A Comparison of Feature Selection and Feature Extraction Techniques for Condition Monitoring of a Hydraulic Actuator

Stephen Adams[1], Ryan Meekins[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
rmm6ey@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 many applications, there are a number of data sources that can be collected and numerous features that can be calculated from these data sources. The error of big data has lead many to believe that the larger the data, the better the results. However, as the dimensionality of the data increases, the effects of the curse of dimensionality become more prevalent. Further, a large feature set also increases the computational cost of data collection and feature calculation. In this study, we evaluated four dimensionality reduction techniques as part of a system for condition monitoring of a hydraulic actuator. Two feature selection techniques, ReliefF and variable importance, and two feature extraction techniques, principal component analysis and autoencoders, are used to reduce the input into three classification algorithms. We conclude that variable importance in conjunction with the random forest algorithm outperforms the other dimensionality reduction techniques. Feature selection has the added advantage of being able to remove data sources and features from the data collection and feature calculation process that are not present in the relevant feature subset.

1. INTRODUCTION

In many prognostics and health management (PHM) applications, there are a number of different data sources that could be collected. Further, there are numerous types of features that can be calculated from these data sources and used in conjunction with machine learning algorithms to predict current or future health states. The abundance of data sources and possible features can lead to extremely large feature sets. As the size of the feature sets grows, the curse of dimensionality (Jain, Duin, & Mao, 2000; Keogh & Mueen, 2011) begins to degrade the performance of machine learning algorithms. Data sets with a large number of observations must be collected to overcome the downside of high-dimensional data. However, in application, collecting very large data sets may be very expensive or impossible due to limited resources.

Feature selection is one possible method for alleviating the effects of the curse of dimensionality. Feature selection is the process of selecting a subset of relevant features from the larger set of collected features (Blum & Langley, 1997; Dash & Liu, 1997; Guyon & Elisseeff, 2003). Feature extraction is another method for addressing the curse of dimensionality and constructs new features in a lower dimensional space than the original feature set (Jain et al., 2000). Both of these methods are considered dimensionality reduction techniques. The primary difference between these two methods is that feature extraction requires all of the collected data while feature selection only requires the data associated with the features in the selected relevant subset.

In this paper, we evaluate feature selection and feature extraction for condition monitoring of a hydraulic actuator. The data set for the actuator and some initial results on classification accuracy and computation time were presented at the 2016 Annual Conference for Prognostics and Health Management (Adams, Beling, Farinholt, et al., 2016). This prior work used small feature sets of only 5 and 6 features so no dimensionality reduction was necessary. We build upon our previous work by first expanding the feature set to over 100 features. We then evaluate two feature selection methods and two feature extraction methods. Our objective is to characterize the tradeoffs between lower dimensional data and classification accuracy.

The ultimate goal of this project is to develop distributed hardware that can monitor the health state of the actuator while consuming as little power as possible. It is necessary to minimize power consumption because these hydraulic actuators will be used on Naval vessels where electrical power may be limited or not available. The distributed hardware is being designed to run on a very small battery for several years. The previous study evaluated several classification algorithms on error rate and computation time. Computation time is used as a surrogate for power consumption where it is assumed that less computation equals less power consumed. The previous study concluded that classification trees yield a satisfactory trade-off between accuracy and computation time. The random forest algorithm outperformed the classification trees in terms of accuracy, but the computation time was much greater.

Our previous work did not evaluate the cost of collecting data and calculating features. In the presented study, we begin by assessing the performance of classifiers on feature sets of various sizes. There is significant evidence that feature extraction techniques can provide new feature representations that are both smaller in size than the original data set and representative of the process (Hinton & Salakhutdinov, 2006; Vincent, Larochelle, Bengio, & Manzagol, 2008; Vincent, Larochelle, Lajoie, Bengio, & Manzagol, 2010). Many of these advances in feature extraction are due to the development of deep learning techniques. However, feature selection could offer the added advantage of eliminating the need for collecting some data sources or calculating some features, and thus reducing the cost of the data collection process. Further, we make the assumption that when comparing feature sets a smaller feature set is preferred as long as the predictive accuracy is the same. Further, in some applications, a smaller less costly feature set with lower predictive ability might be preferable over a larger more costly feature set with better predictive ability.

This paper is organized in the following manner. Section 2 outlines background information on feature selection and feature extraction. Section 3 describes the experimental setup and the collected data. Section 4 describes the dimensionality reduction techniques explored in this study. Section 5 presents the results from numerical experiments performed on the data. Section 6 presents our conclusions and plans for future work.

2. BACKGROUND

With the growth in the size of collected data sets, dimensionality reduction has become a larger part of the modeling process. In fields where high-dimensional data is abundant such as bioinformatics, dimensionality reduction has become a prerequisite to model construction (Saeys, Inza, & Larrañaga, 2007). In this section, we give background information on feature selection and feature extraction and provide a brief review of their use in PHM activities.

