• Construction • Façades
13	Man *et al.* (2006)	• Real-time sensors	• Several diagnostic techniques	• Health condition	• SIMAP (intelligent system for predictive maintenance)	• Wind turbine gearbox
14	Simeu-Abazi, Z. and Bouredji, Z. (2006)	• Any variable	• Timed automata	• Degradation	–	• Industrial manufacturing system simulation
15	Crowder, M. and Lawless, J. (2007)	• Wear with random effect	• Optimization	• Cost	–	• General system
16	Moya, M.D.C.C. (2007)	• Vibration • Lubricant	• Vibration analysis • Lubricant analysis • Integrated analysis	• Security • Quality • Availability	• Best technique selection	• Food plant • Pharmaceutical plant

(*continued*)

Table I.
Systems perspective analysis of past research

Table I.

#	Authors	Inputs	Processing	Outputs	Infrastructure/Architecture	Environment/Domain
17	Kumar *et al.* (2009)	• Condition data	• Hierarchical fuzzy inference	• Reliability • Degree of certainty	–	• Large electric motor
18	Magro, M.C. and Pinceti, P. (2009)	• Intelligent field devices	• Rough set theory • If-then rules	• Prediction accuracy	–	• Intelligent pressure transmitter
19	Flores-Colen *et al.* (2011)	• 23 mechanical and physical-chemical parameters	• In situ tests • Lab tests	• Degradation state	–	• Construction • Façades
20	E Costa *et al.* (2012)	–	–	–	• Multi-criteria model for evaluation of predictive maintenance programs	• General hospital
21	Guilherme, F.T. and Miguel, A.S. (2015)	–	• Vibration analysis • Thermography • Ferrography • Inspection sensitive	• Availability • Costs	–	• Graphics department of a company
22	Wen *et al.* (2016)	–	–	–	• Economic production quantity model enhanced with predictive maintenance strategy	• General
23	Wang *et al.* (2017)	• Mobile agents	• General algorithms for analysis	• Reliability • Availability • Adaptability • Safety	• Cloud computing	• Manufacturing industry • Motor tested system
24	Lee, D. and Pan, R. (2017)	• Onboard sensors	• Markov chain models • Bayesian network models	• Reliability	–	–
25	You, M. (2017)	• Condition data	• Stochastic linear degradation model	• Degradation • Sudden damage	–	• Numerical simulation
26	He *et al.* (2017)	• Key process variables	• Quality deviation index • Minimization of total cost	• Reliability • Quality	–	• Manufacturing

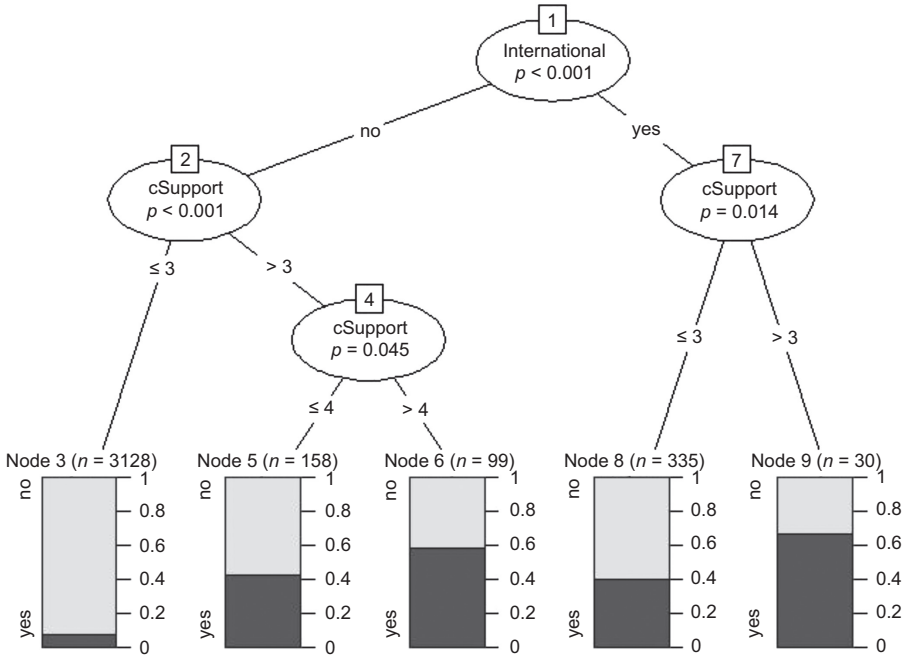


Figure 2.
A decision tree

process of dividing observations into just two subsets is referred to as binary partitioning. The algorithms then proceed in a recursive fashion and further subdivide the two subsets by optimizing the same metric. The recursive partitioning stops when the metric cannot be improved much. The decision trees are then pruned to prevent overfitting of the data and to enable generalization of the results.

In a comprehensive survey (Grąbczewski, K., 2014) of several decision tree induction techniques, Grąbczewski examines the ID3, CART, C4.5, Cal5, FACT, QUEST, CRUISE, CTree, SSV, ROC-Based Trees, LMDT, OC1, LTree, QTree, LgTree, DT-SE, DT-SEP, DT-SEPIR, and LDT. He attributes the differences in the techniques to the following mechanisms:

(1) Whether one variable or more are used in the nodes of the tree for conditions for branching;

(2) Whether the number of branches at the nodes is two or more;

(3) Whether an exhaustive search is done for finding the best split or a statistical sampling is done;

(4) Whether the technique for finding the best split looks ahead or not at what the sub-splits might contribute to the impurity reduction;

(5) Whether the variable for the split is chosen first before choosing the criteria for the splitting or both are done at the same time;

(6) The stopping criteria used;

(7) Whether pruning is needed after the tree generation or not; and

(8) Several other ways.

In this study, we focus on the Conditional inference Tree (CTree) technique. Hothorn *et al.* (2006, 2018) describe the CTree technique as follows: CTree is a statistical approach to recursive partitioning. Formal hypothesis tests for both variable selection and stopping criterion to ensure unbiased results. This choice leads to tree-structured regression models for all kinds of regression problems, including models for censored, ordinal, or multivariate response variables. Because well-known concepts are the basis of variable selection and stopping criterion, the resulting models are easier to communicate to practitioners. Simulation and benchmark experiments indicate that conditional inference trees are well-suited for both explanation and prediction. Another advantage of using the CTree technique is that there is no need for pruning the decision tree.

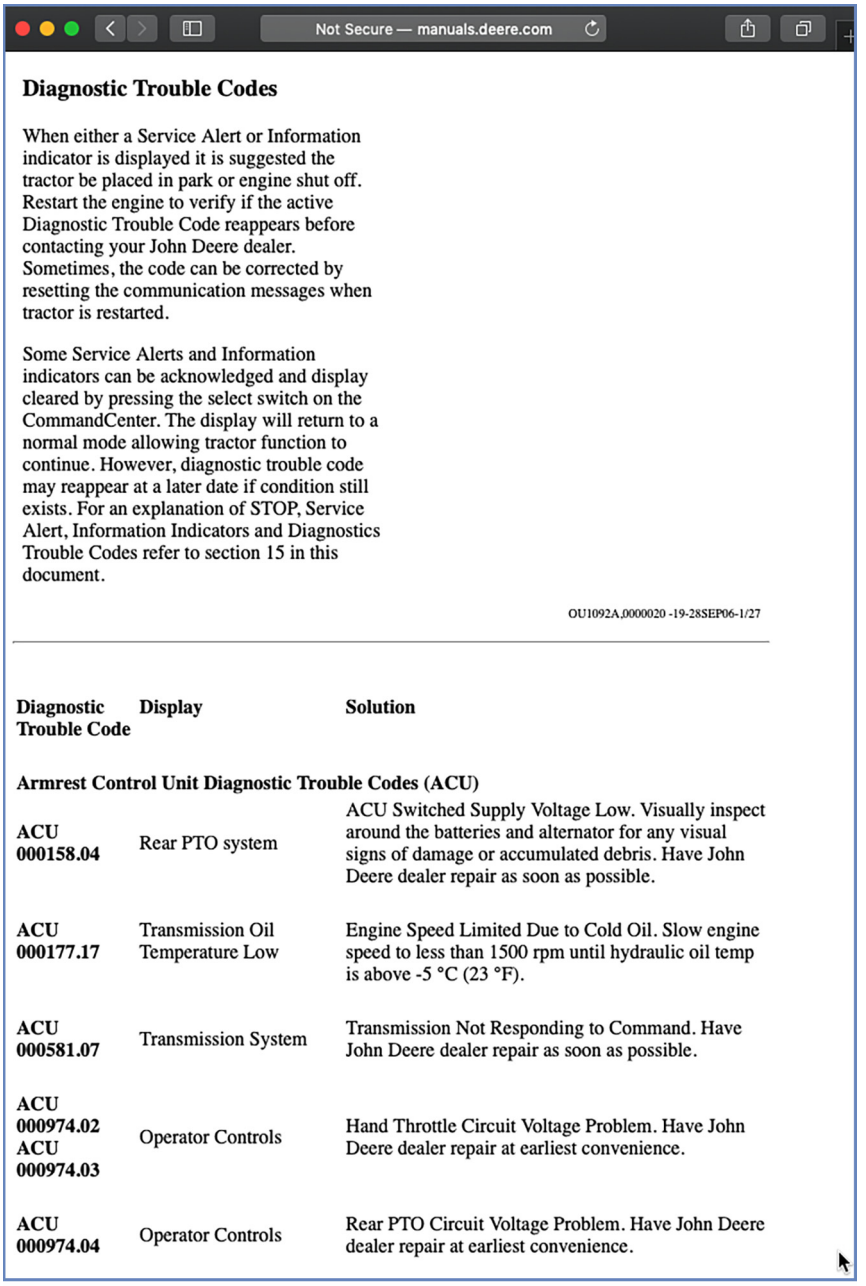
The ability of classification techniques to make predictions on future or unseen observations, that is, the generalization ability is important for practical applications. Overfitting is the phenomenon when predictions on training data (data used for learning) are highly accurate, but predictions on testing data (unseen data that have not been used for learning) are not very accurate. Overfitting reduces the generalization capability of the techniques. One way to improve the generalization capability and reduce overfitting is by the use of ensemble techniques. Ensemble techniques generate multiple decision trees and select the final outcome by a voting mechanism. Random forest is one such ensemble technique.

