A Hydraulic Actuator Condition Monitoring Dataset for Machine Learning

Stephen Adams[1], Floyd Steele[2], Nate Brown[3], Kevin Farinholt[4], Peter A. Beling[5], and Sherwood Polter[6]

[1, 4, 5] *Virginia Tech National Security Institute, Arlington, VA, USA*

[2, 3] *Luna Labs, Charlottesville, VA, USA*

[6] *Naval Surface Warfare Center Philadelphia Division, Philadelphia, PA, USA*

Abstract

Hydraulic actuators are used across the shipping industry and are installed in various environments performing a wide array of operations. Their widespread use necessitates a condition-based maintenance program, ideally utilizing machine learning models to provide real-time estimates of the health condition, to maintain an acceptable level of fleet readiness. This paper presents a hydraulic actuator dataset collected from six industrial actuators under various fault conditions and loads. The dataset is publicly available for use in other studies. This paper describes the actuator test stand and provides a series of baseline machine learning experiments. The numerical experiments demonstrate that a machine learning model can identify classes across a range of machine learning problems. However, these numerical experiments use data from each actuator during training. The final numerical experiment withholds data from individual actuators and fault types during training. The model is unable to correctly classify the withheld fault type during this experiment. This presents an opportunity for the data set to be used in further research on generalizing models.

1. Introduction

Hydraulic actuators are used in many industries including shipping and transportation where they can be installed in harsh environments on the vessels. Generally in these domains, hydraulic actuators are serviced through a time-based-maintenance (TBM) program. The TBM strategy can also be affected by the operational aspects, i.e., actuators may only be available for service when a vessel is in port. This can cause a sub-optimal maintenance routine where either actuators are missing scheduled service because the vessel is deployed or actuators are being serviced at a higher rate than necessary because they are in port causing waste in both time and resources. Therefore, it would be beneficial to move towards a condition-based-maintenance (CBM) program that utilizes prognostic models to estimate remaining useful life (RUL). The RUL estimates can be used in conjunction with planned operational schedules to optimize maintenance planning.

Over the past decade, the field of prognostics and health management (PHM) has moved from physics-based models to data-driven models leveraging artificial intelligence and machine learning (Tsui, Chen, Zhou, Hai, & Wang, 2015; Surucu, Gadsden, & Yawney, 2023; Wen, Rahman, Xu, & Tseng, 2022). Supervised learning is one method to train deep neural networks for classification or regression operations and require large sets of accurately labeled data. Deep neural networks are the basis for most state-of-the-art machine learning and beginning to play a prominent role in PHM systems (Fink et al., 2020; Rezaeianjouybari & Shang, 2020). While this fact has lead to a number of benchmark data sets being publicly released (Helwig, Pignanelli, & Schütze, 2015; Lessmeier, Kimotho, Zimmer, & Sextro, 2016; Saxena & Goebel, 2008; Saxena, Goebel, Simon, & Eklund, 2008), there are limitations to any static data set. In order to move a predictive model from a laboratory setting to an operational setting, the model must be able to perform in a wide range of states including new damage cases and generalizing to various systems, installations, and operating environments. For example, the same type of hydraulic actuator could be used for different functions across a vessel and models may not be able to generalize across this functionality. Furthermore, models must be robust to changes in the system over time stemming from both natural causes (e.g., natural wear to a system as it ages that can not be alleviated through maintenance) and interaction with the system (e.g., performing maintenance activities may change the physical characteristics of the system that could be detected by machine learning models).

There is a need to study the ability for machine learning PHM

Stephen Adams et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.
https://doi.org/10.36001/IJPHM.2026.v17i1.4337

models to generalize across systems and operating environments. To this end, we have collected a data set from a test stand containing six industrial hydraulic actuators under operating states and damage conditions. This study contains benchmarks for performance under various training conditions to test generalization capability. These numerical experiments demonstrate that standard training procedures do not produce models that generalize to different actuators. The objective of this data set is to provide researchers with a means to study and develop models and algorithms that can adapt to various operating states as well as damage conditions. The data set and the code for the benchmark experiments are publicly available on the Virginia Tech National Security Institute github[1]. There are numerous condition monitoring data sets that contain multiple damage conditions and operating states, however most of these are focused on rolling element bearings for rotating equipment (Neupane & Seok, 2020).

This paper is organized as follows. Section 2 provides a brief review of the literature on PHM approaches for hydraulic actuators. Section 3 describes the test stand and the collected data set. Section 4 outlines the general machine learning problem for condition monitoring. The benchmark numerical experiments are outlined in Section 5, and we provide our conclusions in Section 6.

2. BACKGROUND

There is an abundance of work investigating the use of artificial intelligence and machine learning for PHM on hydraulic actuators. In the first of a series of studies, a hydraulic actuator test stand was constructed to produce data over common fault conditions (Adams et al., 2016) including valve leaks and external loads. The objective in this initial study was to characterize the trade-off between model complexity (computation time is used as a surrogate for model complexity) and performance. Model complexity is important when attempting to deploy the models to the edge, i.e., local compute on the actuator. Successfully deploying a model at the edge requires engineering decisions around several factors including power consumption and memory usage. Some of the practical aspects of this problem were studied in (Farinholt et al., 2018).

Feature selection is a machine learning technique that estimates the relevance of features to a learning problem (Adams & Beling, 2019). Feature extraction or transformation techniques project the feature set into a new lower-dimension feature space. Both strategies could be used to reduce the feature set and input space of a model and help reduce the complexity. Feature selection and feature extraction were evaluated on the hydraulic actuator problem (Adams et al., 2017). To further develop this concept and improve system design, a method for feature selection that incorporates the cost of the features was developed and evaluated on the hydraulic actuator data and other problems (Meekins et al., 2018; Meekins, Adams, Farinholt, Polter, & Beling, 2020).

Hierarchical classification organizes classes into a hierarchy and is especially suited for CBM tasks. In a general fault classification task where each fault can have varying degrees of severity, the first layer of the hierarchy is damage detection (a binary problem to determine if a fault is present in the system). The next layer is damage classification, and the third layer is severity estimation. This hierarchical formulation was applied to the hydraulic actuator problem (Adams et al., 2019). The hierarchical formulation was demonstrated to have several advantages over the standard flat classification problem. First, hierarchical performance metrics can provide a better characterization of the usefulness of the system in an operational setting where it is better to be closer in the hierarchy. For example, it is better to predict the correct fault type but be incorrect in the fault severity than predict the wrong fault type and severity. A standard flat classification would treat each prediction as equally incorrect. Second, different classifiers can be utilized at each node in the hierarchy. This can reduce the total complexity and power consumption of the system. For example, a simple model can be used at the first level for the fault detection problem. When a fault is detected, more complex models can be utilized. Furthermore, a different feature set or a different set of sensors could be used to estimate the fault type. Third, non-mandatory leaf node prediction can be implemented in order to stop the classifier to moving to lower parts of the hierarchy if it is uncertain about a prediction, e.g., only predicting the fault type if uncertain about the fault severity. In a following study, it was demonstrated that this hierarchical formulation performs better than standard classifiers when faults classes are excluded from the training set (Adams, Cody, Beling, Polter, & Farinholt, 2020).