There are numerous methods for feature selection. One possible method is to exhaustively test every possible combination of features, but this approach quickly becomes impractical as the number of features grows. There are general feature selection techniques that can be applied to any model (Almuallim & Dietterich, 1991; John, Kohavi, & Pfleger, 1994; Kira & Rendell, 1992; Kohavi & John, 1997) and model specific techniques (Adams, Beling, & Cogill, 2016; Law, Figueiredo, & Jain, 2004). Senoussi *et al.* (Senoussi, Chebel-Morello, Dena, & Zerhouni, 2011) develop a method for feature selection and classification for general fault detection systems. In a more specific application, the Euclidean distance technique has been used to select features for gear box fault diagnostics (Li, Zhao, Yang, Zhao, & Teng, 2014). An ensemble of feature selection techniques were used to select relevant control variables for IT infrastructure monitoring (Paljak, Kocsis, Égel, Tóth, & Pataricza, 2009). Minimum-redundancy maximal-relevance has been used with support vector machines in a railcar diagnostics application (Shahidi, Maraini, & Hopkins, 2016).

As with feature selection, there are a number of feature extraction techniques. Generally, feature extraction is an unsupervised process meaning class labels are not needed to extract the features. Many view this as a very attractive quality because labeled data can often be difficult and expensive to collect. Further, the unsupervised nature of feature extraction makes the methods robust to mislabeled data. One of the most popular feature extraction techniques is principal component analysis (PCA) and has been used for fault detection (Harmouche, Delpha, & Diallo, 2014), damage detection (Shao, Hu, Wang, & Qi, 2014), and remaining useful life estimation (Benkedjouh, Medjaher, Zerhouni, & Rechak, 2015; Le Son, Fouladirad, Barros, Levrat, & Iung, 2013). There are several examples of autoencoders being used in PHM applications including diagnostics of rotating machinery (Verma, Gupta, Sharma, & Sevakula, 2013), health indicator extraction (Hu, Palmé, & Fink, 2016), and structural health monitoring (Sarkar, Reddy, Giering, & Gurvich, 2016). Linear discriminant analysis has been used to extract features for partial discharge diagnostics (Yan, 2012).

3. EXPERIMENTAL SETUP AND DATA

In this section, we briefly outline the experimental setup and the data set. For the numerical experiments, we use the data collected in (Adams, Beling, Farinholt, et al., 2016). The experimental system is composed of 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*. A 5-horsepower pump supplies 3000 psi of pressure at a 2 GPM flow rate to produce an actuation stroke time of approx-

imately one second. We are interested in using data collected during this actuation stroke to monitor the condition of the actuator. An adjustable pressure relief valve with a range of 400-3000 psi sets the maximum system pressure. A turbine flow meter is located on the inlet to the side of the actuator. Temperature and pressure sensors are located at each inlet and the gear case relief port. A rotary position sensor tracks the angle of the actuator through its actuation stroke. Resistive torque from the matched actuator is produced by the backpressure generated when forcing fluid through a flow restriction. A friction braking can be applied using a manual pump with a precision pressure gauge. Figures 1 and 2, both originally printed in (Adams, Beling, Farinholt, et al., 2016), display a schematic and a labeled photograph of the experimental test stand..

This test stand is designed to simulate a number of common faults for a hydraulic actuator. Further, it can simulate the fault at different severity levels. A total of five different fault conditions are simulated with a varying number of severity levels. Several baseline conditions are also simulated. All baseline conditions are considered normal operating conditions and grouped into a single class. Under each simulated condition, data from multiple actuation strokes are collected.

In the numerical experiments, we study two separate problems. The first, called the 6-Class problem, attempts to classify an observation into either the baseline condition or one of the five fault conditions. The second, called the 24-Class problem, attempts to classify an observation into either the baseline condition or one of the 23 conditions with severity. A binary problem was also studied in our previous work, but it is omitted from this study. Table 1 displays class labels, a description of each condition, and the number of observations for each condition.

Data from three pressure gauges, three temperature gauges, a flow meter, an angular position sensor, and an accelerometer are collected during actuation. From these nine data streams several features can be calculated. The mean, standard deviation, skewness, and kurtosis are calculated on each of these streams during actuation. These four moments are also calculated on the difference in pressure between pressure gauge 1 and 2. These four moments are also calculated on these data sources for 5 and 10 data points past the end of the actuation stroke (see Figure 3). A binary variable representing the direction of the actuation stroke is also added to the feature set. This results in a feature set of 121 features (120 calculated features plus the binary direction feature).

4. METHODS

In this section, we outline the feature selection and feature extraction methods used on the collected feature set described in the previous section. Both feature selection methods rank features from most important to least important and provide a weight symbolizing the importance of each feature. Generally, the four methods used in this paper were selected due to their prevalence in the literature and their ease of use. More specifically, ReliefF and PCA have both been used as dimensionality reduction methods in numerous machine learning and PHM applications. Deep feature extraction using autoencoders was selected because of the growing popularity of deep learning methods in machine learning and PHM applications. Variable importance was selected because this metric is easily extracted from the training phase of the random forest algorithm. Each of these methods has advantages and disadvantages which will be discussed in their corresponding subsections.

4.1. ReliefF

Relief is a feature selection algorithm first introduced in (Kira & Rendell, 1992). This algorithm finds features that are statistically relevant to the provided class label. The original version of the algorithm was designed for a binary classification problem. At each instance of the algorithm, an observation is chosen at random and compared to the observation in each class that is closest based on a distance measure. The observation from the same class is called a *near hit* and the observation from the other class is called a *near miss*. The weight w_l for the l^{th} feature is updated by

$$w_l = w_l - (x_l - x_l^h) + (x_l - x_l^m), \tag{1}$$

where x_l is the l^{th} feature value from the randomly chosen sample, x_l^h represents the feature value from the near hit, and x_l^m represents the feature value from the near miss.