Random forest technique (Breiman, 2001) works by randomly selecting a sample of the observations and a random subset of the input variables to generate a decision tree. This generation is similar to the decision tree-based classification techniques mentioned earlier. A second tree is then generated by another random sampling of observations and features. The process is repeated, and many trees are generated resulting in a forest of decision trees. The output value is then identified by a voting mechanism based on the outputs of each of the decision trees in the forest. The generalization ability of the forest converges due to the law of large numbers, and the stopping criteria for building the trees depends on when there are negligible improvements in the generalization capability. In addition to the CTree technique, we also use the random forest technique is this study.

Proposed methodology for a predictive maintenance system (PdMS)

Heavy industrial equipment like agricultural equipment, construction and earth-moving equipment, and electrical equipment are complex systems with many subsystems. For meeting regulatory requirements and for assisting customers and dealers with service management, all such machinery is equipped with extensive onboard diagnostic (OBD) systems that monitor machine conditions and report diagnostic trouble codes (DTCs). Over the years, many standards have been developed to streamline this process among manufacturers and supporting device vendors. OBD-II, HD-OBD, SAE J-1962, and ISO 14230 are some of the standards for diagnostic trouble codes (Wikipedia, 2019). Even though the origin of such systems is in the automobile industry, all heavy equipment manufacturers use such DTC codes. For example, Figure 3 shows a screenshot of DTC codes from an online John Deere tractor manual.

As seen in Figure 3, the first set of DTC codes are those displayed in the Armrest Control Unit subsystem. The other subsystems in the entire system include Front Brake Control Unit, Cab Control Unit, Central Control Unit, Cab Load Center, Engine Control Unit, Hitch Control Unit, Instrument Control Unit, Transmission Control Unit, Hydraulic System, and Suspension Control Unit as seen in the rest of the online manual. A DTC is fired (turned on) when a particular subsystem or a component has a fault. The manual specifies the solution procedure or the action to be taken when a DTC is fired. Several of these situations involved stopping the use of the equipment and scheduling a repair call from the dealer. Whether a DTC is fired or not is a binary state variable {0,1} in our proposed methodology.



Not Secure — manuals.deere.com

Diagnostic Trouble Codes

When either a Service Alert or Information indicator is displayed it is suggested the tractor be placed in park or engine shut off. Restart the engine to verify if the active Diagnostic Trouble Code reappears before contacting your John Deere dealer. Sometimes, the code can be corrected by resetting the communication messages when tractor is restarted.

Some Service Alerts and Information indicators can be acknowledged and display cleared by pressing the select switch on the CommandCenter. The display will return to a normal mode allowing tractor function to continue. However, diagnostic trouble code may reappear at a later date if condition still exists. For an explanation of STOP, Service Alert, Information Indicators and Diagnostics Trouble Codes refer to section 15 in this document.

OU1092A,0000020 -19-28SEP06-1/27

Diagnostic Trouble Code	Display	Solution
Armrest Control Unit Diagnostic Trouble Codes (ACU)		
ACU 000158.04	Rear PTO system	ACU Switched Supply Voltage Low. Visually inspect around the batteries and alternator for any visual signs of damage or accumulated debris. Have John Deere dealer repair as soon as possible.
ACU 000177.17	Transmission Oil Temperature Low	Engine Speed Limited Due to Cold Oil. Slow engine speed to less than 1500 rpm until hydraulic oil temp is above -5 °C (23 °F).
ACU 000581.07	Transmission System	Transmission Not Responding to Command. Have John Deere dealer repair as soon as possible.
ACU 000974.02 ACU 000974.03	Operator Controls	Hand Throttle Circuit Voltage Problem. Have John Deere dealer repair at earliest convenience.
ACU 000974.04	Operator Controls	Rear PTO Circuit Voltage Problem. Have John Deere dealer repair at earliest convenience.

Source: http://manuals.deere.com/omview/OMAR232106_19/OU1092A_0000020_19_28SEP06_1.htm

Figure 3.
Screenshot of a John Deere tractor manual showing diagnostic trouble codes

Our proposed methodology described in the rest of this section answers the following question: how can we design a system that will predict whether a DTC will fire or not based on machine conditions? If we are able to predict that a DTC will fire, then the corrective action can be scheduled before the DTC is fired. This can be done while operating the equipment at the same time and increasing uptime as a consequence. Prioritizing critical DTC codes for predictive maintenance is the first step in implementing our proposed methodology.

Our proposed methodology for a PdMS for each of the critical DTC codes is a two-phase process. The first phase is the design of the PdMS, and the second phase is its use and operation. There are several tasks in each of these phases. Figure 4 depicts the tasks involved in this system design phase. Onboard sensors sample machine conditions at regular intervals and transmit data to central servers. With the technological advances in the field of IoT (Internet of Things) sensors and cellular communication networks (e.g., 4G LTE and 5G), the machine conditions for equipment that is operating in the field can be monitored from a remote location. The measurement data are envisioned here as a time-series data stream. This is a series of values arranged chronologically and arriving in real-time as and when the conditions are measured and transmitted by the sensors. Other onboard sensors are also monitoring and transmitting the state of the equipment as to whether it is operating normally according to specifications or otherwise. The state of the subsystem is represented as binary digits {0,1} as to whether a DTC has fired (1) or the DTC has not fired (0). Recording this system state is the next task. An organizational prerequisite to this methodology is that products are engineered with onboard sensors with transmission capability and the IT infrastructure is configured according to these requirements.

The next task in the design phase is to process these data streams and calculate the features. Features are summary statistics that capture the characteristics of the data stream. The features extraction task is done by the transformation of the time-series data stream into a smaller vector of values (F) that capture its key characteristics. We also refer to these data streams as signals.

Figure 5 shows a typical signal from an input sensor. The top chart in Figure 5 plots the time of measurement on the horizontal axis and the signal-measure on the vertical axis. At the bottom of Figure 5, two features of the signal are annotated. For a given window length, the mean value of all the measurements is the first feature, and the range of values is the second feature. The first feature is a measure of central tendency, and the second one is that of dispersion. Other features could be utilized depending on the situation like the maximum or the minimum value within the window. Proxies for first- and second-order derivatives for measuring velocity and acceleration of input value changes are other possibilities.

Data have to be collected in the field long enough to get sufficient observations depicting all possible system states. Once sufficient data are collected, the next step is to extract

