Condition Based Monitoring for a Hydraulic Actuator

Stephen Adams[1], Peter A. Beling[1], Kevin Farinholt[2], Nathan Brown[2], Sherwood Polter[3], and Qing Dong[3]

[1] *University of Virginia, Charlottesville, VA, 22904, USA*
sca2c@virginia.edu
pb3a@virginia.edu

[2] *Luna Innovations Inc., Charlottesville, VA, 22903, USA*
farinholtk@lunainc.com
[3] *Naval Surface Warfare Center Philadelphia Division, Philadelphia, PA*

Abstract

In some environments where prognostics and health management would be beneficial, for example on board U.S. naval vessels, installation location and accessibility to power system must be considered. In this study, we investigate condition based maintenance and fault diagnosis for hydraulic actuators in power constrained environments. The experimental setup for collecting data is outlined, and a data set replicating multiple types of faults is collected. Several types of machine learning classifiers, including random forest and classification trees, are tested on the data set. Prediction accuracy as well as training and testing times are compared, which are used as a surrogate for power consumption in this study. We find that the random forest algorithm provides the lowest error rate of the tested classifiers but has some of the highest training and testing times. Classification trees, on the other hand, provide a better tradeoff between accuracy and computation time.

1. Introduction

Within the U.S. Navy, both surface ships and submarines can have hundreds of different hydraulic actuators performing critical and non-critical shipboard functions. Currently, most of these actuators are maintained according to a time-based schedule. A shift toward condition-based maintenance (CBM) strategies has the potential to reduce operations and maintenance costs in many applications that have relied on time-based inspection and overhaul cycles. While there is a significant need for a PHM system, there are physical restrictions that must be considered. For one, there are locations where access to electrical power is limited, or not available. Sensors that monitor these remotely installed actuators and collect data for a fault diagnosis system must be low-power. Second, there are weight and space restrictions. Therefore in addition to being low-power, the number of sensors and the amount of corresponding hardware is limited.

As another application, actuators are widely used in mechanical systems and are an essential part of these systems. For example, heating, ventilation, and cooling (HVAC) systems in environmentally friendly smart buildings rely heavily on the use of remotely controlled actuators. However, undetected failures of these actuators can cause the HVAC system to under perform causing poor air quality (Weimer et al., 2012). The poor air quality and under performance leads to a bad reputation and a negative view of smart buildings even when these systems can reduce cost and be environmentally friendly if maintained properly. Therefore, a prognostics and health management (PHM) system, and specifically fault diagnosis and classification, is imperative for mechanical systems that use a large number of actuators.

In this paper, we study CBM and fault diagnostics for actuators in a power constrained environment. There are two competing evaluation metrics: the accuracy of the diagnostics algorithms and power consumption of that algorithm. At this stage of the research project, the power consumption on hardware is not available so computation time is used as a surrogate for this metric. This work has three primary contributions to the literature:

- A description of the experimental setup used to collect the data set.
- A comparison of the accuracy and computation time for several classification algorithms on the collected data.
- The introduction of the idea of fault diagnostics and CBM in power constrained environments.

This paper is organized as follows. Section 2 contains a brief survey of the literature on diagnostics for actuators. Section 3 describes the experimental setup used for collecting the data. Section 4 gives brief description of the classification algorithms compared in this study. Section 5 provides the accuracy and computation time for the algorithms. Section 6 discusses these results and gives our conclusions.

2. BACKGROUND

Fault diagnosis for actuators has been widely studied. A method for performing fault detection and diagnosis on actuators specifically for HVAC systems was recently developed which utilizes a two-tier detection system (Weimer et al., 2012). The first tier uses a quantitative dynamic model while the second tier uses a qualitative model that makes assumption about the steady state of the system. Earlier work into developing fault diagnostics for actuators used fuzzy logic approaches (Bartyś & Kościelny, 2002) and neural networks (Patan & Parisini, 2002).

A benchmark actuator fault detection problem was formulated, and a benchmark data set for fault detection and diagnosis on actuators was created in order to compare and contrast multiple techniques (Bartyś & de las Heras, 2003; Bartyś, Patton, Syfert, de las Heras, & Quevedo, 2006). Data was generated from a mechanical setup composed of multiple actuators. This data is used for benchmarking as well as creating a MATLAB Simulink model for the actuator. The simulator can be used to generate data under multiple types of faults. Several studies have used this benchmark data set and the data generation example to test fault diagnostic algorithms including interval dynamic system models (Puig et al., 2003, 2006), neural networks (Witczak, Korbicz, Mrugalski, & Patton, 2006), neuro-fuzzy multiple-modeling (Uppal, Patton, & Witczak, 2006), and model-free approaches (Previdi & Parisini, 2006). We generate a new data set using the test bed described in Section 3 because we are studying faults on a specific type of hydraulic actuator with specific types of faults.

Low power or energy efficient computation is of interest to the field of mobile applications. Research has been conducted into energy efficient machine learning algorithms, specifically in the field of reinforcement learning and planning use patrially observable Markov decision processes (Grzes, Poupart, Yang, & Hoey, 2015). In some applications, mobile devices can be used as sensors to help classify the state or surroundings of an individual. However, a major limitation is the battery life of the mobile device. Therefore, low power sensing is of great interest to this community (Wang et al., 2009). Further, investigations in the field of power efficient mobile computing have led to studies which ask questions about sampling frequency and class specific features with the objective of finding accurate and low-power resolutions (Yan, Subbaraju, Chakraborty, Misra, & Aberer, 2012).