ReliefF (Kononenko, 1994) is an extension of the original Relief algorithm that can be used on multi-class problems. In this paper, the MATLAB implementation of the ReliefF algorithm is used. This version of the algorithm allows for the number of closest neighbors to be selected. Ten neighbors were chosen for both the 6-Class and 24-Class problems. Selecting the number of neighbors in the ReliefF algorithm can affect the results and is the primary disadvantage to this method.

4.2. Variable Importance

The random forest algorithm can produce a variable importance metric. This metric is calculated by permuting the features over all the trees in the forest and then calculating the decrease in the out-of-bag error rate. The variable importance metric as produced by MATLAB is this decrease in error divided by the standard deviation. Features that are relevant should cause the error to drop more than features that are irrelevant. We sort the variable importance of each feature in the feature set and use this to rank features.

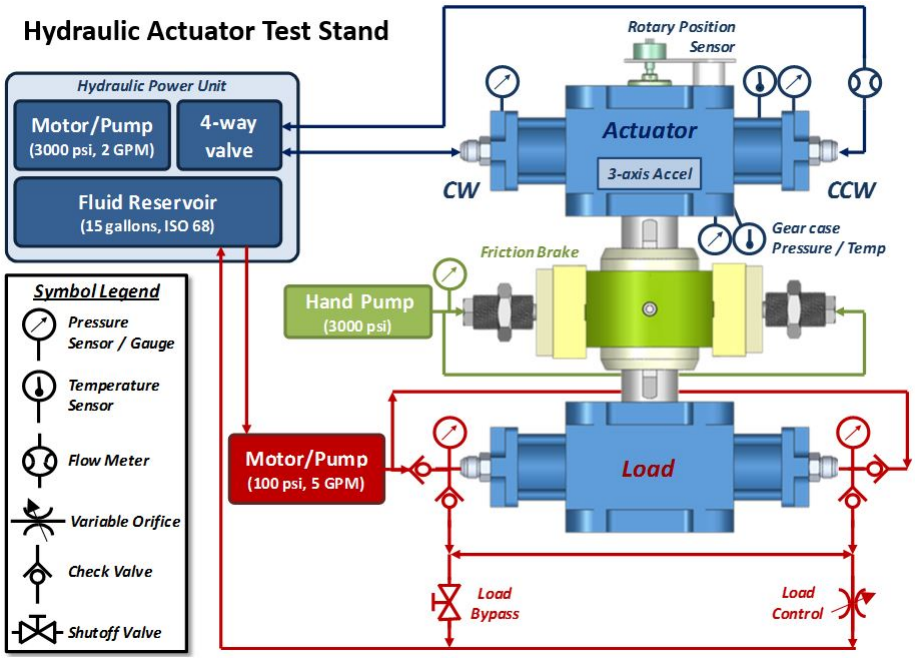

Figure 1. Actuator test stand hydraulic layout, including instrumentation locations (CW, CCW indicate direction of shaft rotation when pressure is applied) (Adams, Beling, Farinholt, et al., 2016).

There are several drawbacks to using this method as a general feature selection technique. First, these metrics are derived from the trained random forest and might not generalize to other classifiers. Second, research has shown that this metric can be biased when the scale of the feature differs greatly or the number of categories for categorical variables differ greatly (Strobl, Boulesteix, Zeileis, & Hothorn, 2007). Third, this measure can also give preference to strongly correlated features (Strobl, Boulesteix, Kneib, Augustin, & Zeileis, 2008). The primary advantage of this method is that it is easily extracted from the training phase of the random forest algorithm, and that it tends to outperform other feature selection methods when used with the random forest algorithm because it is specific to this classifier.

4.3. PCA

PCA is one of the oldest and most widely used feature extraction techniques. It uses an orthogonal transformation to project features that may be correlated into a new feature space where there is minimum correlation between the new features. Let $\mathbf{X}$ be a matrix of data with N observations and L features and let $\mathbf{x}_{(n)}$ represent the n^{th} row vector. This data is transformed into the principal component space by $\mathbf{t}_{k(n)} = \mathbf{w}_k \cdot \mathbf{x}_{(n)}$, where $\mathbf{w}_k$ is the L-dimension loading vector and $\mathbf{t}_{k(n)}$ is the k^{th} component score. The weight of the first principal component $\mathbf{w}_1$ is found by

$$\mathbf{w}_1 = \text{argmax} \left\{ \frac{\mathbf{w}^T \mathbf{X}^T \mathbf{X} \mathbf{w}}{\mathbf{w}^T \mathbf{w}} \right\}. \tag{2}$$

The subsequent principal components can be found by subtracting the first k components from the data

$$\hat{\mathbf{X}}_k = \mathbf{X} - \sum_{m=1}^{k-1} \mathbf{X} \mathbf{w}_m \mathbf{w}_m^T, \tag{3}$$

and then finding the loadings

$$\mathbf{w}_k = \text{argmax} \left\{ \frac{\mathbf{w}^T \hat{\mathbf{X}}^T \hat{\mathbf{X}} \mathbf{w}}{\mathbf{w}^T \mathbf{w}} \right\}. \tag{4}$$

Generally, a majority of variance resides in the first few principal components. The dimensionality of the transformed space can be reduced by only using the first k components.