Non-stationarity is another concern when designing data-driven PHM models. Changes in the distribution of collected features can cause a degradation in model performance. Transfer learning (Pan & Yang, 2009) is one technique for addressing non-stationarity in AI-enabled systems. Sample transfer has been applied to the hydraulic actuator problem when a degradation in model performance was observed after a maintenance procedure (Cody, Adams, Beling, Polter, et al., 2019). A systems-theoretic approach to transfer learning was developed and the concept was applied to the hydraulic actuator problem (Cody, Adams, & Beling, 2019). Transfer distance is a concept for measuring the difference between learning problems and can inform a practitioner on the ability for transfer learning to be successful. The hydraulic actuator problem was used as one example in an empirical study of transfer distance for system design (Cody, Adams, & Beling, 2022). Hydraulic actuators were also used as the motivating example for a simulation study demonstrating the value of la-

[1] $https://github.com/vtnsi/actuator\_dataset$

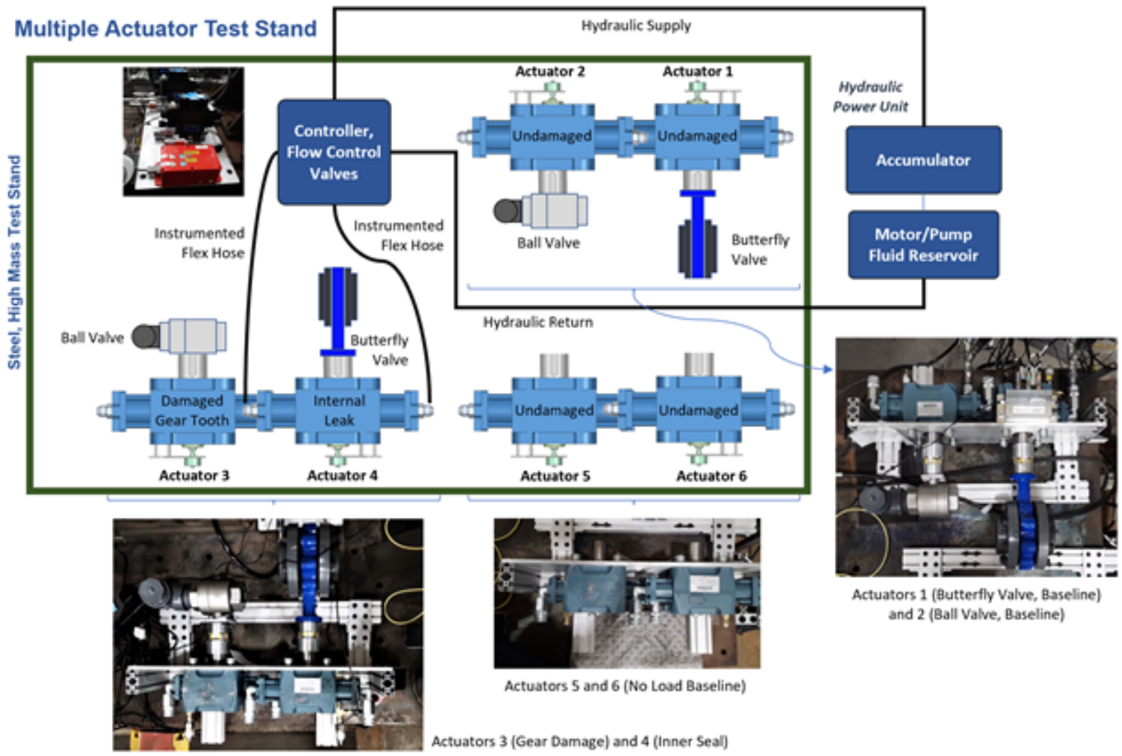


Figure 1. Equipment layout and pictures of hydraulic rotary actuators tested using Luna's v3.0 eCBM sensor node.

bels when implementing a PHM system in a production line (Cody, Adams, Beling, & Freeman, 2022).

3. DATA GENERATION

This section describes the actuator data set and the test stand used to generate the data set.

3.1. Actuator Teststand

The test stand used in this study was built using a collection of six 15,000 in lbf industrial rotary actuators manufactured by Moog Flo-Tork. The actuators were mounted to extruded aluminum structural framing fastened to a heavy steel test platform. Each actuator was actuated independently using manual and automated modes of operation through a control system built and programmed by KCF Technologies. These actuators produce 90 degrees of rotation and are energized by a small hydraulic power unit that includes a hydraulic pump, accumulator, and reservoir. Three separate unit assemblies / loading conditions are available for testing, including 1) a 2.5" ball valve, 2) a 4" butterfly valve, and 3) an unloaded state (Figure 1). Hydraulic power was delivered by an electrical pump to precharge a hydraulic accumulator so that the pump could be turned off during the actuation cycles.

Luna Labs' eCBM node was used to monitor inlet / outlet pressure conditions, temperature, position, and acceleration. The eCBM node was magnetically mounted to the top of the hydraulic actuator's steel casing, above the rotating shaft. Each actuator was run independently using a pair of instrumented flex hoses with pressure and temperature transducers, as well as piezoelectric elements used to trigger data acquisition. The hose assemblies were used for each actuator throughout the experimental study and provided consistent sensor measurements from one actuator to the next. The same rotary potentiometer was used for each of the test actuators, with shaft mounted adapter disks installed directly to the pinion face using a Loctite 454 superglue gel adhesive.