In this study, we wish to present the idea of PHM in low power environments and assess the tradeoffs between accuracy and power consumption. In application, questions about sampling frequency and feature selection, as in (Wang et al., 2009) and (Yan et al., 2012), will need to be made. We briefly outline future work and some initial experiments involving these issues in Section 6. At this point in our research, we wish to explore how selection of the classifier can affect the consumption of power and balance this requirement with the need for accuracy. In the next phase of the project, experiments will be conducted on prototype hardware and power will be directly measured.

3. EXPERIMENTAL SETUP

In this section, we outline the test stand built for collecting the data. The system is built around two matched Moog Flo-Tork 15,000 in.-lbf. rotary actuators that are coupled together such that one serves as the *actuator* and the other as the *load*. Power is supplied to the actuator by an industrial style hydraulic power unit (HPU) using a standard solenoid-actuated 4-way valve to permit rotation in both directions (Figure 1). The HPU's reservoir is filled with 15 gallons of ISO 68 grade Mobil DTE conventional hydraulic fluid. A 5-horsepower pump is capable of supplying the full 3000 psi pressure at 2 GPM flow rate to produce an actuation stroke time of approximately one second, given the actuator's stroke volume of 8.4 in^3. An adjustable pressure relief valve sets the maximum system pressure and can be set between 400-3000 psi. A turbine flow meter is located on the inlet to the side of the actuator that produces a counterclockwise (CCW) rotation when viewed from the actuator end of the machine. Pressure and temperature sensors are located at each inlet as well as the gear case relief port. A rotary position sensor tracks the angle of the actuator through its stroke.

Resistive torque from the matched actuator (i.e., load) is produced by the backpressure generated when forcing fluid through a flow restriction. Using a series of one-way check valves and a steady supply of oil from the reservoir, the load device accepts fluid from either a low-pressure pump or forces it through the variable orifice. The check valves passively direct the flow through an orifice for each rotation direction without user intervention. A bypass valve is included to remove the load restriction without altering the orifice settings. Analog pressure gauges are included on both sides of the load to permit accurate adjustment of the orifice settings during operation.

Friction braking is applied using a manual pump with a precision pressure gauge. The brake uses opposing hydraulic cylinders to balance the force to provide resistive torque without significant radial load on the actuator shafts.

Both internal and external leakage due to seal failures are common faults that should be considered. The test stand is provisioned to simulate both effects in a controllable manner

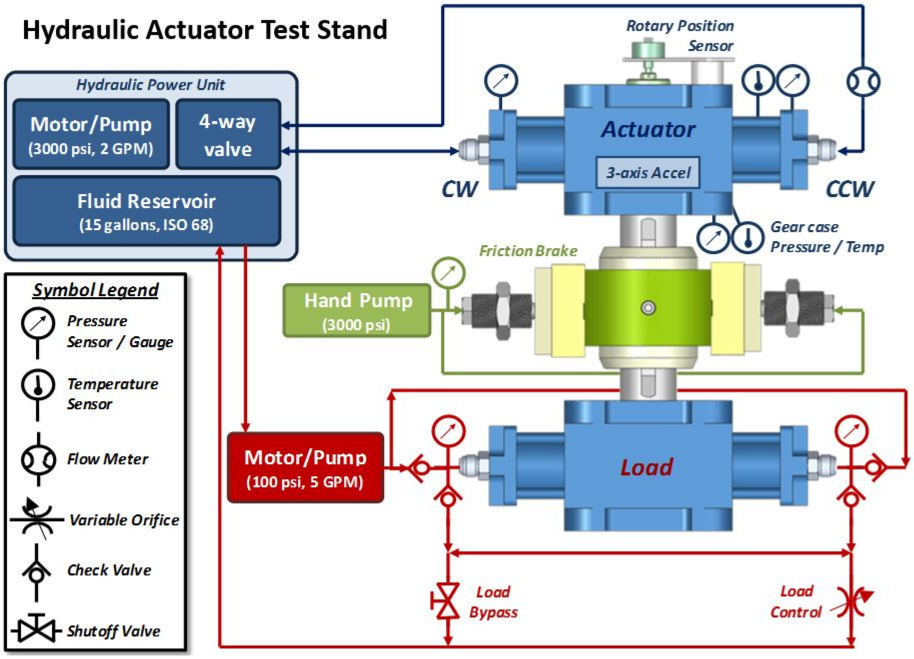

Figure 1. Actuator test stand hydraulic layout, including instrumentation locations (CW, CCW indicate direction of shaft rotation when pressure is applied)

by redirecting supply fluid according to each type of fault. The external leak path is positioned so that it redirects flow out of the CCW side of the hydraulic actuator, between the flow meter and inlet. This produces some asymmetry in the actuation response between the clockwise (CW) and counter-clockwise (CCW) direction, however the response is consistent from one measurement to the next. The actuator gear case is sealed, but has a relief valve to prevent catastrophic rupture in the event that it becomes pressurized (e.g., internal seal failure). A small manifold was developed for the gear case relief port to integrate a miniature pressure/temperature sensor and a fluid connection to simulate the leak path, while maintaining protection from the relief valve. Given that an internal seal leak can be slight and still contribute to significant case pressurization due to the limited free volume within the case, a highly restrictive leakage path was used to provide very small fluid additions per actuation cycle. The leakage path consisted of a stainless steel capillary tube with an inner diameter of 0.022 inches and a length of 72 inches, wrapped in a coiled geometry. The capillary tube was connected between the gear case and a pressure tap machined into the actuator cylinder endcap. The external leakage path was sourced from the same location, and connected back to the main sump with a needle valve to adjust external leakage rate (Figure 2).