PCA is a well developed method for feature extraction. The primary disadvantages have to do with the underlying assumptions such as that the component with the most variance is the most important and that the collected data is Gaussian. A majority of the collected data streams in this paper roughly follows a Gaussian distribution.

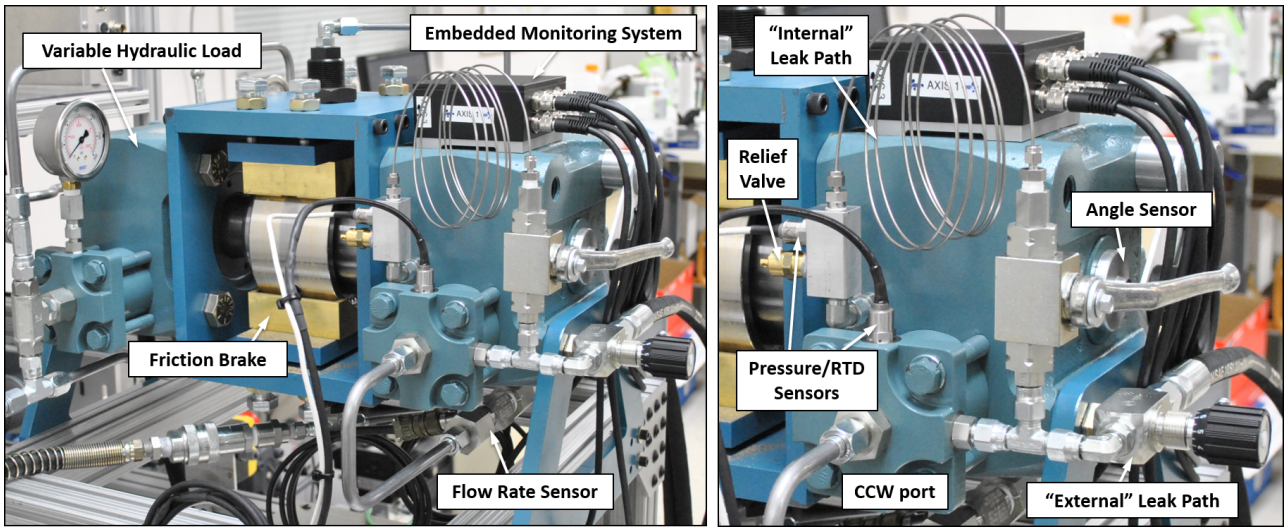

Figure 2. Detailed view of actuator/load, instrumentation locations, and leakage paths (Adams, Beling, Farinholt, et al., 2016).

Table 1. Fault Conditions and Class Labels

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

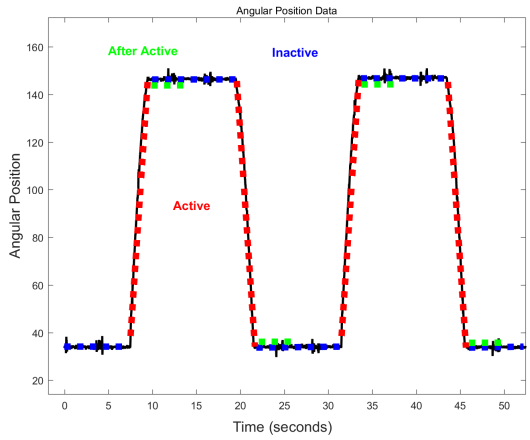

Figure 3. Angular position data. The red squares indicate data collected during the actuation stroke or the active region. The blue squares indicate data collected at rest or inactive. The green squares indicate the after active region used for calculating some features.

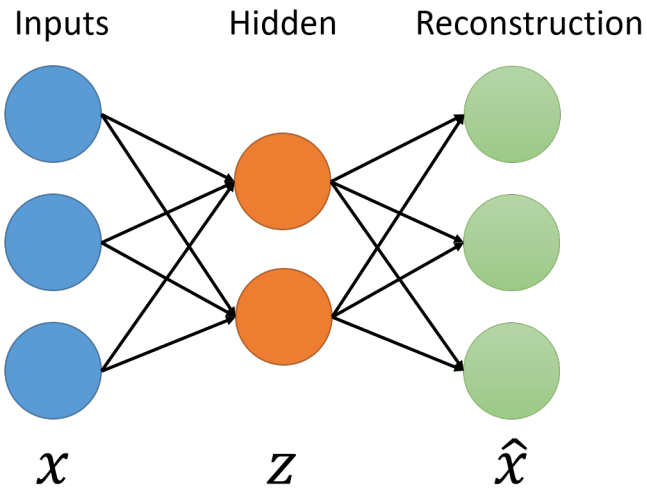

Figure 4. Basic structure of a single layer autoencoder.

4.4. Autoencoders

Autoencoders are an unsupervised feature extraction technique based on neural networks. Here, we explain an autoencoder with a single hidden layer (Figure 4) but this can be easily expanded to multiple layers. Assume that there are L inputs represented by $x_1, ..., x_L$ and that the hidden layer contains M nodes. The values at the M hidden nodes are represented by

$$z_m = h\left(\sum_{l=1}^{L} \omega_l x_l\right), \tag{5}$$

where $h(\cdot)$ is a non-linear transformation function and w represents weights. This is called the encoding step of the autoencoder. The decoding step reconstructs the input signals, represented by $\hat{x}$ using the same weights. The autoencoder is trained using back propogation of the loss between the inputs and the reconstruction

$$L(x, \hat{x}) = ||x - \hat{x}||^2. \tag{6}$$

Autoencoders have been growing in popularity due to their success in extracting features in several domains. The primary disadvantage of this method is selecting the architecture of the network, i.e. the number of layers and the number of nodes in each layer. There is no method for this other than trial and error.