Luna Labs' eCBM electronics and associated sensors were installed and the hoses primed with the hydraulic fluid used in operation of the test loop. Once lines were filled and bled to remove entrained air, preliminary measurements were made to ensure proper operation of the hardware. Once this was confirmed, data was collected throughout Day 1 for Actuators 1, 3, 4. This provided process and dynamic response data for one of the healthy baseline assemblies (Actuator 1), one with internal gear damage (Actuator 3) and one with an internal seal defect (Actuator 4). Testing on Day 2 involved collecting a second set of data from Actuator 1, as well as measurements from the three remaining healthy actuators (Actuators 2, 5, 6). In each case, hydraulic power was provided in a closed configuration, using the accumulator to energize clockwise (CW) and counterclockwise (CCW) rotations. The accumu-

Table 1. Actuator Fault Condition and Load Summary

Equipment	Fault Condition	Load	Cases
Actuator 1	Baseline	4" Butterfly Valve	104
Actuator 2	Baseline	2.5" Ball Valve	105
Actuator 3	Gear Damage	2.5" Ball Valve	101
Actuator 4	Seal Damage	4" Butterfly Valve	105
Actuator 5	Baseline	No Load Condition	106
Actuator 6	Baseline	No Load Condition	77

Table 2. Sensor Summary

Sensor	Column Name	Description
Acceleration 1	Accel_1	ADXL1001 monitoring X-axis acceleration
Acceleration 2	Accel_2	ADXL1001 monitoring Y-axis acceleration
Acceleration 3	Accel_3	ADXL1001 monitoring Z-axis acceleration
Angle	Angle	Vishay Rotary potentiometer used to measure angular position
Temperature 1	Temp_1	Minco RTD temperature probe (Port 1)
Temperature 2	Temp_2	Minco RTD temperature probe (Port 2)
Pressure 1	PG_1	UNIK 5000 pressure transducer (Port 1
Pressure 2	PG_2	UNIK 5000 pressure transducer (Port 2)
Limit Switch	Lim	Limit switch

lator was charged to an operating pressure of 1,500 psi and used to actuate 10-12 cycles before testing was paused and the accumulator was recharged by the hydraulic pump. The supply pressure dropped to approximately 1,000 psi over these 10-12 cycles, providing a range of supply pressure levels as has been done in previous testing by Luna Labs and Moog (Adams et al., 2016).

3.2. Data Set Description

The data set is composed of examples collected from hydraulic actuators in a baseline condition and two faulty conditions - Gear Damage and Seal Damage. The examples are also collected under varying load conditions. The actuator number corresponding to the health condition and load are displayed in Table 1. This table also contains the number of cases collected from each actuator.

Each case consists of data collected during a clockwise or counter-clockwise actuation cycle. Each actuation cycle is approximately one second but data collection lasts for three seconds to ensure capturing information that may be present after the actuation cycle. The sampling rate is 1 KHz, and each sample is collected for three seconds. The sensors and their descriptions are displayed in Table 2. Figure 2 displays an example of the collected data under a baseline condition. Each example is stored in a tab separated text file.

4. MACHINE LEARNING FOR CONDITION MONITORING

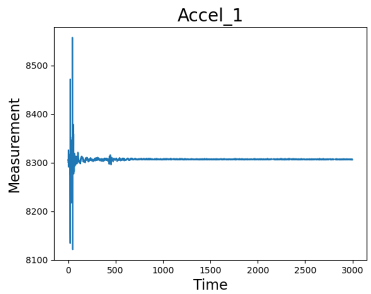
Condition monitoring can be modeled as a binary or multi-class classification problem where $\mathcal{Y} = \{y_1, ..., y_C\}$ represents the set of C discrete classes corresponding the set of health conditions. The objective is to learn a predictive function $f(x) = \hat{y}$ that maps inputs $x \in \mathcal{X}$ to a prediction of the class label

$$f(x) : \mathcal{X} \to \mathcal{Y}. \tag{1}$$

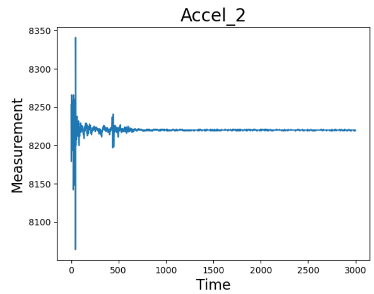
For condition monitoring, the input is generally raw data collected from the system using sensors or features extracted from the collected signals. Furthermore, collected data and features can be represented by real values or categorical variables.

Under the supervised machine learning paradigm, a training set of N observations $\{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$ is used to learn model parameters. Unsupervised learning is a subset of machine learning algorithms that are not provided labels during training so the training set only consists of the inputs $\{x_1, x_2, ..., x_N\}$. Unsupervised learning can be applied to condition monitoring and other PHM problems when the collected labels are difficult to collect (Cody, Adams, & Beling, 2017). When data can only be collected under normal conditions or when the number of collected fault conditions is far fewer than the number of collected normal conditions, condition monitoring can be modeled as an anomaly detection problem. Anomaly detection is a subfield of machine learning where $\mathcal{Y} = \{0, 1\}$ representing the normal condition and the anomaly state, respectively. However, only observations from the normal condition are used for training the model $\{(x_1, 0), (x_2, 0), ..., (x_N, 0)\}$. One-class support vector machines have been used for structural health monitoring (Adams et al., 2018).

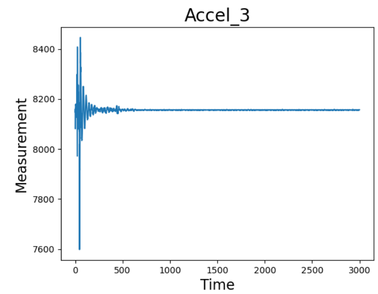
The collected data and corresponding labels are critical for