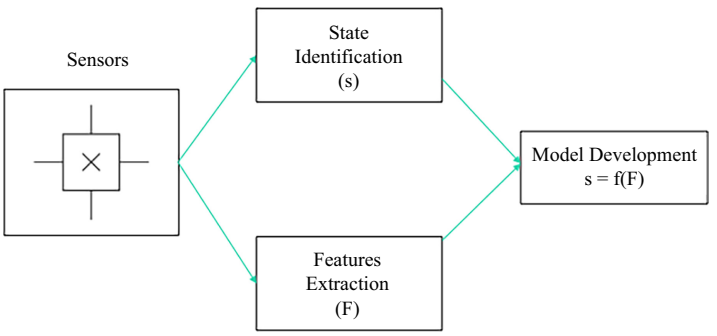


Figure 4.
System design phase

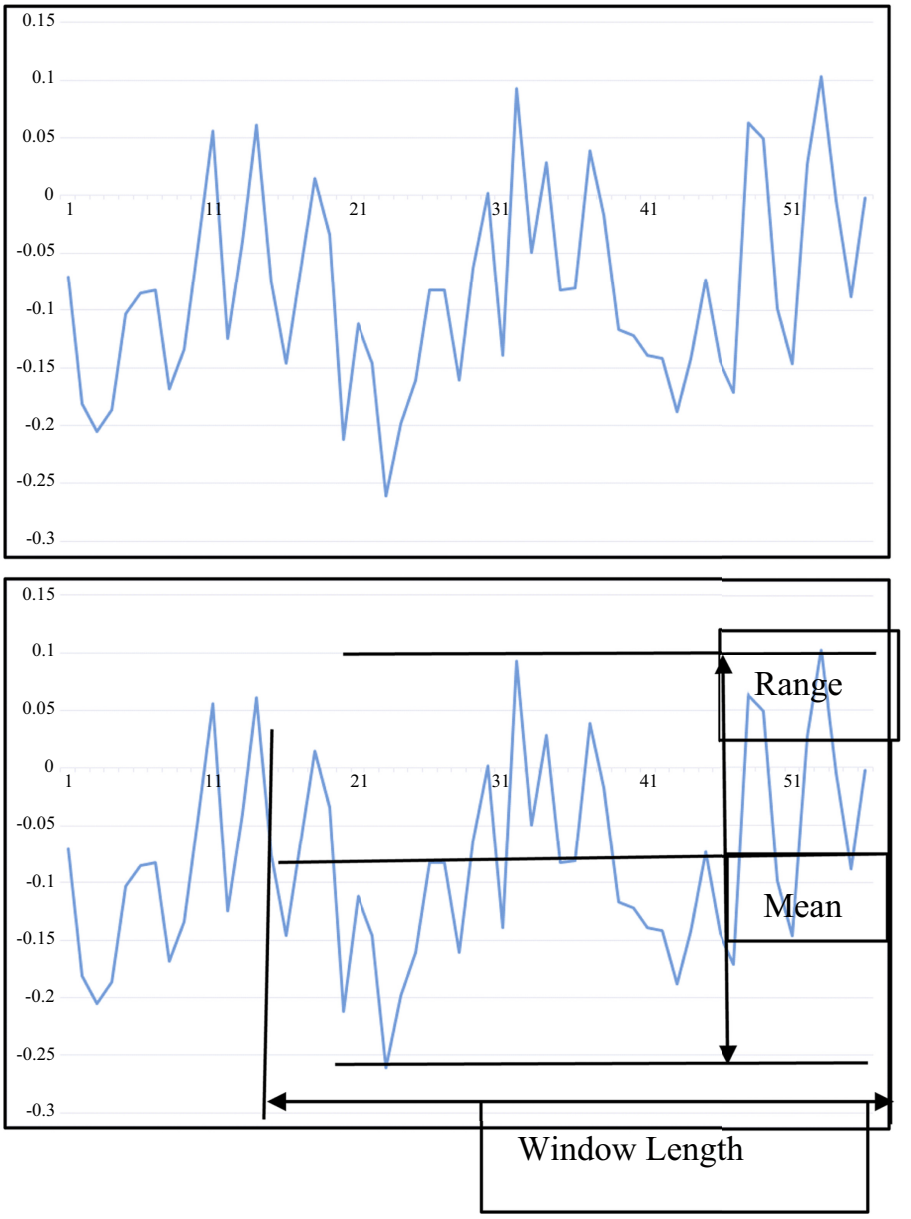


Figure 5.
Signal from an input sensor and some of its features

features from the signals and create data records containing these features and the associated state of the system. These data records are then processed, as described in the decision tree-based classification section, to induce the knowledge (model) for finding the relationships between the machine conditions and the state of the system. This is the model development stage of our proposed methodology. In this study, this knowledge is represented as a decision

tree as well as a forest of decision trees. When the CTree technique for decision tree induction is used, the model can be represented as shown in Figure 2 in the previous section. When the random forest technique is used, the knowledge is captured as a set of such trees that are polled for the final result.

To test the generalizability of the model, the ML of the relationship between the input variables and the state is done by dividing the data set into two subsets. One subset, the training data set, is used for inducing the knowledge or ML. The other subset, the testing subset, is used for measuring the accuracy of the model prediction. The testing data set is also referred to as a hold-out sample in the literature, as the learning phase has not utilized it for model development. To evaluate the performance of the model, a confusion matrix is created by making predictions on the data sets and comparing them to the actual state. According to Fawcett, T. (2006), a confusion matrix is a simple intuitive way to summarize the results. A confusion matrix is shown in Table II.

In Table II, the four cells represent the counts of:

(1) True positives (TP): Observations where the actual and predicted transactions were failures,

(2) True negatives (TN): Observations where the actual and predicted transactions weren't failures,

(3) False positives (FP): Observations where the actual transactions weren't failures but predicted to fail, and

(4) False negatives (FN): Observations where the actual transactions were failures but weren't predicted to fail.

The overall accuracy of the method in classifying the data set correctly is (TP+TN)/(TP+FP+FN+TN). This accuracy can be expressed as a percentage that could be any value between 0 percent and 100 percent. Higher accuracy means that the method correctly predicted the actual state of the equipment more consistently. In the final stage of the system design phase, experimentation with the window length, the choice of features, and choice of classification technique is done to maximize accuracy on the testing data set. These selected features and the techniques would then be used in the system operation phase.

Figure 6 depicts the system operation phase of the PdMS. In the system operation phase, the input measurement data stream is continuously processed to extract the features vector

Table II.
Confusion matrix

		Actual	
		Failure	Not failure
Prediction	Failure	True positives (TP)	False positives (FP)
	Not failure	False negatives (FN)	True negatives (TN)

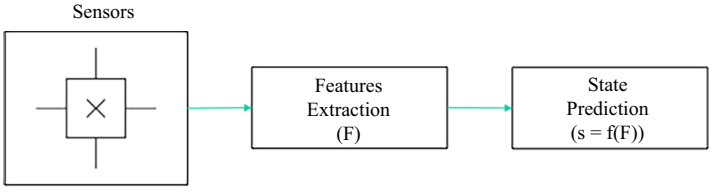


Figure 6.
System
operation phase

(F). Using the functional model developed in the system design phase, the likelihood of each state of the system is the output, which then forms the basis for the decision to intervene to perform predictive maintenance.

An illustrative case study

The agricultural company mentioned in the Introduction has the infrastructure and a strategy for predictive maintenance of its equipment and to assist their dealers in the process. Understandably, the company wants to protect proprietary data. As such, we have used a publicly available data set to illustrate our proposed methodology. According to Yin (1994), using an illustrative case study is a valid research methodology. This test case is based on hard disk drive smart sensor data in a large data center. Storage is one of several subsystems in operating a data center in addition to power, cooling, communication, and environment. We organize the presentation in this section as follows: first we describe the context and the data set, and in the second subsection, we describe the design strategy and the effectiveness of its performance in this context followed by a summary of the results.

Context and the data set

The data center in our illustrative case study is owned and operated by the Backblaze Cloud Storage & Backup Company. This company was founded in 2007, and provides low-cost data backup services for individuals and business organizations. At the end of 2018, the company was storing over 500 petabytes of data for its customers with over 100,000 hard disk drives in operation in its data centers.

Every day in the Backblaze data center, a snapshot of each operational hard drive is taken (Backblaze, 2019). This snapshot includes basic drive information along with the SMART (Self-Monitoring Analysis and Reporting Technology) statistics reported by that drive. The daily snapshot of a drive is one record in a table, that is, a row of data (see Figure 7). All such drive snapshots for a given day are collected into a file consisting of a row for each active hard drive. The first row of each file contains the column names; the remaining rows are the actual data. The columns are as follows: date, that is, the date the data were collected, the serial number, the model and the capacity of the drive, whether or not the drive failed that day, and over 100 columns of SMART sensor readings. Each raw SMART sensor reading, as well as a normalized value, is provided. As there is a record for each drive, the number of rows is over 100,000.

The data set we used were these sensor files for every day of 2018. We extracted data for every drive that failed during this time frame. All the available sensor readings for each of these drives for all the days they were in operation were combined into a large data file as shown in Figure 8. Notice that the combined data file has the same columns as the sensor logs, but is arranged in chronological order for each of the days in operation for the failed drives. So, the sensor readings for each of the days until failure occurs would be the data streams that form the inputs to a predictive maintenance system and the output is the system state as to whether the drive is operating nominally or has failed. A hard disk failure is a binary event {0,1}, similar to a DTC being fired in heavy equipment operation. There was a total of 232,662 records in this data set.

Figure 9 depicts a histogram of hard disk failures by the month of the year. The data set contained sensor data from 1/1/2018 to the date of failure for a total of 1,381 hard drives. The minimum number of failures of 74 were in the month of April. The maximum number of failures of 140 were in the month of November. The average number of failures per month were 115.

	A	B	C	D	E	F	G	H
1	date	serial_number	model	capacity_bytes	failure	smart_1_normalized	smart_1_raw	smart_2_no
2	12/31/18	Z305B2QN	ST4000DM000	4.00079E+12	0	111	35837176	
3	12/31/18	ZJV0XJQ4	ST12000NM0007	1.20001E+13	0	83	199061960	
4	12/31/18	ZJV0XJQ3	ST12000NM0007	1.20001E+13	0	84	230661000	
5	12/31/18	ZJV0XJQ0	ST12000NM0007	1.20001E+13	0	81	132985536	
6	12/31/18	PL1331LAHG1S4H	HGST HMS5C4040ALE640	4.00079E+12	0	100	0	
7	12/31/18	ZA16NQJR	ST8000NM0055	8.00156E+12	0	80	106063488	
8	12/31/18	ZJV02XWG	ST12000NM0007	1.20001E+13	0	83	187565394	
9	12/31/18	ZJV1CSVX	ST12000NM0007	1.20001E+13	0	82	150825056	
10	12/31/18	ZJV02XWA	ST12000NM0007	1.20001E+13	0	83	210258232	
11	12/31/18	ZA18CEBS	ST8000NM0055	8.00156E+12	0	83	209293144	
12	12/31/18	Z305DEMG	ST4000DM000	4.00079E+12	0	119	207989736	
13	12/31/18	ZA130TTW	ST8000DM002	8.00156E+12	0	83	189797768	
14	12/31/18	ZJV1CSVV	ST12000NM0007	1.20001E+13	0	82	154983248	
15	12/31/18	ZA18CEBF	ST8000NM0055	8.00156E+12	0	79	73423480	
16	12/31/18	ZJV02XWV	ST12000NM0007	1.20001E+13	0	76	40618976	
17	12/31/18	PL2331LAG9TEEJ	HGST HMS5C4040ALE640	4.00079E+12	0	100	0	
18	12/31/18	PL2331LAH3WYAJ	HGST HMS5C4040BLE640	4.00079E+12	0	100	0	
19	12/31/18	Z3023VGH	ST4000DM000	4.00079E+12	0	116	109721928	
20	12/31/18	PL1331LAHG53YH	HGST HMS5C4040BLE640	4.00079E+12	0	100	0	
21	12/31/18	88Q0A0LGF97G	TOSHIBA MG07ACA14TA	1.40005E+13	0	100	0	
22	12/31/18	PL2331LAHDUVVJ	HGST HMS5C4040BLE640	4.00079E+12	0	100	0	
23	12/31/18	ZA10JDYK	ST8000DM002	8.00156E+12	0	79	80001664	
24	12/31/18	ZA18CEBT	ST8000NM0055	8.00156E+12	0	81	139086800	
25	12/31/18	PL1331LAHD252H	HGST HMS5C4040BLE640	4.00079E+12	0	100	0	
26	12/31/18	PL1331LAGSPEUH	HGST HMS5C4040ALE640	4.00079E+12	0	100	0	
27	12/31/18	PL1331LAHD1HTH	HGST HMS5C4040BLE640	4.00079E+12	0	100	0	
28	12/31/18	ZCH0EBLP	ST12000NM0007	1.20001E+13	0	71	13394248	
29	12/31/18	Z306WYZZ	ST4000DM000	4.00079E+12	0	120	239189728	
30	12/31/18	Z3026Z8H	ST4000DM000	4.00079E+12	0	119	226475688	
31	12/31/18	Z4D062ZT	ST6000DX000	6.00118E+12	0	111	39166976	
32	12/31/18	ZA180Z16	ST8000NM0055	8.00156E+12	0	83	184300768	