A series of LabVIEW graphical user interfaces were developed to automate the testing process to control each of the main components of the hydraulic test stand. This provided the ability to generate more expansive datasets that became beneficial to algorithm development activities. Data from this test loop was routinely collected, and initial pre-processing of the data yielded evident trends in process measurements such as pressure, flow, and vibrations when damage cases was introduced. Data for two such test cases that simulate seal damage within the primary hydraulic actuator are shown in Figure 3. These results show clear trends in how average flow versus actuator velocity track the presence and extent of external leaks using a straightforward damage metric such as slope (Figure 3 left). Conversely, this time dependent differential pressure measurement can be used as input into more intensive classification techniques such as random forest or classification tree algorithms to generate predictors of equipment health by fusing them with other data such as angular position and average vibration (Figure 3 right). Analogous results have been seen in vibration data for load related cases such as binding when simulated in the lab. An array of test cases have been conducted, with varying operational (HPU drive frequency, variable reaction load, etc.) and environmental (hot and cold hydraulic fluid, etc.), conditions, as well as several damage scenarios that can be implemented using the existing test loop (Table 1). Variability was designed into the baseline conditions to ensure that some process conditions were incorporated into the normal operating modes of the actuator. In these cases, the severity of response corresponds to whether damage may be present in the measured response.

3.1. Data Collection

Using the experimental setup, data is generated under the 24 conditions displayed in Table 1 with a column display-

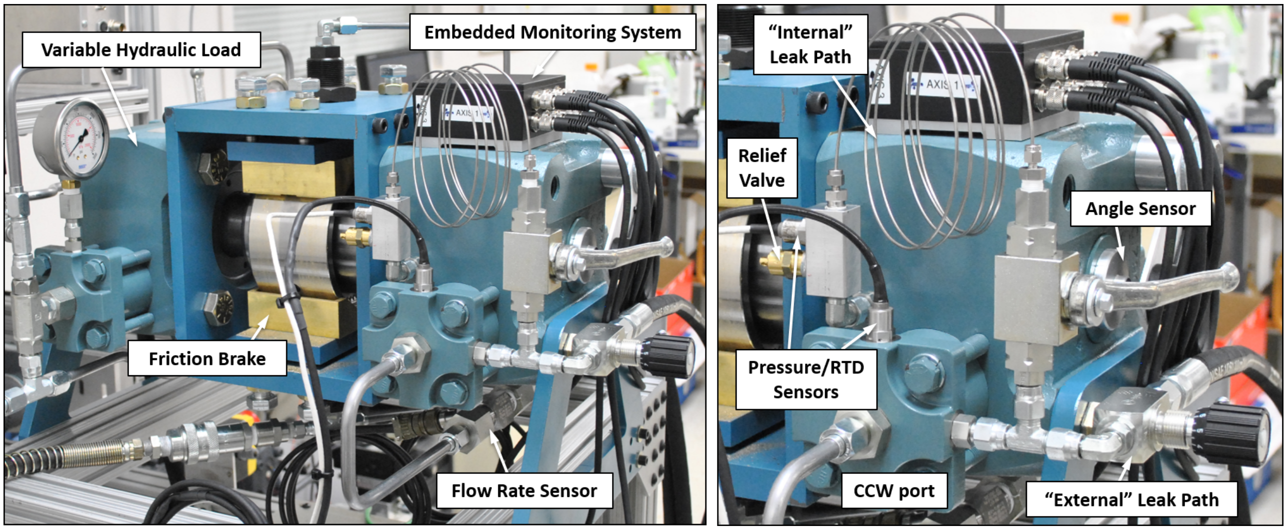

Figure 2. Detailed view of actuator/load, instrumentation locations, and leakage paths

Table 1. Damage Cases and Class Labels

Cases	# Observations	Damage Case	Class Label	6 Class Label	Fault
Baseline	2035	0	1	1	0
40 Hz	120	1	2	2	1
50 Hz	120	1	3	2	1
40 Hz 1000 PSI Backdrive/Opposing Load	20	2	4	3	1
50 Hz 1000 PSI Backdrive/Opposing Load	20	2	5	3	1
60 Hz 1000 PSI Backdrive/Opposing Load	129	2	6	3	1
60 Hz Bypass valve at 10% first turn	130	4	7	5	1
60 Hz Bypass valve at 25% first turn	69	4	8	5	1
60 Hz Bypass valve at 50% first turn	129	4	9	5	1
60 Hz Bypass valve at 100% first turn	129	4	10	5	1
60 Hz Leak Valve into case at 50%	83	5	11	6	1
60 Hz Leak Valve into case at 100%	139	5	12	6	1
60 Hz External load at 1500 PSI	129	3	13	4	1
60 Hz External load at 2500 PSI	130	3	14	4	1
60 Hz Opposing Load 1500 PSI	59	2	15	3	1
60 Hz External Load 250 PSI	62	3	16	4	1
60 Hz External Load 500 PSI	59	3	17	4	1
60 Hz External Load 1000 PSI	60	3	18	4	1
60 Hz Bypass valve at 5% first turn	60	4	19	5	1
60 Hz Bypass valve at 20% first turn	120	4	20	5	1
60 Hz Bypass valve at 150% first turn	59	4	21	5	1
60 Hz Leak Valve into case at 10%	60	5	22	6	1
60 Hz Leak Valve into case at 100% low heat	60	5	23	6	1
60 Hz Leak Valve into case at 100% high heat	60	5	24	6	1