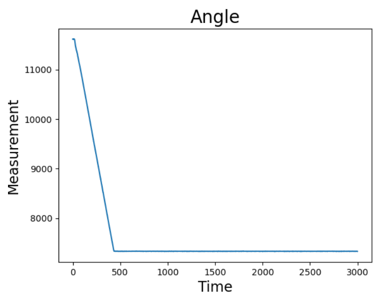
(a) Acceleration 1



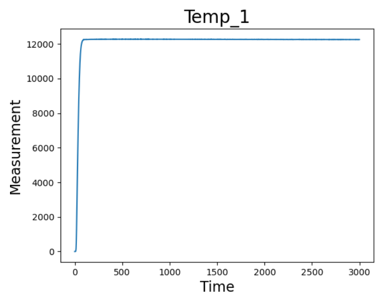
(b) Acceleration 2



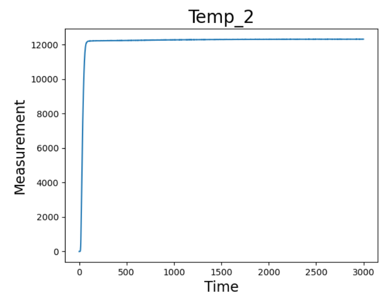
(c) Acceleration 3



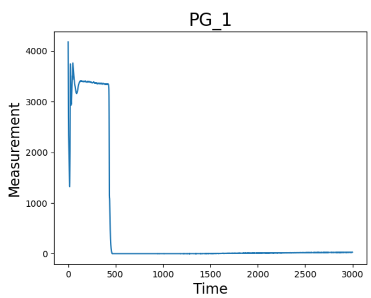
(d) Angle



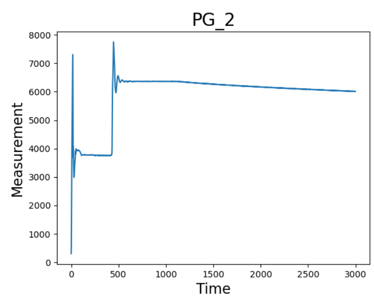
(e) Temperature 1



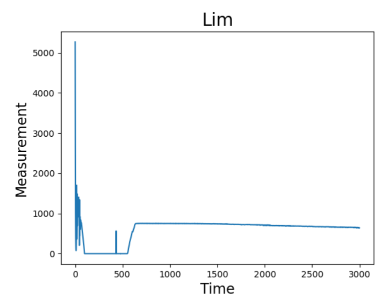
(f) Temperature 2



(g) Pressure 1



(h) Pressure 2



(i) Lim

Figure 2. Examples of sensor data from a baseline condition.

implementing machine learning in a PHM system. However, a major roadblock to widespread deployment of machine learning PHM systems is non-stationarity in the systems being monitored. Shifts in the distribution of data collected from machines can cause a degradation of the performance in the PHM system (Cody, Adams, Beling, Polter, et al., 2019). These shifts in data can be due to natural causes, such as the natural wear of the machine over time, or due to intentional intervention with the system, such as the decision to perform maintenance or operate under a new setting. Furthermore, a classification model is designed assuming a fixed set of classes. However, collecting data under all possible fault conditions is impractical, time consuming, and costly. Machine learning models for PHM must also be able to adapt to or be robust to fault conditions that are not present in the training set (Adams et al., 2020).

Transfer learning (Pan & Yang, 2009) is an area of machine learning that uses information from one problem to improve the performance of a model on a different but similar problem. Formally, let $\mathcal{D} = \{\mathcal{X}, \mathbb{P}(X)\}$ be the domain of a learning problem where $\mathbb{P}(X)$ is the marginal distribution of the feature space $\mathcal{X}$, and let $\mathcal{T} = \{\mathcal{Y}, \mathbb{P}(Y|X)\}$ be the task of a learning problem where $\mathbb{P}(Y|X)$ is the conditional probability distribution of the class $\mathcal{Y}$ given $\mathcal{X}$. In transfer learning, the objective is to use information from the source problem $\mathcal{L}_S = \{\mathcal{D}_S, \mathcal{T}_S\}$ to improve the performance of a machine learning model in the target problem $\mathcal{L}_T = \{\mathcal{D}_T, \mathcal{T}_T\}$. The source and target can be different under in multiple ways.

- The feature space could be different $\mathcal{X}_S \neq \mathcal{X}_T$. An example for a condition monitoring problem is different sensors on different machines.
- The class label space could be different $\mathcal{Y}_S \neq \mathcal{Y}_T$. An example for a condition monitoring problem is different fault conditions on different systems.
- The marginal and conditional probabilities could be dif-

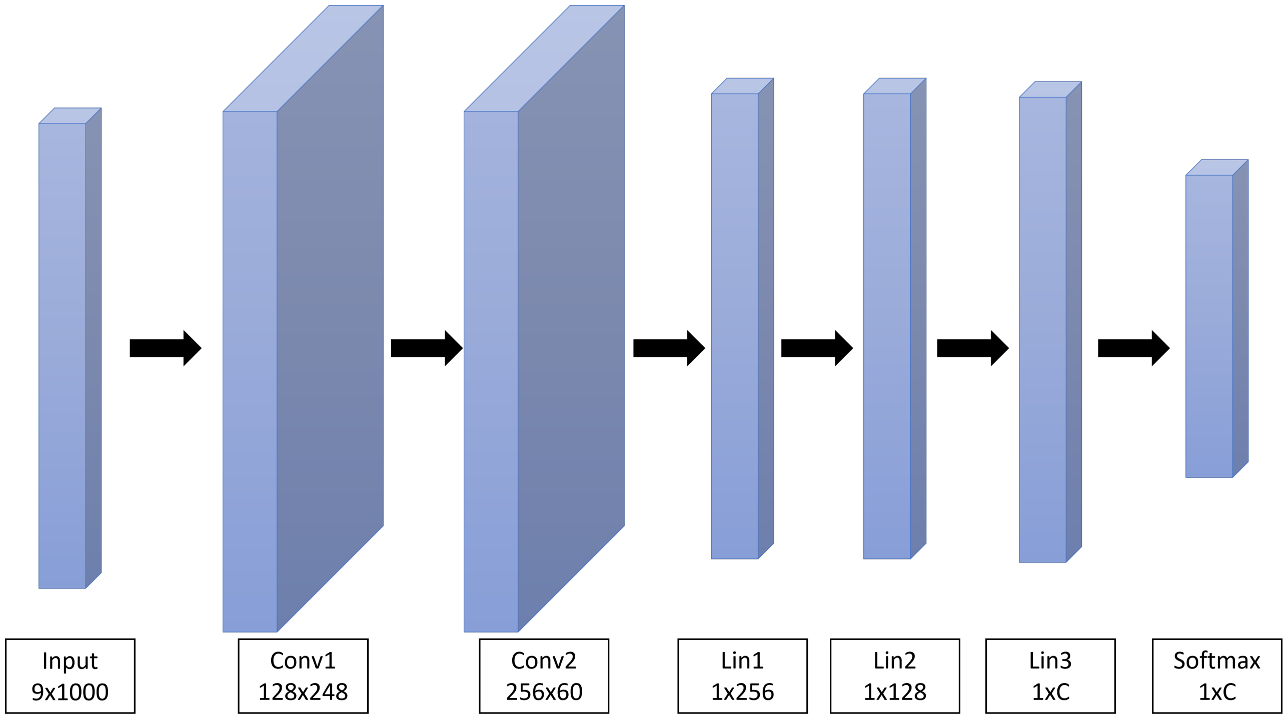


Figure 3. Base model architecture used for each classification problem. The output of each layer is listed under each layer block. The convolutional layer blocks include the ReLU activation function and max pooling. The output of Conv2 is flattened before Lin1. The linear layer blocks include the ReLU activation function (excep the final layer that does not have an activation function). C indicates the number of classes which changes based on the classification problem.

Table 3. Model Training Hyperparameters

Parameter	Value
Epochs	100
Batch Size	8
Learning Rate	0.00005
Training Set Proportion	60%
Validation Set Proportion	20%
Test Set Proportion	20%

ferent $\mathbb{P}_S(X) \neq \mathbb{P}_T(X)$ or $\mathbb{P}_S(Y|X) \neq \mathbb{P}_T(Y|X)$, respectively. An example for a condition monitoring problem is running two systems at different speeds causing a difference in the distributions of the collected data.