Figure 7.
Sample of characteristics, signals, and state data for hard disk drives

	date	serial_number	model	capacity_bytes	failure	smart_1_normalized	smart_1_raw	smart_2_normalized	smart_2_raw	sm
323	2018-11-19	175PP3I7T	TOSHIBA MQ01ABF050M	500107862016	0	100	0	100	0	
324	2018-11-20	175PP3I7T	TOSHIBA MQ01ABF050M	500107862016	0	100	0	100	0	
325	2018-11-21	175PP3I7T	TOSHIBA MQ01ABF050M	500107862016	0	100	0	100	0	
326	2018-11-22	175PP3I7T	TOSHIBA MQ01ABF050M	500107862016	0	100	0	100	0	
327	2018-11-23	175PP3I7T	TOSHIBA MQ01ABF050M	500107862016	0	100	0	100	0	
328	2018-11-24	175PP3I7T	TOSHIBA MQ01ABF050M	500107862016	1	100	0	100	0	
329	2018-01-01	177QT3DUT	TOSHIBA MQ01ABF050M	500107862016	0	100	0	100	0	
330	2018-01-02	177QT3DUT	TOSHIBA MQ01ABF050M	500107862016	0	100	0	100	0	
331	2018-01-03	177QT3DUT	TOSHIBA MQ01ABF050M	500107862016	0	100	0	100	0	
332	2018-01-04	177QT3DUT	TOSHIBA MQ01ABF050M	500107862016	0	100	0	100	0	
333	2018-01-05	177QT3DUT	TOSHIBA MQ01ABF050M	500107862016	0	100	0	100	0	
334	2018-01-06	177QT3DUT	TOSHIBA MQ01ABF050M	500107862016	0	100	0	100	0	
335	2018-01-07	177QT3DUT	TOSHIBA MQ01ABF050M	500107862016	0	100	0	100	0	
336	2018-01-08	177QT3DUT	TOSHIBA MQ01ABF050M	500107862016	0	100	0	100	0	
337	2018-01-09	177QT3DUT	TOSHIBA MQ01ABF050M	500107862016	0	100	0	100	0	
338	2018-01-10	177QT3DUT	TOSHIBA MQ01ABF050M	500107862016	0	100	0	100	0	
339	2018-01-11	177QT3DUT	TOSHIBA MQ01ABF050M	500107862016	0	100	0	100	0	
340	2018-01-12	177QT3DUT	TOSHIBA MQ01ABF050M	500107862016	0	100	0	100	0	
341	2018-01-13	177QT3DUT	TOSHIBA MQ01ABF050M	500107862016	0	100	0	100	0	
342	2018-01-14	177QT3DUT	TOSHIBA MQ01ABF050M	500107862016	0	100	0	100	0	
343	2018-01-15	177QT3DUT	TOSHIBA MQ01ABF050M	500107862016	0	100	0	100	0	
344	2018-01-16	177QT3DUT	TOSHIBA MQ01ABF050M	500107862016	0	100	0	100	0	
345	2018-01-17	177QT3DUT	TOSHIBA MQ01ABF050M	500107862016	0	100	0	100	0	
346	2018-01-18	177QT3DUT	TOSHIBA MQ01ABF050M	500107862016	0	100	0	100	0	

Showing 322 to 347 of 232,663 entries

Failure

Figure 8.
Combined data file for hard disk drives

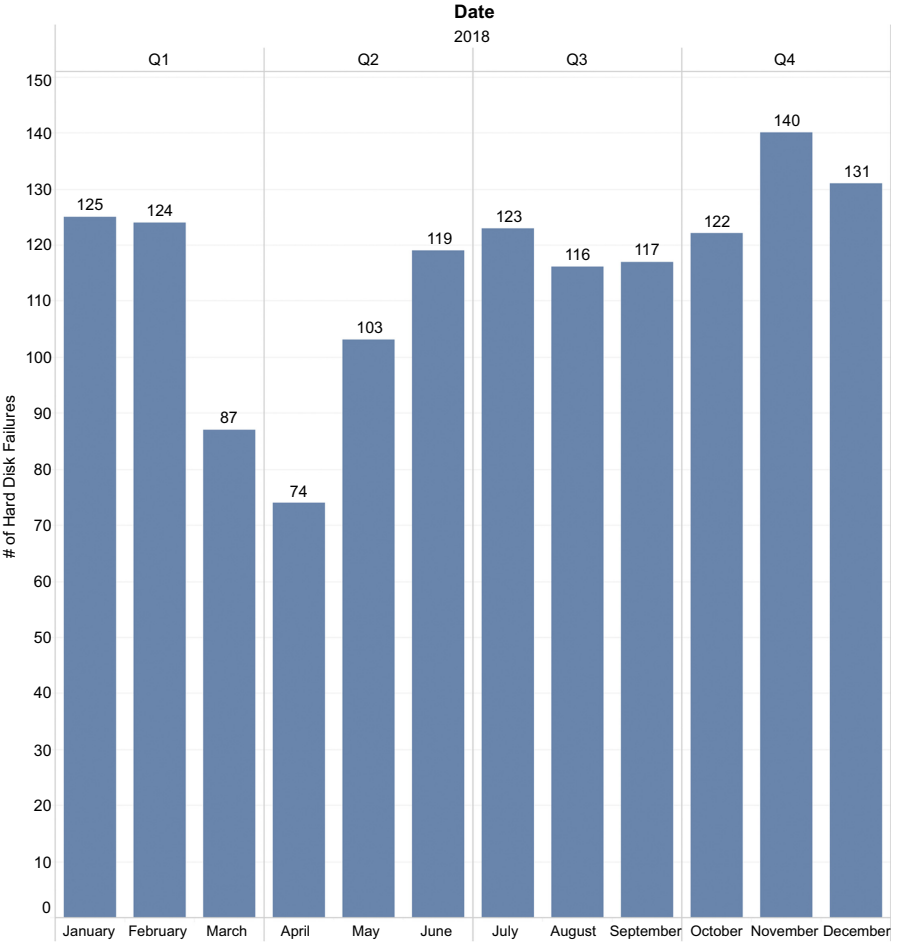


Figure 9.
Hard disk failures by month histogram

Predictive maintenance system design and effectiveness of its performance

We designed and simulated the operation of several predictive maintenance systems using this data set. We studied the performance of three classes of models in this research. These three classes include the two decision-tree-based techniques, CTree and random forest, and a third logistic regression technique. We use a logistic regression as the base case for comparative purposes. Regression techniques are the de facto standard for data analysis and the logistic regression is the standard analytical technique when the dependent variable is discrete (Hosmer *et al.*, 2013). We used R-Studio and the R programming language for the data analysis and illustrations. The dplyr package was used for data manipulation. The partykit package was used for the CTree technique, and the random forest package was used for implementing the random forest technique. The glm function was used for the logistic regression. The caret package was used for creating the confusion matrices and measuring the accuracy of the ML models. Missing values were replaced by the median values using the na.roughfix function in the random forest package. The training set contained 75 percent of the 1,381 hard disk drives chosen randomly, and the rest 25 percent that were held out were used as the testing set.

Our first investigation involves a base case of using logistic regression as the ML technique for system design, using the machine condition data from a random sampling as the signals representing the state that the system is in good condition (pass), and the machine condition data from the last sampling prior to failure as the signals representing that the system is bound to fail. The signals are just the values from these single records, so the window length for feature measurements is one day. The model evaluation results are presented in Table III. A cut-off probability of 0.5 was used to classify classifications as 1 or 0. As seen in Table III, essentially all observations are classified as good in both the training and testing data sets. The accuracy of classification is 63.23 percent on the testing data set, indicating that it does better than a random assignment.

Our second investigation is similar to the first except that it involves using CTree as the ML technique for system design. The model evaluation results are presented in Table IV and are very similar to the first investigation. As seen in Table IV, more observations that were not failures are classified as failures in this scenario. Our third investigation is similar to the first and the second except that it involves using random forest as the ML technique for system design; the model evaluation results are presented in Table V. As seen in Table V, the random forest technique performs the best in terms of the accuracy of 70.64 percent on the testing data set and indicates that the results are generalizable to unseen data.

Our second set of investigations is summarized in Table VI. These experiments were done using a window length of ten days. The average of the signals for the ten days prior to and including the failure date included the features selected as indicators of failure. The averages of random ten-day periods when the drive is in good condition was used as indicators of a healthy state. The random forest techniques perform the best among the three, with an accuracy of 73.46 percent on the testing data set. This accuracy is about 3 percent points more than the accuracy obtained when the window length was just one day. Increasing the window length from one day to ten days has improved the accuracy of the classification.

Training data set	Testing data set
Confusion Matrix and Statistics Reference Prediction 0 1 0 950 642 1 67 375 Accuracy : 0.6514	Confusion Matrix and Statistics Reference Prediction 0 1 0 317 226 1 27 118 Accuracy : 0.6323

Table III. Investigation 1: technique: logistic regression of a random record: pass. Last record: fail, window length: 1, feature: mean

Training data set	Testing data set
Confusion Matrix and Statistics Reference Prediction 0 1 0 911 646 1 106 371 Accuracy : 0.6303	Confusion Matrix and Statistics Reference Prediction 0 1 0 303 222 1 41 122 Accuracy : 0.6177

Table IV. Investigation 2: technique: CTree of a random record: pass. Last record: fail, window length: 1, feature: mean

To pick the best window length and the best technique to use, we evaluated the performance of the techniques for windows lengths varying from 1 to 60. These results are presented in Table VII and then in Figure 10. The best accuracy of about 80 percent is achieved by the random forest technique when a window length of 43 days is used.