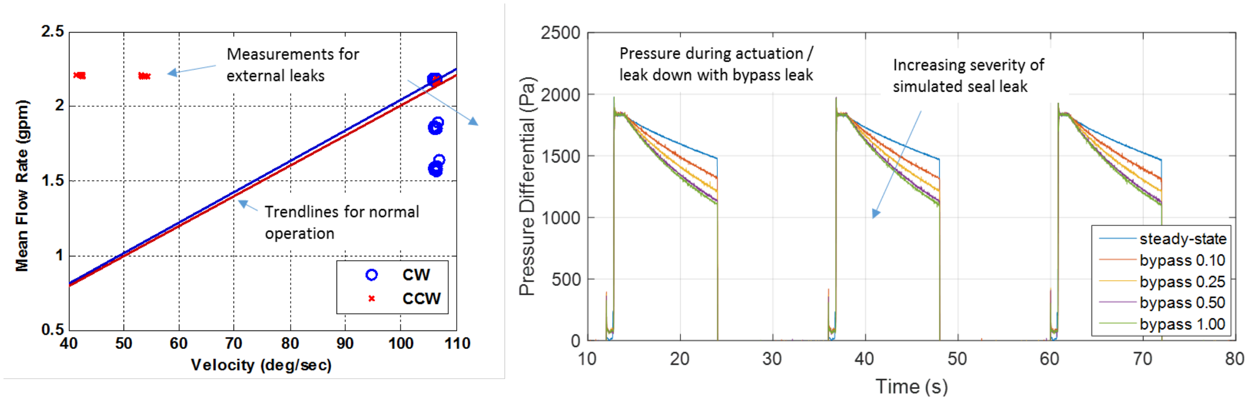

Figure 3. Data for seal leak scenarios. Metrics can be established by comparing trends between mean velocity and angular velocity of the shaft (left), as well as features extracted from time Initial results indicate clear trends in how responses change with escalating damage. CW represents clockwise motion and CCW represents counter clockwise.

ing the number of observations for each condition. There are 5 damage cases (represented by the "Damage Case" column of Table 1) and each damage case can have multiple subcategories. Multiple baseline conditions were used to generate the baseline or no-fault condition and for classification purposes are considered the same class. The "Class Label" column indicates the class number assigned to each condition. Several of the individual damage classes can be grouped together to form a smaller set of 6 classes (one baseline class and 5 damage classes) indicated in the "6 Class Label" column. The "Fault" column indicates the presence of a fault.

Five data streams are collected and used as features for the classification algorithms to identify faults: average change in angle, average differential pressure, average flow rate, average acceleration, and direction. Direction is a binary feature representing the actuator moving in either a clockwise (CW) or counter clockwise (CCW) direction. All the other collected features are continuous valued. The actuator is repetitively operated under each damage case. At the beginning stages of data collection, the acceleration data was not available. Therefore, some damage cases do not have this feature and the it is treated as a missing value during classification. In total, the data set consists of 4041 observations and 436 of these are missing values for the acceleration feature. Approximately half of the data is the baseline or no-fault case. In the multi-class problems, this leads to an issue with class imbalance. For this study, a number of operational conditions are considered normal and classified as the baseline case. It was necessary to collect data on each of these normal conditions which causes the imbalance in class distribution in the multi-class problems.

4. CLASSIFICATION ALGORITHMS

Given the collected data set, we divide the diagnostics problem into three classification problems. The first is a binary problem were the objective is to detect any damage case. In this problem, the baseline case is labeled as a negative response and all other conditions are labeled as a positive response ("Fault" column in Table 1). The next problem is the 6-class problem were the objective is to classify each observation into the baseline class or one of five damage classes (see "6 Class Label" column in Table 1). The third classification problem is the 24-class problem where each of the the individual damage cases is diagnosed.

For the binary problem, eight types of classifiers are tested: random forest (RF) (Breiman, 2001; Murphy, 2012), classification trees (Trees) (Bishop, 2006; Duda, Hart, & Stork, 2001; Murphy, 2012), *k*-nearest-neighbor (KNN) (Bishop, 2006; Duda et al., 2001; Murphy, 2012), linear discriminant analysis (LDA) (Bishop, 2006; Murphy, 2012), logistic regression (Logistic) (Bishop, 2006; Murphy, 2012), logistic regression with a probit link function (Probit) (Bishop, 2006; Murphy, 2012), quadratic discriminant analysis (QDA) (Bishop, 2006; Murphy, 2012), and support vector machines (SVM) (Bishop, 2006; Duda et al., 2001; Murphy, 2012). For the 6-class problem, SVM, Logistic, and Probit are eliminated because the standard versions of these classifiers are specific to binary classification (we acknowledge that multiclass versions of these methods which use one-versus-all comparison are available but decided not to include them in this study). For the 24-class problem, only RF, Trees, and KNN are tested due to the inability of the other classifiers being able to handle entire classes with missing values. When RF and KNN are used, the number of trees and the number of neighbors is varied to assess the affect on the classifier.

These classification algorithms were chosen for this study because they are considered standard classification algorithms that can be found outlined in most pattern recognition or machine learning textbooks. We consider these classifiers as out-of-the-box algorithms as opposed to customized algorithms specifically designed for the task of predicting faults in actuators. Further, these algorithms are readily available in MAT-

LAB. We can therefore assume that the code to train and test each algorithm is written in a similar fashion and optimized, and thus the computation time of each classifier is only affected by the mathematical complexity and not the precision of the code.

For testing, a leave-one-out cross validation (LOOCV) testing strategy was implemented. A single observation is removed from the data set and a classifier is trained on the remaining data. The class label for the the removed observation is then predicted using the trained classifier. Then the process moves on to the next observation and repeats until all observations in the data set have a predicted class label. The classification error for the classifier is assessed by counting the number of misclassified observations.

In addition to classification error, the training and testing times are recorded. The training time is the average clock time for training the classifier over the LOOCV process. Similarly, the testing time is the average clock time for testing the withheld observation over the entire LOOCV process. All tests were performed on the same machine with an Intel i7 core.