5. NUMERICAL EXPERIMENTS

In this section, we outline the numerical experiments performed on the data set. We begin by assessing the error and total cost of the full feature set. Total cost refers to the sum of the computation time for feature calculation and the average testing time. All experiments are performed in MATLAB and run on the University of Virginia high performance computing system – Rivanna. Basic MATLAB functions are used for feature calculation and classification. The computation time is calculated using the *tic* and *toc* functions in MATLAB. We then perform test on a reduced feature set that removes features with a large number of missing values. We conclude by evaluating the dimensionality reduction techniques.

5.1. Full Feature Set

In order to establish a baseline, we first perform numerical experiments on the full feature set. This set contains 121 features and has several observations with missing values. Based on the experiments in (Adams, Beling, Farinholt, et al., 2016), we evaluate three types of classifiers: *k*-nearest-neighbor (KNN) (Bishop, 2006; Duda, Hart, & Stork, 2001; Murphy, 2012), random forest (RF) (Breiman, 2001; Murphy, 2012), and classification trees (Tree) (Bishop, 2006; Duda et

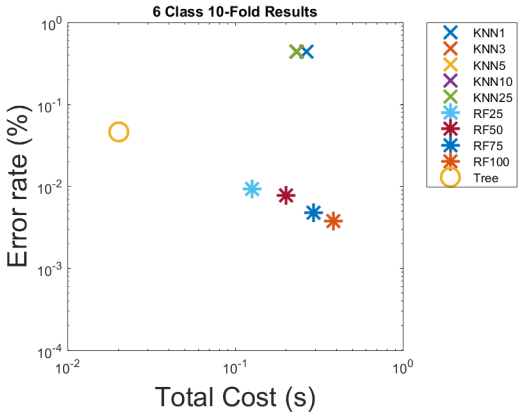

Figure 5. Error and total cost for 6 class 10-fold experiments.

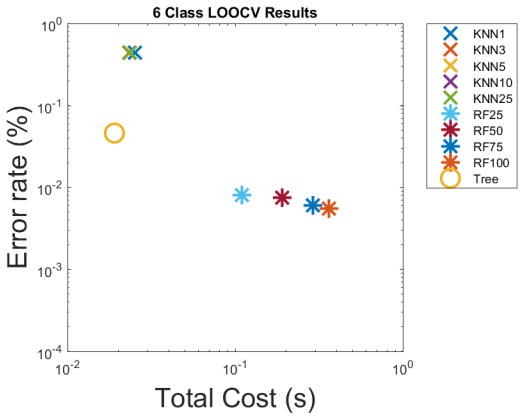

Figure 6. Error and total cost for 6 class LOOCV experiments.

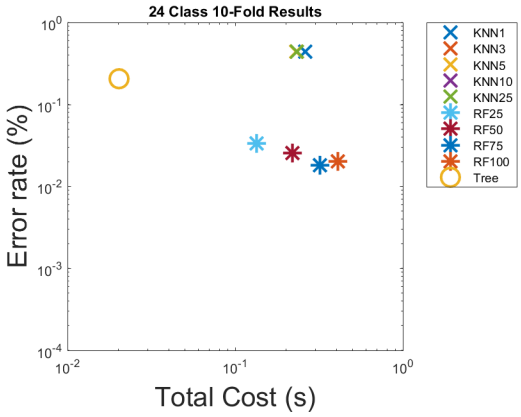

Figure 7. Error and total cost for 24 class 10-fold experiments.

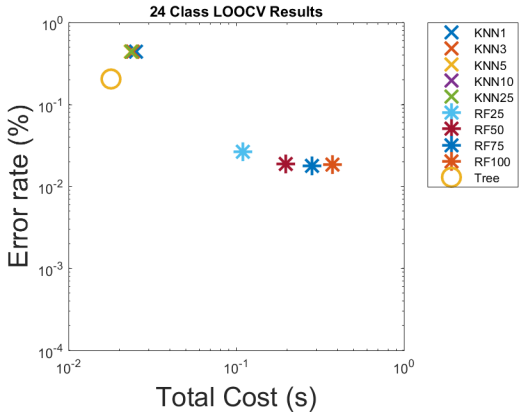

Figure 8. Error and total cost for 24 class LOOCV experiments.

al., 2001; Murphy, 2012). For this set of numerical experiments, KNN classifiers with 1, 3, 5, 10, and 25 neighbors and RF classifiers with 25, 50, 75, and 100 trees were evaluated. Both a 10-fold cross validation (CV) and leave-one-out cross validation (LOOCV) testing scheme were implemented.

Figures 5 to 8 display the results from this set of numerical experiments. The cost of feature extraction is the same for all algorithms in this experiment. The difference in total cost is solely attributed to the computation time for prediction because the feature set is the same for classifiers. It is interesting to note that while there is little difference between the LOOCV and the 10-fold for RF and Tree, the prediction time for KNN is much larger for the 10-fold experiments. The prediction time for all algorithms increases for the 10-fold experiments because the number of observations in the test set is larger, and the presented testing time is the time to predict a label for all observations in the test set. However, the increase in prediction time for the KNN classifier is greater than for the other two algorithms because KNN requires the calculation of distance, which contains several operations, while the other two only have logical comparisons.