Table VIII summarizes the features that contributed most to the decrease in impurity averaged over all trees. For classification, the node impurity is measured by the Gini index. Reduction in the Gini index indicates the improvement in the classification accuracy contributed by that variable. The smart_9_raw is the best feature for identification of failure followed by smart_242_raw and smart_197_raw.

Our final set of investigations included feature extraction by using the means, maximum values, minimum values, and standard deviation of the signals in the window length varying from 2 to 60 days. Results are shown in Table IX and Figure 11.

Training data set	Testing data set
Confusion Matrix and Statistics Reference Prediction 0 1 0 631 453 1 386 564 Accuracy : 0.5875	Confusion Matrix and Statistics Reference Prediction 0 1 0 268 126 1 76 218 Accuracy : 0.7064

Table V.
Investigation 3: technique: random forest of a random record: pass. Last record: fail, window length: 1, feature: mean

Technique	Training data set	Testing data set
Logistic regression	Confusion Matrix and Statistics Reference Prediction 0 1 0 853 574 1 73 352 Accuracy : 0.6506	Confusion Matrix and Statistics Reference Prediction 0 1 0 281 198 1 28 111 Accuracy : 0.6343
CTree	Confusion Matrix and Statistics Reference Prediction 0 1 0 810 565 1 116 361 Accuracy : 0.6323	Confusion Matrix and Statistics Reference Prediction 0 1 0 275 194 1 34 115 Accuracy : 0.6311
Random forest	Confusion Matrix and Statistics Reference Prediction 0 1 0 625 372 1 301 554 Accuracy : 0.6366	Confusion Matrix and Statistics Reference Prediction 0 1 0 236 91 1 73 218 Accuracy : 0.7346

Table VI.
Second set of experiments: technique: all features from a random set of 10 records: pass. Features from last 10 days: fail. Window length: 10, feature: Mean

Window length	Logistic regression Training set Accuracy	Logistic regression Testing set Accuracy	Ctree technique Training set Accuracy	Ctree technique Testing set Accuracy	Random forest Training set Accuracy	Random forest Testing set Accuracy	Num of Drives
1	65.14%	63.23%	63.03%	61.77%	58.75%	70.64%	1361
2	65.22%	63.39%	62.21%	61.61%	60.66%	69.94%	1335
3	65.72%	63.06%	62.64%	61.71%	60.77%	72.52%	1322
4	65.16%	62.31%	63.58%	62.77%	61.58%	71.28%	1305
5	64.47%	61.59%	65.65%	62.65%	60.61%	71.34%	1299
6	64.28%	62.54%	62.50%	61.46%	61.04%	71.05%	1279
7	64.75%	62.54%	63.32%	61.60%	61.47%	71.79%	1265
8	65.57%	62.03%	63.28%	61.87%	62.65%	72.15%	1257
9	64.55%	63.58%	64.01%	63.26%	62.09%	74.28%	1248
10	65.06%	63.43%	63.23%	63.11%	63.66%	73.46%	1235
11	64.38%	60.91%	63.67%	62.05%	63.62%	74.10%	1225
12	64.61%	62.25%	65.32%	61.93%	61.76%	74.35%	1220
13	65.23%	62.46%	66.61%	63.28%	62.79%	72.95%	1208
14	64.70%	61.97%	64.08%	62.46%	62.39%	73.11%	1193
15	64.55%	62.34%	63.42%	61.51%	62.17%	72.53%	1187
16	64.68%	63.25%	64.22%	63.58%	63.03%	72.52%	1181
17	64.03%	63.00%	63.80%	63.00%	63.17%	75.50%	1173
18	64.73%	63.04%	64.27%	63.55%	61.45%	71.91%	1168
19	64.14%	62.20%	63.96%	62.71%	63.09%	73.22%	1158
20	65.35%	62.97%	63.72%	62.46%	63.37%	74.57%	1153
21	63.71%	61.17%	64.72%	62.89%	62.12%	71.65%	1137
22	64.81%	63.33%	59.14%	56.84%	61.95%	75.26%	1122
23	64.29%	65.02%	64.53%	63.07%	62.83%	76.33%	1109
24	64.96%	63.83%	65.51%	64.01%	64.59%	73.23%	1101
25	63.63%	63.88%	67.26%	66.90%	63.56%	75.09%	1092
26	65.23%	64.26%	67.67%	65.16%	65.04%	74.55%	1078
27	50.51%	50.36%	65.61%	62.50%	65.11%	75.00%	1067
28	64.45%	63.47%	60.20%	58.67%	64.13%	76.75%	1060
29	65.27%	64.18%	68.64%	67.35%	65.01%	76.87%	1054
30	65.07%	63.53%	65.52%	62.41%	64.43%	77.44%	1049
31	64.88%	62.03%	65.01%	62.78%	65.72%	76.50%	1042
32	50.65%	50.38%	63.73%	61.93%	67.23%	77.46%	1036
33	62.35%	61.45%	69.08%	66.79%	64.64%	77.29%	1027
34	63.60%	62.69%	60.51%	59.23%	64.26%	76.73%	1021
35	64.76%	63.62%	68.97%	67.90%	65.15%	78.40%	1016
36	63.21%	62.50%	60.82%	58.20%	64.08%	75.59%	1009
37	64.06%	61.96%	69.88%	64.71%	66.47%	76.67%	1002
38	66.78%	65.69%	68.67%	65.29%	65.09%	74.71%	997
39	64.15%	63.24%	61.29%	58.30%	64.56%	76.09%	988
40	64.77%	65.34%	65.18%	63.55%	64.77%	77.49%	982
41	50.07%	50.00%	64.65%	62.35%	65.96%	77.49%	978
42	64.98%	65.12%	65.46%	63.91%	64.70%	78.43%	969
43	65.81%	65.79%	66.16%	65.38%	65.53%	79.76%	965
44	65.31%	63.06%	64.69%	61.84%	64.90%	78.78%	960
45	61.08%	62.76%	64.80%	62.34%	65.15%	76.78%	952
46	65.17%	63.81%	65.24%	63.60%	65.03%	78.87%	951
47	64.57%	64.23%	65.13%	63.60%	65.49%	77.62%	946
48	66.00%	63.92%	64.85%	62.45%	65.85%	76.79%	934
49	63.26%	64.96%	69.31%	65.60%	66.86%	76.28%	928
50	63.35%	66.24%	60.32%	58.33%	66.02%	76.50%	927
51	65.65%	63.73%	67.03%	64.16%	66.45%	76.18%	923
52	66.81%	64.29%	56.21%	54.76%	66.74%	77.06%	915
53	63.99%	64.07%	66.42%	62.12%	67.38%	77.06%	910
54	65.93%	64.94%	66.15%	62.99%	67.19%	75.54%	906
55	51.64%	51.74%	65.47%	63.70%	65.62%	77.17%	899
56	66.47%	64.25%	65.86%	62.50%	67.07%	77.19%	893
57	65.99%	66.67%	65.99%	62.28%	66.89%	78.95%	891
58	65.79%	66.52%	65.48%	64.54%	66.39%	77.09%	889
59	65.98%	65.04%	65.98%	64.38%	65.14%	78.76%	883
60	65.01%	63.56%	65.31%	62.89%	66.23%	78.89%	878