5. RESULTS

In this section, the classification error, average training time, and average testing time for each classifier on each of the three classification problems are presented. The direction feature may not be available to the classifier during a real-world application, therefore, models are trained with and without this feature for comparison. Random forests are trained with 25, 50, 75, and 100 trees. *k*-nearest neighbor classifiers are trained using 1, 3, 5, and 10 neighbors. Classification error is presented as the fraction of incorrectly classified observations. The testing and training times are in seconds.

Plots of error rate versus the training and testing times are in Figure 4 for the binary classification problem. Plots for the binary problem when using the direction features are in Figure 5. Similar figures for the 6 class (Figures 6 and 7) and the 24 class problems (Figures 8 and 9) are also presented. Note that the x axis for these plots is in the logarithmic domain.

For a better understanding of misclassification, a confusion matrix can be calculated for each classifier. In a confusion matrix, the rows represent the true class while the columns represent the predicted class. The diagonal cells in the matrix contain the number of observations correctly classified, and the off-diagonal cells contain the number of observations misclassified and the class to which they were incorrectly assigned. Confusion matrices for the 6-class problem without direction for RF25, Trees, KNN5 are given as examples in Tables 2 to 4. The labels of the rows and columns ("C#") correspond to the class label in the "6 Class Label" column of Table 1. These three classifiers where chosen to display their confusion matrices because they are the top 3 classifiers in terms of the accuracy/computation time tradeoff and could require further investigation into their classification abilities (KNN5 was chosen over the other KNN classifiers because it generally has the best error rate but not for all problems and features). For example, one of these classes could have a high misclassification cost and the confusion matrix would assist in selecting a classifier under these conditions. A high misclassification cost could represent a third objective to consider during classifier selection.

Table 2. Confusion Matrix for Random Forest with 25 Trees on 6-Class Problem.

		Predicted Class					
		C1	C2	C3	C4	C5	C6
True Class	C1	1992	0	3	5	14	21
	C2	32	208	0	0	0	0
	C3	44	0	183	1	0	0
	C4	21	1	0	417	0	1
	C5	37	1	0	0	654	4
	C6	35	1	0	0	24	342

Table 3. Confusion Matrix for classification trees on 6-Class Problem.

		Predicted Class					
		C1	C2	C3	C4	C5	C6
True Class	C1	1933	0	3	12	57	30
	C2	8	223	0	0	9	0
	C3	4	40	184	0	0	0
	C4	16	1	0	421	0	2
	C5	11	11	0	0	662	12
	C6	24	1	0	0	69	308

Table 4. Confusion Matrix for KNN classifier with 5 Neighbors on 6-Class Problem.

		Predicted Class					
		C1	C2	C3	C4	C5	C6
True Class	C1	1936	3	2	38	15	41
	C2	40	200	0	0	0	0
	C3	60	0	168	0	0	0
	C4	89	0	0	347	2	2
	C5	100	0	0	7	556	33
	C6	109	0	0	0	44	249

6. DISCUSSION AND CONCLUSION

The numerical experiments on the binary problem demonstrate that random forest outperforms the other classifiers in terms of accuracy but has significantly higher training and testing times. As the number of trees in the forest increases, these times increase but with little or no improvement in the error rate. We can conclude that there is not a significant benefit of building a forest with more than 25 trees on this data

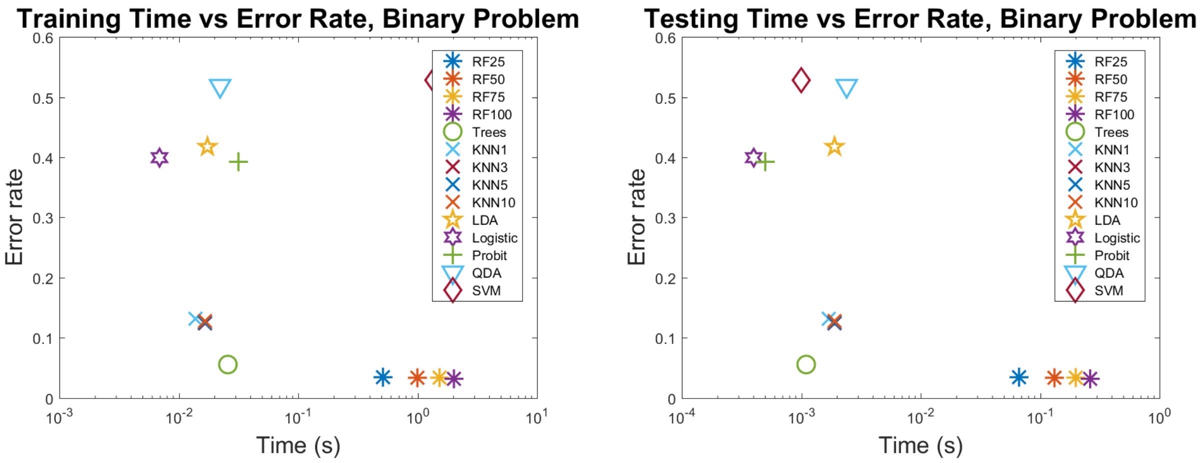

Figure 4. Left: Error rate and training time for binary problem classifiers. Right: Error rate and testing time for binary problem classifiers.