5. MACHINE LEARNING EXPERIMENTS

This section outlines a basic set of machine learning experiments conducted on the data set described in Section 3.2. It provides a framework for numerical experiments and a benchmark for classification accuracy across multiple classification problems.

In this section, each actuation cycle is considered as a single observation, and the objective is to classify the observation. The input data is treated as a multidimensional time-series where x has dimensions $[N, T, L]$ where $n = 1, ..., N$ indexes the sample, $t = 1, ..., T$ indexes time, and $l = 1, ..., L$ indexes the dimension or feature. The label space $\mathcal{Y}$ can change for the different classification problems.

Convolutional layers are used as the backbone of the classifier (Li, Liu, Yang, Peng, & Zhou, 2021). More specifically, the model consists of two one-dimensional convolutional layers and three fully connected layers. Rectified linear unit (ReLU) activation functions and max pooling are used after each convolutional layer. A ReLU activation function is also used after the first two fully connected layer. A softmax function is used after the final fully connected layer of the network during inference. Figure 3 displays the architecture of the classifier. A model utilizing long short-term memory (Hochreiter & Schmidhuber, 1997) layers was also tested but yielded poor results on many of the classification problems. The following experiments use all of the collected data as features for the classifier $L = 9$.

The model was trained for 100 epochs using a batch size of 8 and a learning rate of 5×10^{-5}. 60% of the data is used for training the model, and 20% of the data is used to calculate validation loss in order to detect over fitting. The remaining 20% is used for evaluating the model. The hyperparameters for the numerical experiments are displayed in Table 3. The hyperparameters were selected through trial and error experimentation. Future work using this data set could explore optimal hyperparameter selection. The training, validation, and

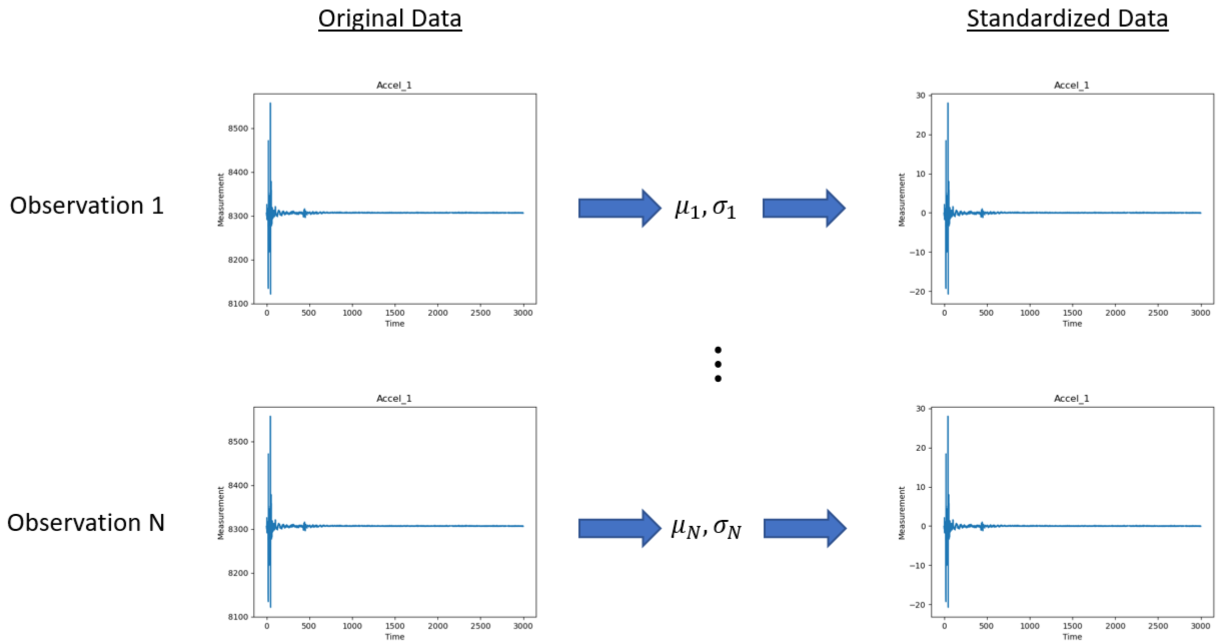


Figure 4. Visual representation for the data standardization process. Each observation sequence is now centered on 0 and has a standard deviation of 1.

Table 4. Class label for each actuator under each classification problem.

Actuator	Damage Detection	Damage Classification	Load Classification	Actuator Classification	New Condition
1	Undamaged	Undamaged	Butterfly	Act1	Undamaged
2	Undamaged	Undamaged	Ball	Act2	Undamaged
3	Damaged	Gear Damage	Ball	Act3	Damaged
4	Damaged	Seal Damage	Butterfly	Act4	Damaged
5	Undamaged	Undamaged	No Load	Act5	Undamaged
6	Undamaged	Undamaged	No Load	Act6	Undamaged
# Classes	2	3	3	6	2

test sets are randomly sampled. The test set accuracy is calculated and a confusion matrix is created for each classification problem. Due to the small size of the training set, the training process can overfit or get stuck in local minima. The results can vary depending on a number of stochastic factors including the training/validation/test split and the initial parameters of the model.

The following sections apply the base ML model to five classification problems. The class labels are displayed in Table 4.

5.1. Data Preparation

The data is collected from six individual actuators each with a different load and fault condition. During initial experiments, simple classifiers such as a random forest (Breiman, 2001) could achieve perfect classification using the mean of each observation as the features. This may indicate that each actuator possesses unique characteristics that are captured by the data collection sensors. Because each actuator is used to generate a fault and load condition, it might be difficult to determine if a model is learning to identify a fault or a specific actuator. Therefore, for the numerical experiments, each observation is standardized using the mean and standard deviation of the time series example (see Figure 4). We believe that this standardization process simulates a scenario where load conditions and faults are generated from the same actuator. We also believe this would be beneficial in an operational setting. Individual actuators can be installed in a variety of environments and this could create offsets that are detected by the sensors. In addition, sensors may have variance during the installation and setup process that could also create offsets.

During the data collection process, each actuation cycle was approximately one second but data was collected for three seconds in order to collect any signals that might occur after the actuation cycle is completed. Only the first second of the data is used for the numerical experiments $T = 1000$ because no activity was visually observed after the actuation cycle.