Table VII.
Results for windows lengths from 1 to 60 for means as features

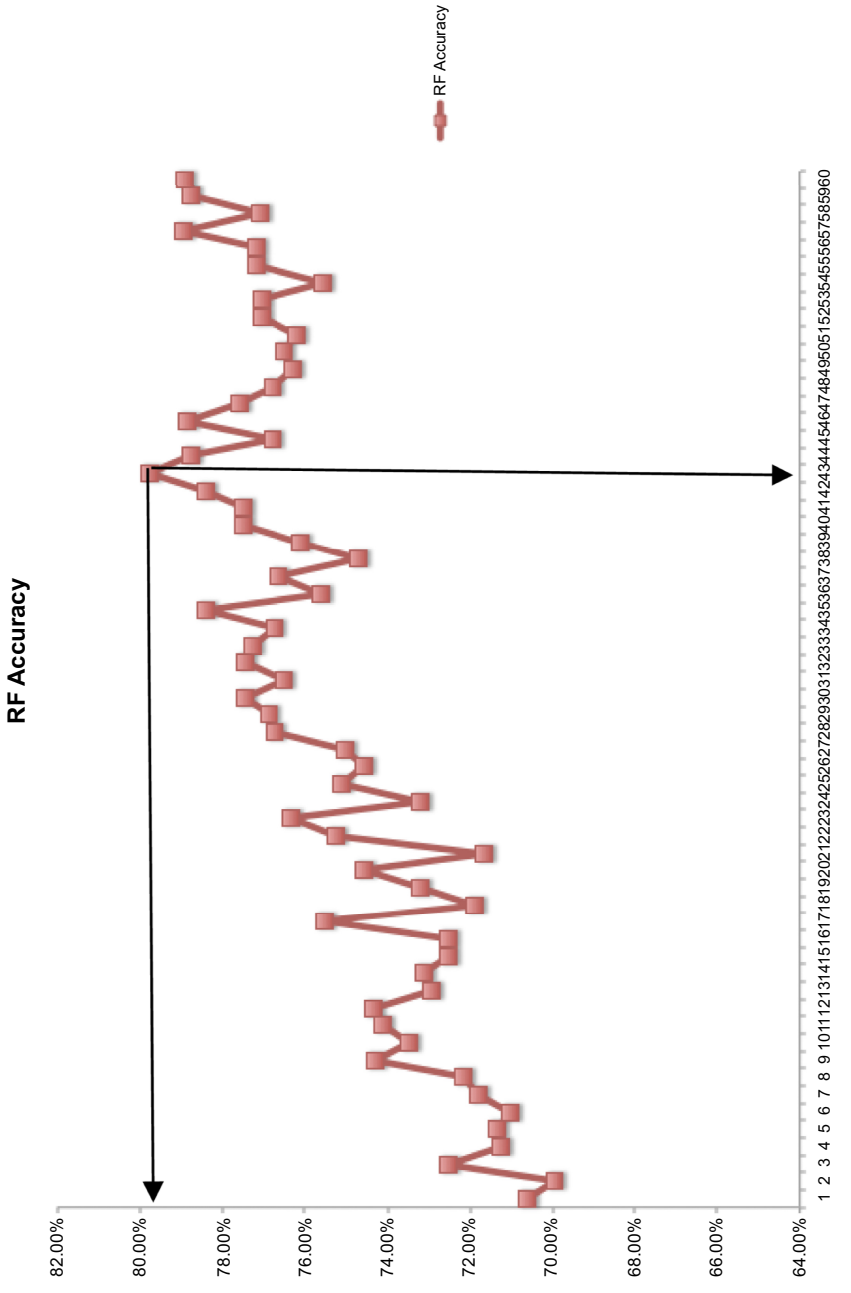


Figure 10.
Results for windows lengths from 1 to 60 for means as features

	MeanDecreaseGini
smart_9_raw	39.15
smart_242_raw	35.29
smart_197_raw	33.31
smart_9_normalized	31.77
smart_240_raw	26.80
smart_241_raw	26.77
smart_193_raw	25.46
smart_192_raw	24.33
smart_7_normalized	23.96
smart_187_normalized	23.89
smart_7_raw	23.64
smart_1_raw	22.61
smart_1_normalized	22.17
smart_194_raw	21.81
smart_187_raw	21.66
smart_5_raw	21.52
smart_194_normalized	20.87
smart_198_raw	20.00

Table VIII. Importance of the features in the random forest technique

The best accuracy occurs by using the random forest technique at the window length of 35 (See Figure 11). Even though the accuracy is similar to using just the means as features, this occurs at a shorter window length, resulting in a more manageable model. The features that are most important in terms of contribution to classification accuracy are shown in Table X. The new features do contribute to classification accuracy. The standard deviation of the smart_9_raw feature is the most important feature for identification of failure, followed by the standard deviation of smart_197_raw, and then its maximum value.

The series of experiments conducted resulted in the following findings:

(1) The accuracy of failure prediction can be improved by choosing an appropriate ML technique,

(2) The random forest technique performed the best in terms of prediction accuracy in this experiment,

(3) Feature extraction from machine condition data streams improves the prediction accuracy when compared with using just snapshot signal readings,

(4) Failure prediction accuracy can be improved by choosing appropriate features,

(5) Using statistics of machine condition data streams like the mean, standard deviation, maximum, and minimum values over a particular window length improve prediction accuracy,

(6) Choosing a proper window length involves experimentation and out-of-sample testing for generalizability of results,

(7) Decision-tree-based ML techniques provide a viable strategy for design of predictive maintenance systems.

This case provides insights for managerial decision-makers to make maintenance decisions. Similar to traditional demand forecasting and modeling, evaluating multiple techniques and reviewing their error or deviations can help a manager determine which ML technique historically performs best for specific assets and systems.

Window Length	logistic regression Training set Accuracy	logistic regression Testing set Accuracy	Ctree technique Training set Accuracy	Ctree technique Testing set Accuracy	Random forest Training set Accuracy	Random forest Testing set Accuracy	Num of drives
2	65.72%	60.00%	64.63%	63.48%	64.93%	73.33%	1335
3	50.05%	50.31%	63.67%	63.30%	64.42%	71.25%	1322
4	62.95%	59.72%	66.56%	68.83%	64.22%	77.47%	1305
5	49.95%	49.85%	68.44%	67.03%	64.29%	73.07%	1299
6	64.20%	61.16%	65.40%	67.30%	64.05%	74.37%	1279
7	50.00%	50.00%	66.53%	67.46%	67.00%	73.02%	1265
8	49.68%	50.16%	65.75%	66.08%	65.69%	75.64%	1257
9	63.86%	62.90%	62.95%	64.52%	67.22%	75.00%	1248
10	64.66%	59.77%	70.26%	65.31%	67.24%	75.57%	1235
11	66.50%	61.63%	62.82%	61.46%	67.75%	74.58%	1225
12	51.85%	51.84%	64.17%	61.87%	67.92%	73.75%	1220
13	64.71%	59.60%	65.75%	62.96%	67.12%	73.57%	1208
14	66.33%	63.18%	69.57%	68.75%	67.06%	75.00%	1193
15	65.04%	58.61%	69.70%	66.72%	66.55%	73.31%	1187
16	50.00%	50.00%	63.79%	66.22%	66.16%	74.83%	1181
17	50.00%	50.17%	68.99%	65.81%	67.86%	75.60%	1173
18	50.00%	49.83%	68.68%	66.21%	66.40%	75.52%	1168
19	50.00%	50.00%	68.27%	64.74%	67.30%	73.68%	1158
20	49.94%	50.00%	69.31%	66.78%	67.82%	75.97%	1153
21	65.21%	63.80%	64.28%	64.52%	66.08%	74.19%	1137
22	66.00%	61.27%	68.77%	66.55%	68.24%	74.73%	1122
23	67.24%	61.25%	71.24%	66.97%	69.27%	74.72%	1109
24	50.12%	50.37%	70.16%	68.70%	67.93%	74.63%	1101
25	67.62%	61.09%	65.80%	64.85%	68.04%	76.13%	1092
26	68.61%	63.64%	69.41%	67.61%	68.18%	75.57%	1078
27	50.00%	50.00%	67.02%	64.31%	68.88%	76.34%	1067
28	66.94%	67.88%	65.25%	66.15%	68.50%	77.88%	1060
29	50.00%	50.19%	68.39%	67.12%	68.64%	77.50%	1054
30	50.00%	50.00%	63.67%	64.48%	69.62%	77.99%	1049
31	51.46%	50.00%	66.56%	65.37%	70.51%	75.49%	1042
32	68.21%	63.28%	63.91%	61.91%	70.19%	75.59%	1036
33	50.00%	50.00%	64.15%	64.03%	69.19%	75.10%	1027
34	67.99%	63.55%	67.92%	68.13%	69.61%	76.29%	1021
35	66.84%	62.60%	68.15%	68.60%	69.97%	79.40%	1016
36	50.00%	49.80%	64.11%	65.38%	68.31%	77.73%	1009
37	70.24%	65.04%	65.01%	60.16%	70.44%	76.22%	1002
38	69.68%	62.24%	71.21%	68.98%	68.82%	76.94%	997
39	69.80%	63.37%	62.28%	62.35%	69.46%	75.72%	988
40	67.43%	63.02%	66.08%	62.81%	70.61%	78.31%	982
41	47.49%	50.00%	66.42%	68.05%	70.90%	75.93%	978
42	70.73%	62.61%	66.21%	65.55%	71.89%	76.89%	969
43	75.38%	66.53%	64.68%	66.74%	70.03%	78.60%	965
44	70.76%	63.83%	66.76%	67.02%	72.62%	77.87%	960
45	70.92%	62.34%	66.04%	64.26%	71.62%	76.17%	952
46	69.94%	63.46%	65.90%	65.81%	72.38%	74.79%	951
47	71.22%	64.44%	65.69%	65.30%	71.50%	76.51%	946
48	48.37%	53.07%	66.36%	66.67%	70.89%	76.97%	934
49	50.00%	50.00%	66.29%	62.89%	70.06%	78.67%	928
50	51.00%	50.44%	66.52%	64.00%	70.09%	78.00%	927
51	50.07%	49.78%	68.14%	66.14%	70.71%	78.25%	923
52	71.93%	63.06%	70.78%	66.89%	70.85%	75.90%	915
53	49.93%	49.10%	64.66%	64.25%	70.83%	74.66%	910
54	73.22%	68.72%	66.81%	61.19%	71.54%	73.29%	906
55	49.93%	50.00%	68.16%	63.66%	70.42%	76.85%	899
56	73.45%	62.33%	65.93%	65.81%	72.94%	76.28%	893
57	72.75%	65.42%	66.69%	65.89%	71.12%	77.57%	891
58	73.82%	64.55%	66.86%	65.02%	71.82%	75.59%	889
59	49.78%	50.23%	68.28%	67.37%	72.91%	76.76%	883
60	50.07%	50.00%	70.46%	66.82%	72.86%	75.59%	878

Table IX.
Results for windows lengths from 2 to 60 for means, maximums, minimums, and standard deviations as features