The RF algorithm has the lowest error rate but the Tree classifier has the lowest total cost. Selection of the optimal classifier depends on the stake holder assessment of error rate versus total cost. This set of numerical experiments confirms the overall results from the prior study but the larger feature set has decreased the error rate for for the Tree and RF algorithms. The error rate for KNN has increased due to the effects of the curse of dimensionality.

5.2. Reduced Feature Set

The full feature set contains a number of features with a large number of missing values. Our first effort to reduce the dimensionality of the feature set is to remove the features that are missing more than 100 values. This value was chosen

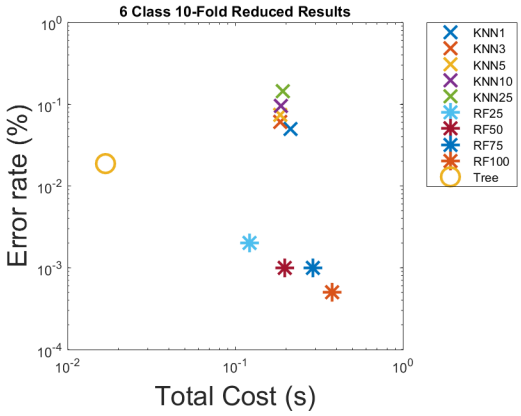

Figure 9. Error and total cost for 6 class 10-fold experiments on reduced feature set.

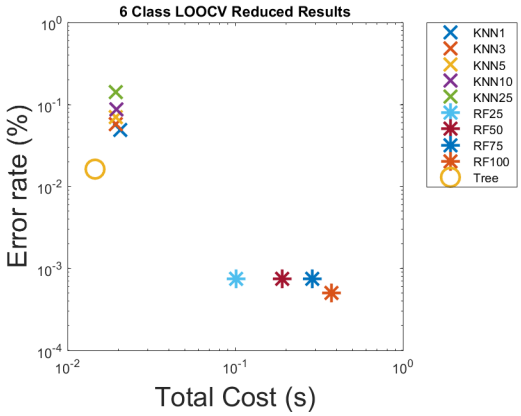

Figure 10. Error and total cost for 6 class LOOCV experiments on reduced feature set.

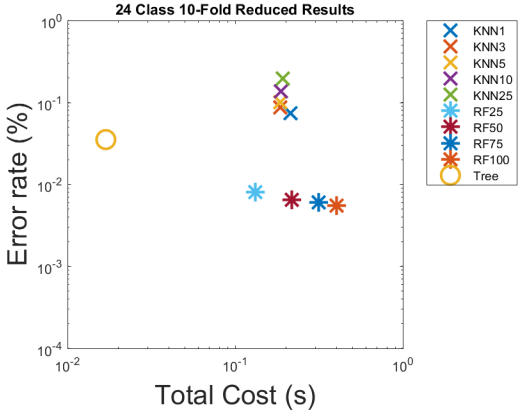

Figure 11. Error and total cost for 24 class 10-fold experiments on reduced feature set.

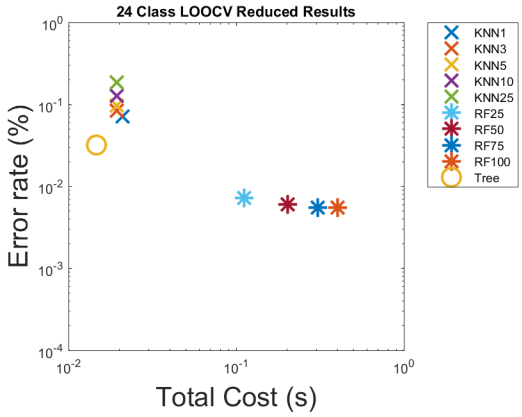

Figure 12. Error and total cost for 24 class LOOCV experiments on reduced feature set.

after observing the number of missing values for each feature. A majority of the features containing missing values had more than 100. This results in a reduced set of 97 features. The numerical experiments performed on the full feature set are performed on this reduced feature set.

Figures 9 to 12 display the results for the numerical experiments on the reduced feature set. Removing the 24 features with a number of missing values only reduce the total feature extraction time by 0.0034 s per observation, but the error rate is reduced for all classifiers. While this reduction in computation time seems small on a per observation basis, with over 4000 observations in the data set this equates to a 13 second reduction in feature calculation time. This set of numerical experiments demonstrates that both error rate and total cost can be reduced by reducing the size of the feature set.

5.3. Feature Selection and Feature Extraction

In this set of numerical experiments, we evaluate two feature selection methods and two feature extraction methods. These experiments use the reduced feature set that have the features with a large number of missing values removed. The reduced feature set still contains a small number of observations with missing values. ReleifF and variable importance can accommodate data with missing values but PCA and autoencoders cannot. When evaluating PCA and autoencoders, all missing values are replaced with the mean of the data for that feature.