5.2. Damage Classification

The objective of the damage classification machine learning problem is to detect any type of damage in the system. This is constructed as a binary classification problem $\mathcal{Y} = \{0, 1\}$ where $y = 0$ is the undamaged class and $y = 1$ is the damaged class. Figure 5a displays the training and validation loss, and Figure 5b displays the confusion matrix for the test set. Note that the validation loss is less than the training loss at the beginning of the training process. This is due to summing the loss over each batch, and the validation set having fewer batches. As can be seen by the loss, the model converges quickly, and the confusion matrix shows that the model has an accuracy of 1.0. The trained model is easily able to identify either type of damage when examples from each actuator are present in the training set.

5.3. Damage Type Classification

The objective of damage type classification machine learning problem is to identify the type of damage if present. This is a multiclass classification problem with three classes $\mathcal{Y} = \{1, 2, 3\}$ where $y = 1$ is the undamaged class, $y = 2$ is the gear damage class, and $y = 3$ is the seal damage class. Figure 6a displays the training and validation loss, and Figure 6b displays the confusion matrix for the test set. As can be seen by the loss, the model converges quickly, and the confusion matrix shows that the model has an accuracy of 1.0. The trained model is easily able to identify the type of damage when examples from each actuator are present in the training set.

5.4. Load Classification

The objective of the load classification machine learning problem is to identify the load on each actuator. This is a multiclass classification problem with three classes $\mathcal{Y} = \{1, 2, 3\}$ where $y = 1$ is the butterfly valve load, $y = 2$ is the ball valve load, and $y = 3$ is no load. Figure 7a displays the training and validation loss, and Figure 7b displays the confusion matrix for the test set. As can be seen by the loss, the model converges quickly, and the confusion matrix shows that the model has an accuracy of 1.0. The trained model is easily able to distinguish between the three loads when examples from each actuator are present in the training set.

5.5. Actuator Classification

The objective of the actuator classification machine learning problem is to identify each actuator. This experiment was conducted as a gauge for the uniqueness of each actuator. This is a multiclass classification problem with six classes $\mathcal{Y} = \{1, 2, 3, 4, 5, 6\}$ where the class label corresponds to the actuator number. Figure 8a displays the training and validation loss, and Figure 8b displays the confusion matrix for the test set. As can be seen by the loss, the model converges quickly, an the confusion matrix shows that the model has an accuracy of 0.99.

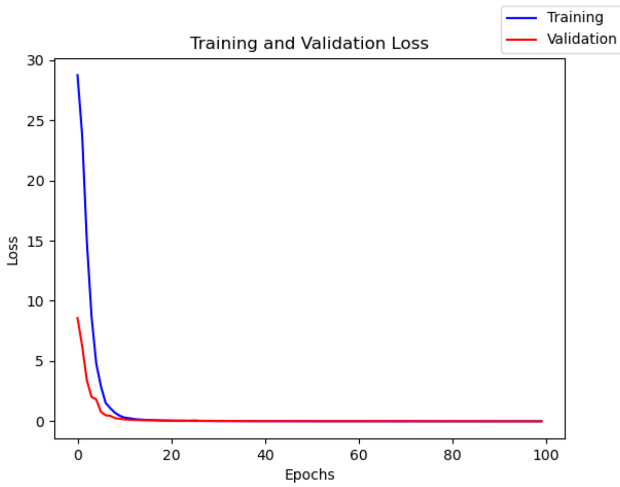
5.6. New Condition

The objective of the new condition machine learning problem is to test if a model can identify a damage condition that is not present in the training set. This is a binary classification problem $\mathcal{Y} = \{0, 1\}$ where $y = 0$ is the undamaged class and $y = 1$ is the damaged class. Specifically, actuators 1, 2, and 3 are used for training, and actuators 4, 5, and 6 are used for testing. In terms of transfer learning, actuators 1, 2, and 3 would be considered the source learning problem, and actuators 4, 5, and 5 would be considered the target learning problem. Figure 9a displays the training and validation loss, and Figure 9b displays the confusion matrix for the test set. As can be seen by the loss, the model converges quickly but the model cannot identify the new damage type. The accuracy of the model drops to 0.625. More specifically, the model has difficulty distinguishing the damaged state from the undamaged state where 85 of the 115 damaged observations are classified as undamaged. The model is not able to generalize to new damage types. The input space for the source and the target are the same $\mathcal{X}_S = \mathcal{X}_T$ because the input is composed of the same sensor information and extracted features. The label space may be considered the same $\mathcal{Y}_S = \mathcal{Y}_T$ because the source and target learning problems are both binary classification. However, the fault types in the label space are different in the two learning problems. The marginal and conditional distributions in the source and target learning problems may also be different. Therefore, it is difficult to determine which part of the domain and task are contributing to the drop in performance.

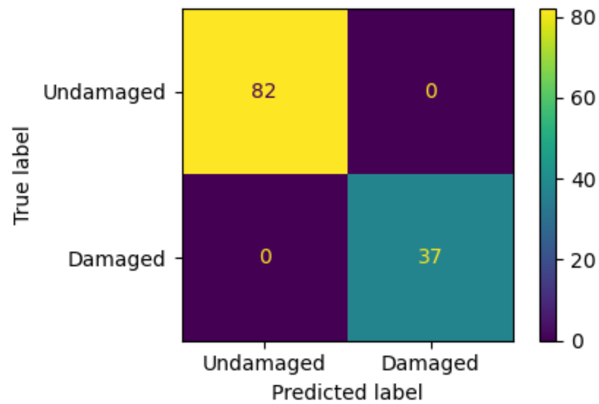
6. CONCLUSION

This paper presents a hydraulic actuator dataset and a series of baseline numerical experiments using machine learning models. This data set is unique because it is composed of data collected from several actuators under different fault conditions and loads. The numerical experiments demonstrate that a machine learning model can be trained to distinguish many classes depending on the machine learning problem. However, the model has difficulty generalizing to fault types that are not present in the training set, as is demonstrated by the new condition numerical experiments. The presented dataset could be used for developing models and methods for generalizing across individual actuators and the introduction of new data types. These problems are critical to address when designing a PHM system with AI/ML as the backbone.