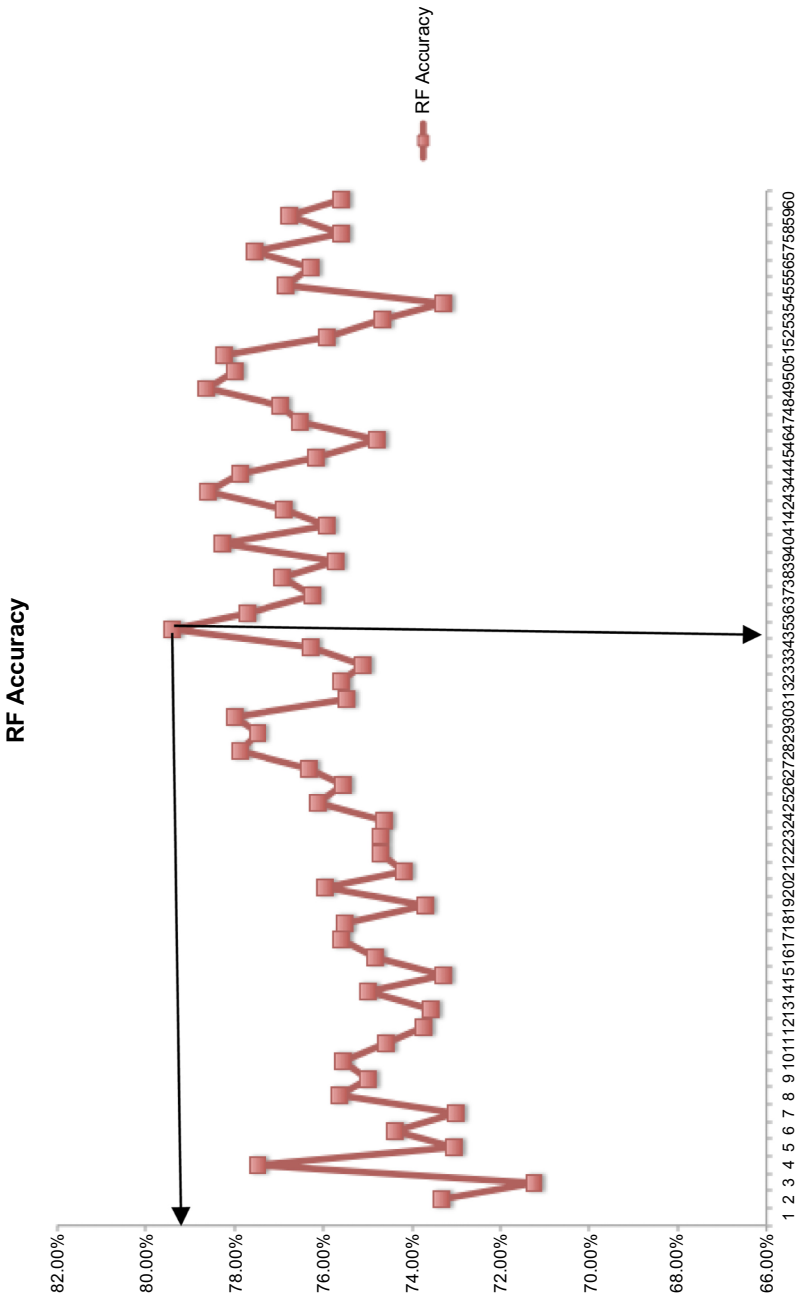


Figure 11.
Results for windows lengths from 1 to 60 for means, maximums, minimums, and standard deviations as features

	MeanDecreaseGini
smart_9_raw_sd	19.77
smart_197_raw_sd	18.18
smart_197_raw_max	14.13
smart_187_normalized_	13.47
smart_9_raw_min	12.27
smart_242_raw max	12.27
smart_5_raw_sd	12.21
smart_9_raw_max	12.12
smart_9_raw_mean	11.87
smart_242_raw_mean	11.58
smart_242_raw_min	11.53
smart_187_raw_sd	11.35
smart_197_raw_mean	10.88
smart_241_raw_sd	10.70
smart_9_normalized_m	10.04

Table X. Importance of the features in the final investigation

Conclusions and future work

While we have used Backblaze as our illustrative use case in this article, the method and application can be applied and scaled in many industries to improve maintenance decision-making. This method has been used for bearing vibration failure data and can be deployed for agricultural and construction equipment, electrical equipment, jet engines, and so forth.

Practitioner surveys regularly indicate that customer satisfaction, service profitability, first-time fix rate, service revenue, SLA/contract compliance rate, service costs, customer retention, and serviceable asset uptime/availability are the most cited KPIs for service management. In this article, we use a system theoretic perspective to review the literature in the predictive maintenance area. Using this research, we propose a framework for the design and operation of predictive maintenance systems. We further suggest the use of decision tree-based ML techniques for the design of such systems. A cloud storage provider is used as a case study to illustrate our approach, which is scalable to other industrial applications.

We proposed a strategy for the design and operation of predictive maintenance systems. Machine conditions and state are continuously monitored and reported back via IoT and cellular communication networks. Machine condition signals are processed to extract features, and the relationship between features and state is derived. These relationships are used in the operation phase for predicting the likelihood of failure and subsequently the decision to perform service. The relationships derived are in the form of decision trees and decision forests.

A data set from a cloud storage provider is used to illustrate the methodology. It is found that the random forest technique performs the best in terms of predictive accuracy. Impact of additional features is studied. This work continues to build in the area of ML and AI for predictive maintenance and service management applications. Further work could include the studying of the impact of other features, the impact of using other techniques, using remaining useful life as the dependent variable, and illustrations on other data sets. ML is inherently an evolving field, and this work adds to the body of literature being published on applications of ML to improve decision-making.

References

Backblaze (2019), "Hard drive data & stats," available at: https://www.backblaze.com/b2/hard-drive-test-data.html (accessed 29 January 2019).

Breiman, L. (2001), "Random forests", *Machine Learning*, Vol. 45 No. 5, pp. 5-32, doi: 10.1023/A:1010933404324.

Bumblauskas, D. (2015), "A Markov decision process model case for optimal maintenance of serially dependent power system components", *Journal of Quality in Maintenance Engineering*, Vol. 21 No. 3, pp. 271-293, available at: http://www.emeraldinsight.com/doi/abs/10.1108/JQME-09-2014-0050.

Bumblauskas, D., Meyer, B. and Keegan, R. (2015), "A comparative analysis of continuous improvement in Ireland and the United States", in June 2015, European Operations Management Association (EurOMA) 2015 Annual Conference, Neuchatel, Switzerland, available at: https://www.researchgate.net/publication/303518505_A_Comparative_Analysis_of_Continuous_Improvement_in_Ireland_and_the_United_States (accessed 7 February 2019).

Bumblauskas, D., Gemmill, D., Igou, A. and Anzengruber, J. (2017), "Smart maintenance decision support systems (SMDSS) based on corporate big data analytics", *Expert Systems with Applications*, Vol. 90, pp. 303-317, doi: 10.1016/j.eswa.2017.08.025.

Bumblauskas, D., Igou, A., Nold, H. and Bumblauskas, P. (2017), "Big data analytics: transforming data to action", *Business Process Management Journal*, Vol. 23 No. 3, pp. 703-720, doi: 10.1108/BPMJ-03-2016-0056.

Bumblauskas, D., Meeker, B. and Gemmill, D. (2012), "Maintenance and recurrent event data analysis of circuit breaker population data", *International Journal of Quality and Reliability Management*, Vol. 29 No. 5, pp. 560-575, available at: http://www.emeraldinsight.com/doi/abs/10.1108/02656711211230526.

Carbonell, J.G., Michalski, R.S. and Mitchell, T.M. (1983), "An overview of machine learning", in Michalski, R.S., Carbonell, J.G. and Mitchell, T.M. (Eds), *Machine Learning: An Artificial Intelligence Approach*, Tioga Publishing Company, Palo Alto.

Chan, G.K. and Asgarpoor, S. (2006), "Optimum maintenance policy with Markov processes", *Electric Power Systems Research*, Vol. 76 No. 6-7, pp. 452-456.

Carnero, M.C. (2005), "Selection of diagnostic techniques and instrumentation in a predictive maintenance program. A case study", *Decision Support Systems*, Vol. 38 No. 4, pp. 539-555.

Carnero Moya, M.C. (2004), "The control of the setting up of a predictive maintenance programme using a system of indicators", *Omega*, Vol. 32 No. 1, pp. 57-75.

Chu, C., Proth, J. and Wolff, P. (1998), "Predictive maintenance: the one-unit replacement model", *International Journal of Production Economics*, Vol. 54 No. 3, pp. 285-295, available at: https://search.proquest.com/docview/215572842?accountid=14691C.

Crowder, M. and Lawless, J. (2007), "On a scheme for predictive maintenance", *European Journal of Operational Research*, Vol. 176 No. 3, p. 1713, available at: https://search.proquest.com/docview/204142007?accountid=14691.

Eker, Ö.F., Camci, F. and Jennions, I.K. (2012), "Major challenges in prognostics: study on benchmarking prognostic datasets", *Proceedings of the 1st European Conference of the Prognostics and Health Management Society,* Dresden, Germany, 3 - 5 July 2012, PHM Society, pp. 148-155.

E Costa, C., Bana, A., Carnero, M.C. and Oliveira, M. (2012), "A multi-criteria model for auditing a predictive maintenance programme", *European Journal of Operational Research*, Vol. 217 No. 2, p. 381.

Edwards, D.J., Holt, G.D. and Harris, F.C. (1998), "Predictive maintenance techniques and their relevance to construction plant", *Journal of Quality in Maintenance Engineering*, Vol. 4 No. 1, pp. 25-37.

Fawcett, T. (2006), "An introduction to ROC analysis," *Pattern Recognition Letters*, Vol. 27 No. 8, pp. 861-874.

Flores-Colen, I., de Brito, J. and Vasco Peixoto, d.F. (2006), "Expedient in-situ test techniques for predictive maintenance of rendered façades", *Journal of Building Appraisal*, Vol. 2 No. 2, pp. 142-156, doi: 10.1057/palgrave.jba.2940047.

Flores-Colen, I., de Brito, J.M.C.L. and Vasco Peixoto, D.F. (2011), "On-site performance assessment of rendering facades for predictive maintenance", *Structural Survey*, Vol. 29 No. 2, pp. 133-146, doi: 10.1108/02630801111132812.

Grąbczewski, K. (2014), "*Meta-Learning in Decision Tree Induction*", Springer International Publishing, Cham.

Guilherme, F.T. and Miguel, A.S. (2015), "Estratégia de manutenção preditiva no departamento gráfico de uma empresa do ramo fumageiro/Predictive maintenance strategy in the graphic department of a tobacco company", *Revista Produção Online*, Vol. 15 No. 3, pp. 783-806, doi: 10.14488/1676-1901.v15i3.1623.