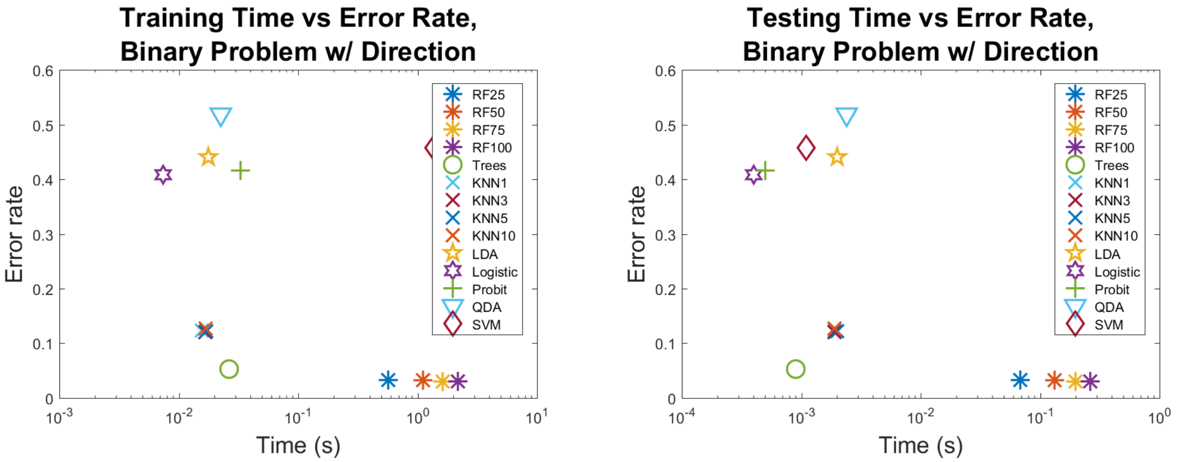

Figure 5. Left: Error rate and training time for binary problem classifiers using direction feature. Right: Error rate and testing time for binary problem classifiers using direction feature.

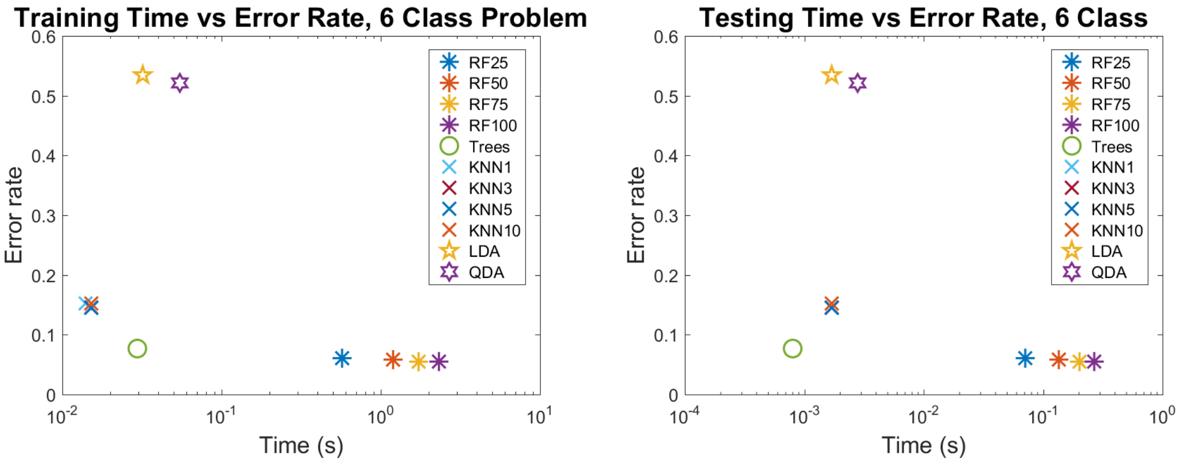

Figure 6. Left: Error rate and training time for 6 class problem classifiers. Right: Error rate and testing time for 6 class problem classifiers.

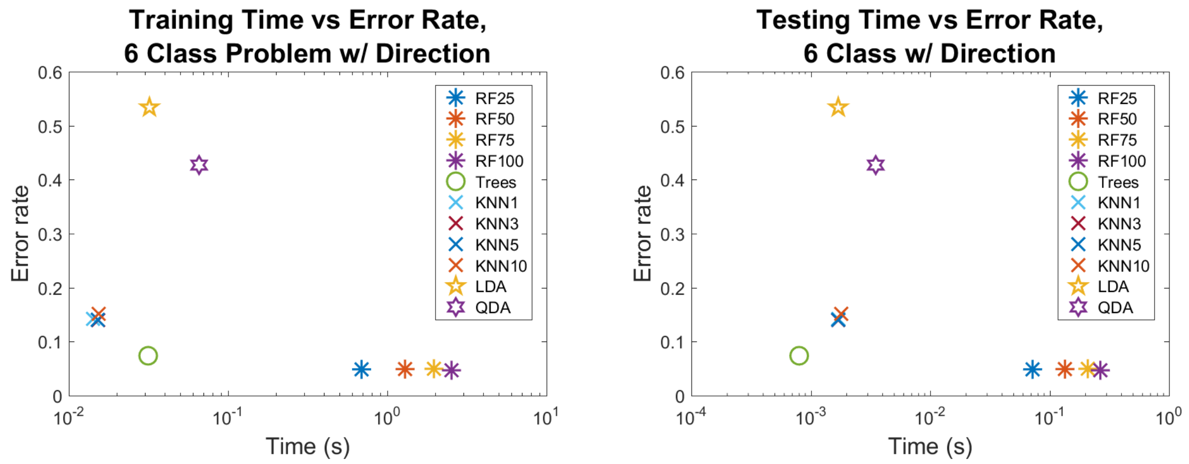

Figure 7. Left: Error rate and training time for 6 class problem classifiers using direction feature. Right: Error rate and testing time for 6 class problem classifiers using direction feature.