The feature selection techniques rank features in order of relevance. We perform experiments by adding features to the subset based on the rankings provided by the selection algorithms, and then performing 10-fold CV to estimate the error associated with the subset. The smallest feature subset in this set of experiments is 5 and one feature is added until all 97 are included. The features are added based on their ranking from the feature selection algorithm. We perform similar experiments with the feature extraction techniques where the size

of the new reduced data set spans from 5 to 97. Based on prior experiments, KNN with 1 neighbor, RF with 100 trees, and Tree are used as the classifiers.

Figures 13 and 14 contain the results of the 6 class and 24 class experiments. The results are consistent across both experiments. The random forest algorithm generally outperforms the other algorithms when using the same feature selection or feature extraction technique. The feature selection methods outperform the feature extraction methods when using RF and Tree. The autoencoder can outperform the other methods when using KNN. This indicates that the autoencoder is extracting information from the data that transforms into meaningful distances and not logical comparisons of being above or below certain thresholds.

The KNN classifier shows some interesting characteristics. First, using the top five features selected by the ReliefF algorithm yields a smaller error rate than all other classifier using the same input dimension except the RF using variable importance on the 6 class problem and outperforms all other classifiers with the same input dimension on the 24 class problem. Second, the error rate when using PCA does not change much with an increase in the number of components. This is most likely due to a large proportion of the variance being characterized by the first few principal components.

6. CONCLUSION

In this study, we evaluated four dimensionality reduction techniques. We have demonstrated that the larger feature set outperforms the smaller feature set collected in our previous study in terms of classification error. The first attempt to reduce the size of the data was to remove features with a large number of missing values. This reduction also improves classification error. Finally, the variable importance feature selection technique yields a relevant feature subset that generally outperforms the other dimensionality reduction techniques. Further, feature selection would allow for data sources and features not in the selected feature subset to be removed from the data collection and feature calculation process. This would reduce the computational cost of this step of the process. The feature extraction techniques would not allow for this because all data sources must be collected and all features must be calculated.

In future work on this project, we plan on further investigating of new features. At this point, we have limited the feature set to statistical moments, but plan on exploring more complex features. Further, the sampling rate of the data has not been investigated. This is another factor that could effect the cost of data collection and feature calculation as well as classification error. Finally, we plan on testing these classification on hardware where power consumption instead of computation time can be used to assess the cost of each algorithm and feature set.

ACKNOWLEDGMENT

This material is based upon work supported by the Naval Sea Systems Command under Contract No N00024-17-C-4008. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Naval Sea Systems Command.

REFERENCES

Adams, S., Beling, P. A., & Cogill, R. (2016). Feature selection for hidden Markov models and hidden semi-Markov models. *IEEE Access*, *4*, 1642–1657.

Adams, S., Beling, P. A., Farinholt, K., Brown, N., Polter, S., & Dong, Q. (2016). Condition based monitoring for a hydraulic actuator. In *Proceedings of the annual conference of the prognostics and health management society 2016.*

Almuallim, H., & Dietterich, T. G. (1991). Learning with many irrelevant features. In *Aaai* (Vol. 91, pp. 547–552).

Benkedjouh, T., Medjaher, K., Zerhouni, N., & Rechak, S. (2015). Health assessment and life prediction of cutting tools based on support vector regression. *Journal of Intelligent Manufacturing*, *26*(2), 213–223.

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

Blum, A. L., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial intelligence*, *97*(1), 245–271.

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

Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent data analysis*, *1*(1-4), 131–156.

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

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, *3*(Mar), 1157–1182.

Harmouche, J., Delpha, C., & Diallo, D. (2014). Incipient fault detection and diagnosis based on kullback–leibler divergence using principal component analysis: Part i. *Signal Processing*, *94*, 278–287.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, *313*(5786), 504–507.

Hu, Y., Palmé, T., & Fink, O. (2016). Deep health indicator extraction: A method based on auto-encoders and extreme learning machines. In *Annual conference of the prognostics and health management society 2016.*

Jain, A. K., Duin, R. P. W., & Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on pattern analysis and machine intelligence*, *22*(1), 4–37.