One avenue for future work is to investigate the use of transfer

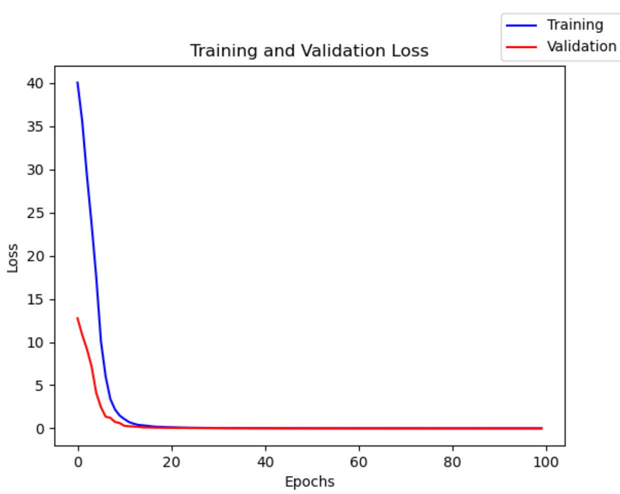


(a) Training loss

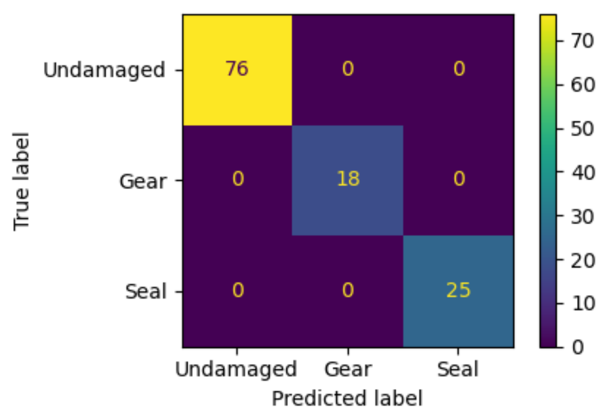


(b) Test confusion matrix

Figure 5. Damage classification results.

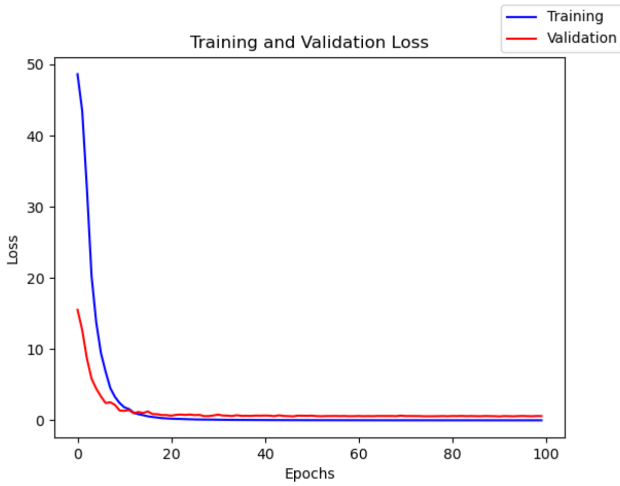


(a) Training loss

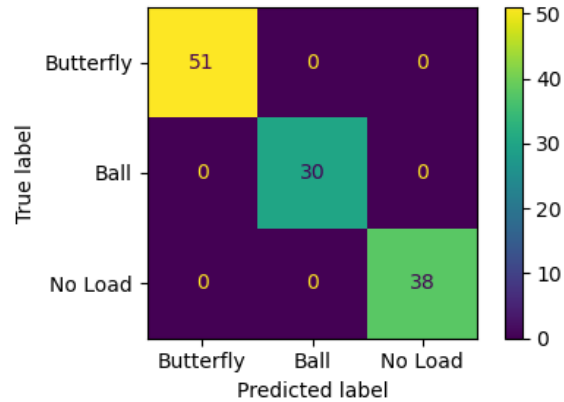


(b) Test confusion matrix

Figure 6. Damage type classification results.

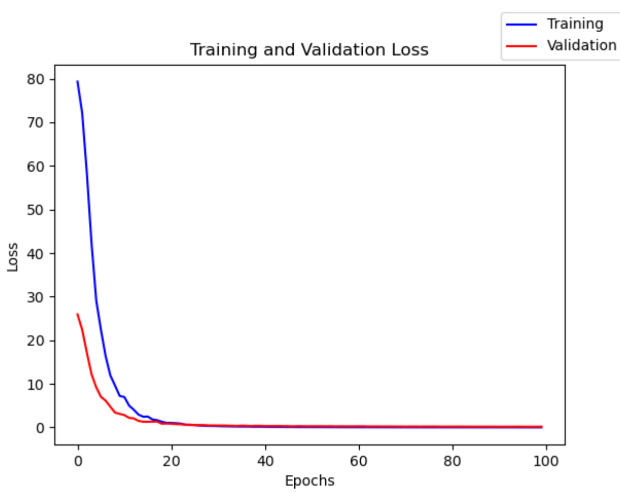


(a) Training loss

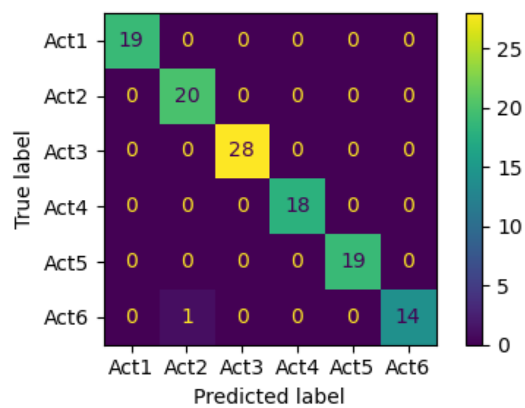


(b) Test confusion matrix

Figure 7. Load classification results.

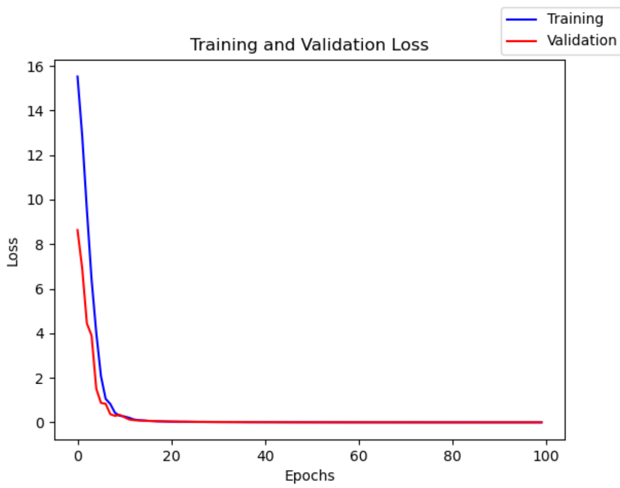


(a) Training loss

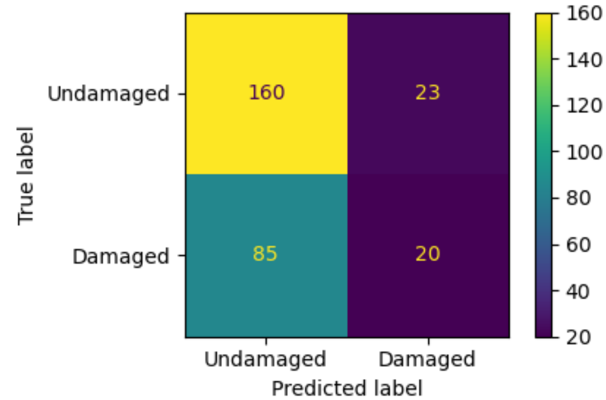


(b) Test confusion matrix

Figure 8. Actuator classification results.