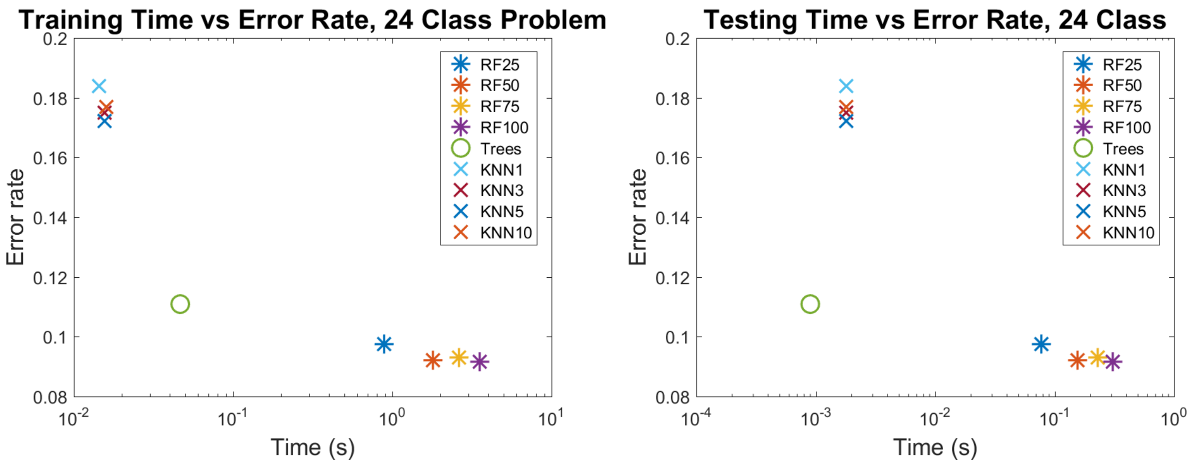

Figure 8. Left: Error rate and training time for 24 class problem classifiers. Right: Error rate and testing time for 24 class problem classifiers.

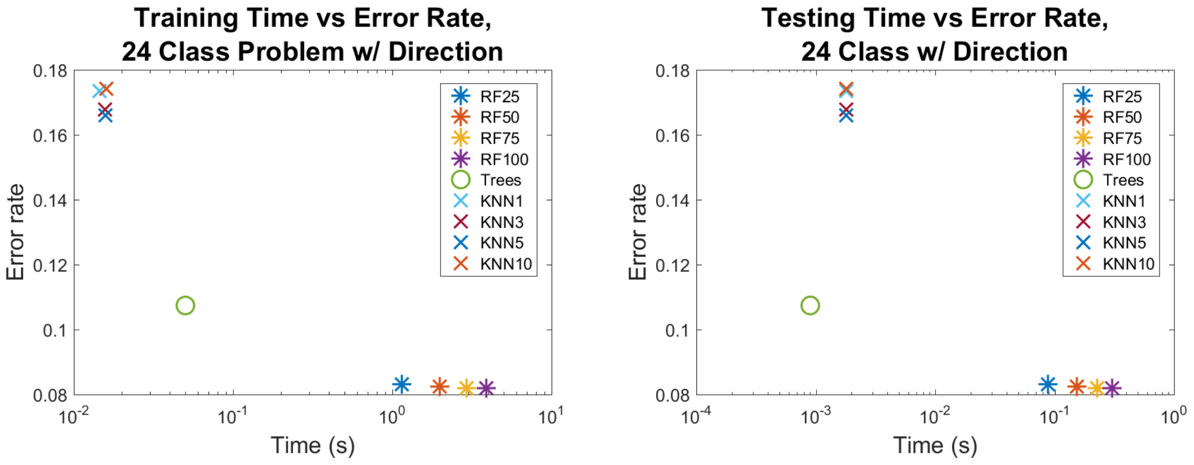

Figure 9. Left: Error rate and training time for 24 class problem classifiers using direction feature. Right: Error rate and testing time for 24 class problem classifiers using direction feature.

set. The KNN classifiers have the fastest training times, and the error rate and times do not appear to be affected by increasing the number of neighbors. The classification tree algorithm has one of the fastest testing times and better predictive ability than all classifiers except the random forest. The remaining classifiers all have error rates above 30%. When considering both evaluation metrics, the tree classifier offers a good balance of accuracy and computational load. Similar conclusions can be made from the results of the numerical experiments on the 6 class and 24 class problems. The random forest algorithm is the most accurate, but trees offer a balance of accuracy and low computational cost.

When the direction feature is added to the data, the accuracy of the classifiers on the binary problem generally improves. However, the added feature increases the training and testing times of some algorithms. We can conclude that the direction feature can improve the accuracy of some classifiers, but further study is needed to conclude if the tradeoff of accuracy for computation is justified for this feature.

This leads to one question that is not addressed in this study – feature selection. The features used as inputs into the classifiers are chosen because the data for these features is easily collected on the test bed. The computational cost of extracting the features used in this study is relatively the same for each feature, i.e. no feature is significantly more computationally expensive than another. In future work, more features will be generated from the collected data and tested. Further, feature selection will be considered as a way to reduce computational cost and improve classification performance.

Similarly, an in-depth analysis of sampling rate was excluded from this study. The sampling rate of the data will greatly affect the computational cost of the entire PHM system. We did perform initial experiments where the data was downsampled and found that this did not significantly affect the error rate. Given that features are calculated as a pre-processing step, the sampling rate would not affect the computational cost of the classifier, which is the subject of this study. An in-depth study of sampling rate will be included in future work when experiments on test hardware are performed. Similarly, data storage and the size of the classification algorithm will be investigated in future studies.

Future work will also include online training algorithms. This is the primary reason training time was used to assess the classifiers in this study. Online algorithms will be useful for updating the classifier in real-world deployment.

In conclusion, we have presented a study of different classification algorithms applied to the actuator fault detection and diagnosis problem. We have added a second evaluation metric, power consumption, and used training and testing times as a surrogate for this metric in our numerical experiments. Power consumption is an important attribute to assess as these types of actuators will most likely be installed in a power constrained environment.