Herterich, M.M., Uebernickel, F. and Brenner, W. (2015), "The impact of cyber-physical systems on industrial services in manufacturing", *Procedia CIRP*, Vol. 30, pp. 323-328.

Hosmer, D.W. Jr, Lemeshow, S. and Sturdivant, R.X. (2013), "*Applied Logistic Regression*", Third Edition, John Wiley & Sons, New Jersey.

Hothorn, T., Hornik, K. and Zeileis, A. (2006), "Unbiased recursive partitioning: a conditional inference framework", *Journal of Computational and Graphical Statistics*, Vol. 15 No. 3, pp. 651-674.

Hothorn, T., Hornik, K. and Zeileis, A. (2018), "*Ctree: Conditional Inference Trees*", The R Foundation, available at: https://cran.r-project.org/web/packages/partykit/vignettes/ctree.pdf (accessed 19 January 2019).

He, Y., Gu, C., Chen, Z. and Han, X. (2017), "Integrated predictive maintenance strategy for manufacturing systems by combining quality control and mission reliability analysis", *International Journal of Production Research*, Vol. 55 No. 19, pp. 5841-5862, doi:10.1080/00207543.2017.1346843.

Huang, R., Xi, L., Lee, J. and Liu, C.R. (2005), "The framework, impact and commercial prospects of a new predictive maintenance system: intelligent maintenance system", *Production Planning and Control*, Vol. 16 No. 7, pp. 652-664.

Kumar, E.V., Chaturvedi, S.K. and Deshpandé, A.W. (2009), "Maintenance of industrial equipment", *The International Journal of Quality and Reliability Management*, Vol. 26 No. 2, pp. 196-211, doi: 10.1108/02656710910928824.

Lawrence, M., Saxena, A. and Knapp, G.M. (1995), "Statistical-based or condition-based preventive maintenance?", *Journal of Quality in Maintenance Engineering*, Vol. 1 No. 1, pp. 46-59.

Lee, J., Ardakani, H.D., Yang, S. and Bagheri, B. (2015a), "Industrial big data analytics and cyber-physical systems for future maintenance & service innovation", *Procedia CIRP*, Vol. 38, pp. 3-7.

Lee, J., Bagheri, B. and Kao, H.A. (2015b), "A cyber-physical systems architecture for industry 4.0-based manufacturing systems", *Manufacturing Letters*, Vol. 3, pp. 18-23.

Lu, S., Tu, Y.C. and Lu, H. (2007), "Predictive condition-based maintenance for continuously deteriorating systems", *Quality and Reliability Engineering International*, Vol. 23 No. 1, pp. 71-81.

Lee, D. and Pan, R. (2017), "Predictive maintenance of complex system with multi-level reliability structure", *International Journal of Production Research*, Vol. 55 No. 16, pp. 4785-4801, doi: 1080/00207543.2017.1299947.

Madu, C.N. (2000), "Competing through maintenance strategies", *International Journal of Quality and Reliability Management*, Vol. 17 No. 9, pp. 937-949.

Moubray, J. (1997), "*Reliability-centered Maintenance*", Industrial Press Inc., New York, NY.

Magro, M.C. and Pinceti, P. (2009), "A confirmation technique for predictive maintenance using the rough set theory", *Computers and Industrial Engineering*, Vol. 56 No. 4, p. 1319.

Man, C.G., Sanz-Bobi, M. and Javier, D.P. (2006), "SIMAP: intelligent system for predictive maintenance application to the health condition monitoring of a wind turbine gearbox", *Computers in Industry*, Vol. 57 No. 6, p. 552.

McKone, K.E. and Weiss, E.N. (2002), "Guidelines for implementing predictive maintenance", *Production and Operations Management*, Vol. 11 No. 2, pp. 109-124.

Moya, M.D.C.C. (2007), "Model for the selection of predictive maintenance techniques", *INFOR*, Vol. 45 No. 2, pp. 83-94.

Okogbaa, G., Huang, J. and Shell, R.L. (1992), "Database design for predictive preventive maintenance system of automated manufacturing system", *Computers and Industrial Engineering*, Vol. 23 No. 1-4, p. 7.

Parrondo, J.L., Velarde, S. and Santolaria, C. (1998), "Development of a predictive maintenance system for a centrifugal pump", *Journal of Quality in Maintenance Engineering*, Vol. 4 No. 3, pp. 198-211.

Pievatolo, A. and Ruggeri, F. (2004), "Bayesian reliability analysis of complex repairable systems", *Applied Stochastic Models in Business and Industry*, Vol. 20 No. 3, pp. 253-264.

Pinder, A. Jr, and Aberdeen Group (2015), *State of Service Management–2015*, Report Published by, Aberdeen Group, March 2015.

Pitt, T.J. (1997), "Data requirements for the prioritization of predictive building maintenance", *Facilities*, Vol. 15 No. 3, p. 97.

Quinlan, J.R. (1986), "Induction of decision trees", *Machine Learning*, Vol. 1, pp. 81-106.

Raouf, A. (2005), "Predictive maintenance of pumps using condition monitoring", *Journal of Quality in Maintenance Engineering*, Vol. 11 No. 1, p. 98.

Schlabbach, R. and Berka, T. (2001), "Reliability-centered maintenance of MV circuit-breakers", *2001 IEEE Porto Power Tech Proceedings (Cat. No. 01EX502)*, IEEE, Vol. 4, p. 5.

Settanni, E., Newnes, L., Thenent, N., Bumblauskas, D., Parry, G. and Goh, Y.M. (2015), "A case study in estimating avionics availability from field reliability data", *Quality and Reliability Engineering International*, Vol. 32 No. 4, pp. 1553-1580, available at: http://onlinelibrary.wiley.com/doi/10.1002/qre.1854/full.

Settanni, E., Newnes, L.B., Thenent, N.E., Parry, G.C., Bumblauskas, D., Sandborn, P. and Goh, Y.M. (2016), "Applying forgotten lessons in field reliability data analysis to performance-based support contracts", *Engineering Management Journal*, Vol. 28 No. 1, pp. 3-13, available at: http://www.tandfonline.com/doi/abs/10.1080/10429247.2015.1130508.

Singh Jolly, S. and Jit Singh, B. (2014), "An approach to enhance availability of repairable systems: a case study of SPMs", *International Journal of Quality and Reliability Management*, Vol. 31 No. 9, pp. 1031-1051.

Simeu-Abazi, Z. and Bouredji, Z. (2006), "Monitoring and predictive maintenance: modeling and analyse of fault latency", *Computers in Industry*, Vol. 57 No. 6, p. 504.

Wang, G., Gunasekaran, A., Ngai, E.W. and Papadopoulos, T. (2016), "Big data analytics in logistics and supply chain management: certain investigations for research and applications", *International Journal of Production Economics*, Vol. 176, pp. 98-110.

Wikipedia (2019), "Onboard diagnostics", available at: https://en.wikipedia.org/wiki/On-board_diagnostics#Standards_documents (accessed 17 August 2019).

Wong, J.K. and Li, H. (2008), "Application of the analytic hierarchy process (AHP) in multi-criteria analysis of the selection of intelligent building systems", *Building and Environment*, Vol. 43 No. 1, pp. 108-125.

Wang, J., Zhang, L., Duan, L. and Gao, R.X. (2017), "A new paradigm of cloud-based predictive maintenance for intelligent manufacturing", *Journal of Intelligent Manufacturing*, Vol. 28 No. 5, pp. 1125-1137, doi: 10.1007/s10845-015-1066-0.

Wen, D., Ershun, P., Ying, W. and Wenzhu, L. (2016), "An economic production quantity model for a deteriorating system integrated with predictive maintenance strategy", *Journal of Intelligent Manufacturing*, Vol. 27 No. 6, pp. 1323-1333, doi: 10.1007/s10845-014-0954-z.

Yang, S.K. (2003), "A condition-based failure-prediction and processing-scheme for preventative maintenance", *IEEE Transactions on Reliability*, Vol. 52 No. 3, pp. 373-383.

Yam, R.C.M., Tse, P.W., Li, L. and Tu, P. (2001), "Intelligent predictive decision support system for condition-based maintenance", *The International Journal of Advanced Manufacturing Technology*, Vol. 17 No. 5, pp. 383-391.

Yin, P.R. (1994), *Case Study Research: Design and Methods*, Sage, Thousand Oaks, CA.

You, M. (2017), "A predictive maintenance system for hybrid degradation processes", *The International Journal of Quality and Reliability Management*, Vol. 34 No. 7, pp. 1123-1135.

Further Reading

Allella, F., Chiodo, E. and Pagano, M. (2002), "Dynamic discriminant analysis for predictive maintenance of electrical components subjected to stochastic wear", *Compel*, Vol. 21 No. 1, pp. 98-115.

Complete guide to preventive and predictive maintenance (2003), *Mechanical Engineering*, Vol. 125 No. 8, p. 67.

Lee, J., Qiu, H., Yu, G. and Lin, J. (2009) and Rexnord Technical Services (2007), "Bearing Data Set", *IMS, University of Cincinnati. NASA Ames Prognostics Data Repository*, NASA Ames, Moffett Field, CA, available at: http://ti.arc.nasa.gov/project/prognostic-data-repository.

Qiu, H., Lee, J., Lin, J. (2006), "Wavelet filter-based weak signature detection method and its application on roller bearing prognostics", *Journal of Sound and Vibration*, Vol. 289, pp. 1066-1090.

Walia, A.S. (2017), *Random Forests in R*, DataSciencePlus, available at: https://datascienceplus.com/random-forests-in-r (accessed 19 January 2019).

Corresponding author
Shashidhar Kaparthi can be contacted at: shashi.kaparthi@uni.edu