(a) Training loss



(b) Test confusion matrix

Figure 9. New condition classification results.

learning techniques to address model generalization. There are numerous methods for transfer learning that could be applied to this data set. Transductive transfer learning techniques, which focus on domain adaptation and co-variate shift, assume that labels are available in the source but not available in the target (Pan & Yang, 2009). Modern transfer learning focuses on fine-tuning deep learning models in the target domain. For the model used in the presented numerical experiments, the feature extraction CNNs layers could be held fixed and the linear layers that map a feature space to class could be retrained. Furthermore, when new classes enter the environment, the final linear layer can be retrained to account for the new class.

Another interesting area for future work is exploring data augmentation methods for generalizing models. The numerical experiments demonstrate that the models can converge very quickly on some of the ML problems. This indicates that the models are identifying some aspect of the data and keying on that aspect for classification. It is likely that these models will not generalize well. Data augmentation modifies the data and can improve generalization capabilities. Furthermore, data augmentation is a method for exploring and characterizing the robustness of the model to sensor errors.

ACKNOWLEDGMENT

This material is based upon work supported by the Naval Sea Systems Command under Contract No N00024-17-C-4008 and Contract No N68335-21-C-0282.

REFERENCES

Adams, S., & Beling, P. A. (2019). A survey of feature selection methods for Gaussian mixture models and hidden markov models. *Artificial Intelligence Review*, *52*, 1739–1779.

Adams, S., Beling, P. A., Farinholt, K., Brown, N., Polter, S., & Dong, Q. (2016). Condition based monitoring for a hydraulic actuator. In *Annual conference of the phm society* (Vol. 8).

Adams, S., Cody, T., Beling, P. A., Polter, S., & Farinholt, K. (2020). Hierarchical classification for unknown faults. In *2020 ieee international conference on prognostics and health management (icphm)* (pp. 1–8).

Adams, S., Meekins, R., Beling, P. A., Farinholt, K., Brown, N., Polter, S., & Dong, Q. (2017). A comparison of feature selection and feature extraction techniques for condition monitoring of a hydraulic actuator. In *Annual conference of the phm society* (Vol. 9).

Adams, S., Meekins, R., Beling, P. A., Farinholt, K., Brown, N., Polter, S., & Dong, Q. (2019). Hierarchical fault classification for resource constrained systems. *Mechanical Systems and Signal Processing*, *134*, 106266.

Adams, S., Meekins, R., Farinholt, K., Hipwell, N., Desrosiers, M., & Beling, P. A. (2018). One-class support vector machines for structural health monitoring on wave energy converters. In *2018 ieee international conference on prognostics and health management (icphm)* (pp. 1–8).

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

Cody, T., Adams, S., Beling, P., & Freeman, L. (2022). On valuing the impact of machine learning faults to cyber-physical production systems. In *2022 ieee international conference on omni-layer intelligent systems (coins)* (pp. 1–6).

Cody, T., Adams, S., & Beling, P. A. (2017). Unsupervised deep learning for gear health monitoring. In *Annual conference of the phm society* (Vol. 9).

Cody, T., Adams, S., & Beling, P. A. (2019). A systems theoretic perspective on transfer learning. In *2019 ieee international systems conference (syscon)* (pp. 1–7).

Cody, T., Adams, S., & Beling, P. A. (2022). Empirically measuring transfer distance for system design and operation. *IEEE Systems Journal*, *16*(3), 4962–4973.

Cody, T., Adams, S., Beling, P. A., Polter, S., Farinholt, K., Hipwell, N., ... Meekins, R. (2019). Transferring random samples in actuator systems for binary damage detection. In *2019 ieee international conference on prognostics and health management (icphm)* (pp. 1–7).

Farinholt, K. M., Chaudhry, A., Kim, M., Thompson, E., Hipwell, N., Meekins, R., ... Polter, S. (2018). Developing health management strategies using power constrained hardware. In *Phm society conference* (Vol. 10).

Fink, O., Wang, Q., Svensen, M., Dersin, P., Lee, W.-J., & Ducoffe, M. (2020). Potential, challenges and future directions for deep learning in prognostics and health management applications. *Engineering Applications of Artificial Intelligence*, *92*, 103678.

Helwig, N., Pignanelli, E., & Schütze, A. (2015). Condition monitoring of a complex hydraulic system using multivariate statistics. In *2015 ieee international instrumentation and measurement technology conference (i2mtc) proceedings* (pp. 210–215).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Lessmeier, C., Kimotho, J. K., Zimmer, D., & Sextro, W. (2016). Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: A benchmark data set for data-driven classification. In *Phm society european conference* (Vol. 3).

Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, *33*(12), 6999–7019.

Meekins, R., Adams, S., Beling, P. A., Farinholt, K., Hipwell, N., Chaudhry, A., ... Dong, Q. (2018). Cost-sensitive classifier selection when there is additional cost information. In *International workshop on cost-sensitive learning* (pp. 17–30).

Meekins, R., Adams, S., Farinholt, K., Polter, S., & Beling, P. A. (2020). ROC with cost Pareto frontier feature selection using search methods. *Data-Enabled Discovery and Applications*, *4*, 1–13.

Neupane, D., & Seok, J. (2020). Bearing fault detection and diagnosis using case western reserve university dataset with deep learning approaches: A review. *Ieee Access*, *8*, 93155–93178.

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, *22*(10), 1345–1359.

Rezaeianjouybari, B., & Shang, Y. (2020). Deep learning for prognostics and health management: State of the art, challenges, and opportunities. *Measurement*, *163*, 107929.

Saxena, A., & Goebel, K. (2008). Turbofan engine degradation simulation data set. *NASA ames prognostics data repository*, *18*.

Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 international conference on prognostics and health management* (pp. 1–9).

Surucu, O., Gadsden, S. A., & Yawney, J. (2023). Condition monitoring using machine learning: A review of theory, applications, and recent advances. *Expert Systems with Applications*, *221*, 119738.

Tsui, K. L., Chen, N., Zhou, Q., Hai, Y., & Wang, W. (2015). Prognostics and health management: A review on data driven approaches. *Mathematical Problems in Engineering*, *2015*(1), 793161.

Wen, Y., Rahman, M. F., Xu, H., & Tseng, T.-L. B. (2022). Recent advances and trends of predictive maintenance from data-driven machine prognostics perspective. *Measurement*, *187*, 110276.