ACKNOWLEDGMENT

This material is based upon work supported by the Naval Sea Systems Command under Contract No N00024-15-P-4526.

REFERENCES

Bartyś, M., & de las Heras, S. (2003). Actuator simulation of the DAMADICS benchmark actuator system. *Safe-Process 2003, Washington DC*.

Bartyś, M., & Kościelny, J. M. (2002). Application of fuzzy logic fault isolation methods for actuator diagnosis. In *Proc. 15th IFAC World Congress, Barcelona, Spain* (pp. 21–26).

Bartyś, M., Patton, R., Syfert, M., de las Heras, S., & Quevedo, J. (2006). Introduction to the damadics actuator FDI benchmark study. *Control Engineering Practice*, *14*(6), 577–596.

Bishop, C. (2006). *Pattern recognition and machine learning*. Springer-Verlag New York.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). Pattern classification. 2nd. *Edition. New York*.

Grzes, M., Poupart, P., Yang, X., & Hoey, J. (2015). Energy efficient execution of POMDP policies. *Cybernetics, IEEE Transactions on*, *45*(11), 2484–2497.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Patan, K., & Parisini, T. (2002). Stochastic approaches to dynamic neural network training. actuator fault diagnosis study. In *Proc. 15th IFAC Triennial World Congress, b* (Vol. 2, pp. 21–26).

Previdi, F., & Parisini, T. (2006). Model-free actuator fault detection using a spectral estimation approach: the case of the DAMADICS benchmark problem. *Control Engineering Practice*, *14*(6), 635–644.

Puig, V., Quevedo, J., Staucu, A., Lunze, J., Neidig, J., Planchon, P., & Supavatanakul, P. (2003). Comparison of interval models and quantised systems in fault detection with application to the DAMADICS actuator benchmark problem. In *5th IFAC Symposium on Fault Detection. Supervision and Safety for Technical Processes (SAFEPROCESS-2003), Washington, USA*.

Puig, V., Stancu, A., Escobet, T., Nejjari, F., Quevedo, J., & Patton, R. J. (2006). Passive robust fault detection using interval observers: Application to the DAMADICS benchmark problem. *Control engineering practice*, *14*(6), 621–633.

Uppal, F. J., Patton, R. J., & Witczak, M. (2006). A neuro-fuzzy multiple-model observer approach to robust fault diagnosis based on the DAMADICS benchmark problem. *Control Engineering Practice*, *14*(6), 699–717.

Wang, Y., Lin, J., Annavaram, M., Jacobson, Q. A., Hong, J., Krishnamachari, B., & Sadeh, N. (2009). A framework of energy efficient mobile sensing for automatic user state recognition. In *Proceedings of the 7th international conference on Mobile systems, applications, and services* (pp. 179–192).

Weimer, J., Ahmadi, S. A., Araujo, J., Mele, F. M., Papale, D., Shames, I., … Johansson, K. H. (2012). Active actuator fault detection and diagnostics in hvac systems. In *Proceedings of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings* (pp. 107–114).

Witczak, M., Korbicz, J., Mrugalski, M., & Patton, R. J. (2006). A GMDH neural network-based approach to robust fault diagnosis: application to the DAMADICS benchmark problem. *Control Engineering Practice*, *14*(6), 671–683.

Yan, Z., Subbaraju, V., Chakraborty, D., Misra, A., & Aberer, K. (2012). Energy-efficient continuous activity recognition on mobile phones: An activity-adaptive approach. In *Wearable Computers (ISWC), 2012 16th International Symposium on* (pp. 17–24).

BIOGRAPHIES

Dr. Stephen Adams is a Research Scientist in the Systems and Information Engineering (SIE) department at the University of Virginia (UVA). He successfully defended his dissertation in August of 2015 and will receive his Ph.D. from SIE in December of 2015. Prior to joining the Ph.D. program, he worked for UVAs Environmental Health and Safety department and completed his Masters degree in Statistics as a part-time student. He is currently part of the Adaptive Decision Systems Lab at UVA and his research is applied to several domains including activity recognition, prognostics and health management for manufacturing systems, and predictive modeling of destination given user geo-information data.

Dr. Peter A. Beling is an associate professor in the Department of Systems and Information Engineering at the University of Virginia (UVA). Dr. Beling received his Ph.D. in Operations Research from the University of California at Berkeley. Dr. Belings research interests are in the area of decision-making in complex systems, with emphasis on adaptive decision support systems and on model-based approaches to system-of-systems design and assessment. His research has found application in a variety of domains, including prognostics and health management, mission-focused cybersecurity, and financial decision-making. He is active in the UVA site of the Broadband Wireless Applications Center, which an Industry-University Cooperative Research Center sponsored by the National Science Foundation.

Dr. Kevin M. Farinholt is a senior research engineer with the Intelligent Systems Group at Luna Innovations, Inc. Dr. Farinholt received his Ph.D. in mechanical engineering from Virginia Tech. Prior to joining Luna, Dr. Farinholt served as a Research Manager for the Commonwealth Center for Advanced Manufacturing, and as a Technical Staff Member at Los Alamos National Laboratory. His research interests are in the area of condition monitoring, structural health monitoring, active material systems, and embedded sensing. He is actively involved in the intelligent structures community, serving on the ASME Technical Committee on Adaptive Structures and Material Systems.

Mr. Nathan Brown s mechanical engineering team leader with the Intelligent Systems Group at Luna Innovations, Inc. Mr. Brown received his M.S. and B.S. degrees in mechanical engineering from the University of Virginia. He oversees research in health monitoring for rotorcraft propulsion lubrication systems, rolling element bearings, as well as corrosion monitoring for process pipelines. Mr. Brown has numerous publications and three patents in the field of physical sensing technologies.