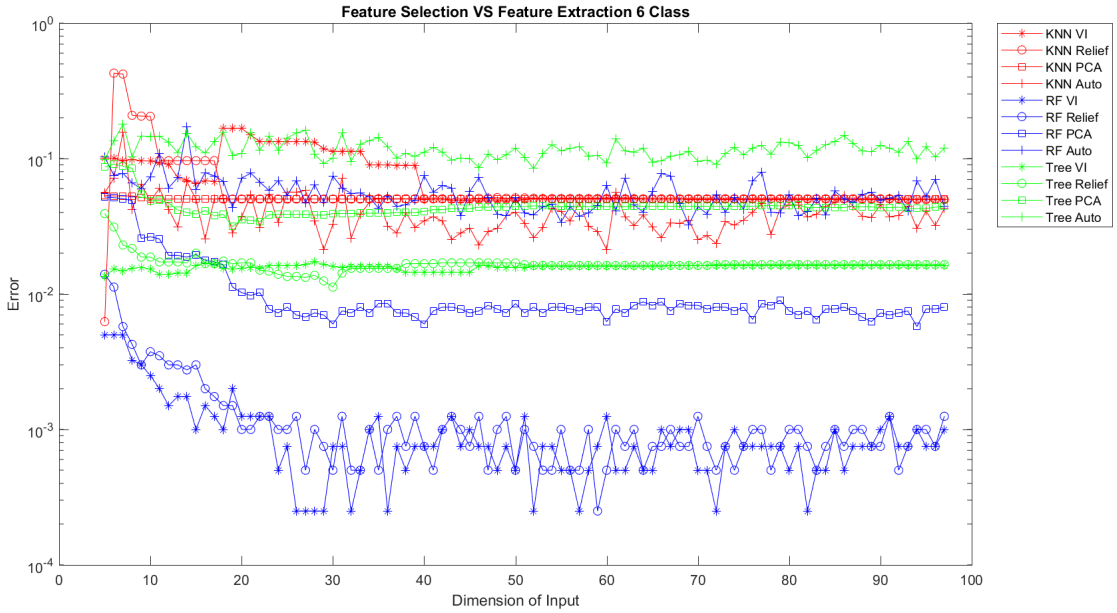

Figure 13. Error rate as dimension of input into each classification algorithm increases for the 6 class problem.

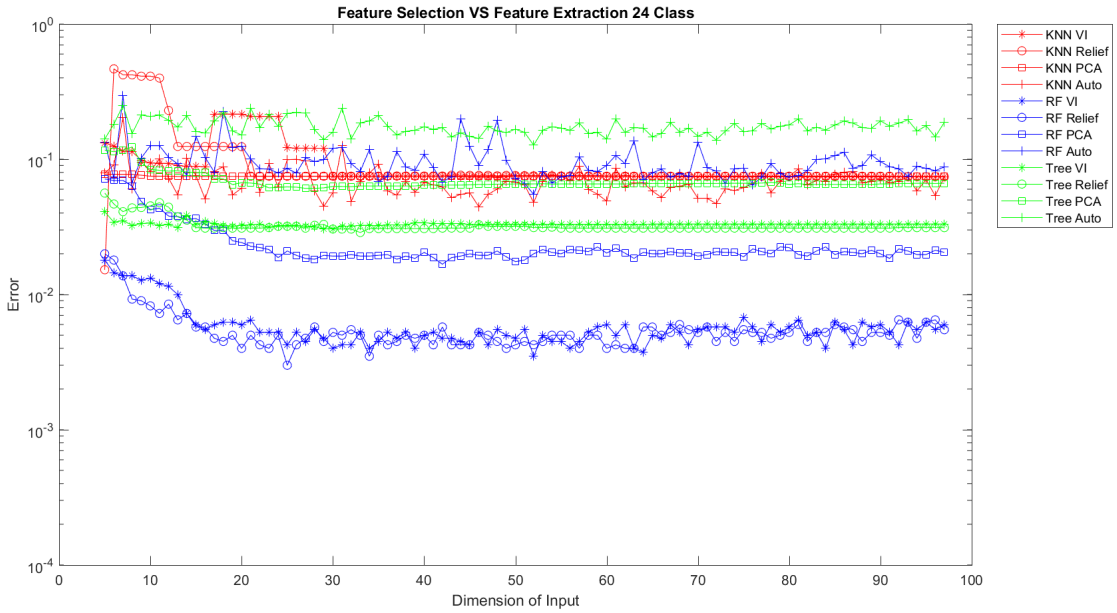

Figure 14. Error rate as dimension of input into each classification algorithm increases for the 24 class problem.

John, G. H., Kohavi, R., & Pfleger, K. (1994). Irrelevant features and the subset selection problem. In *Machine learning: proceedings of the eleventh international conference* (pp. 121–129).

Keogh, E., & Mueen, A. (2011). Curse of dimensionality. In *Encyclopedia of machine learning* (pp. 257–258). Springer.

Kira, K., & Rendell, L. A. (1992). The feature selection problem: Traditional methods and a new algorithm. In *Aaai* (Vol. 2, pp. 129–134).

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial intelligence*, *97*(1-2), 273–324.

Kononenko, I. (1994). Estimating attributes: analysis and extensions of RELIEF. In *European conference on machine learning* (pp. 171–182).

Law, M. H., Figueiredo, M. A., & Jain, A. K. (2004). Simultaneous feature selection and clustering using mixture models. *IEEE transactions on pattern analysis and machine intelligence*, *26*(9), 1154–1166.

Le Son, K., Fouladirad, M., Barros, A., Levrat, E., & Iung, B. (2013). Remaining useful life estimation based on stochastic deterioration models: A comparative study. *Reliability Engineering & System Safety*, *112*, 165–175.

Li, H., Zhao, J., Yang, R., Zhao, J., & Teng, H. (2014). Research on planetary gearboxes feature selection and fault diagnosis based on EDT and FDA. In *Prognostics and system health management conference (PHM-2014 hunan), 2014* (pp. 178–181).

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

Paljak, G. J., Kocsis, I., Égel, Z., Tóth, D., & Pataricza, A. (2009). Sensor selection for it infrastructure monitoring. In *International conference on autonomic computing and communications systems* (pp. 130–143).

Saeys, Y., Inza, I., & Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *bioinformatics*, *23*(19), 2507–2517.

Sarkar, S., Reddy, K. K., Giering, M., & Gurvich, M. R. (2016). Deep learning for structural health monitoring: A damage characterization application. In *Annual conference of the prognostics and health management society 2016.*

Senoussi, H., Chebel-Morello, B., Dena, M., & Zerhouni, N. (2011). Feature selection and categorization to design reliable fault detection systems. In *Annual conference of the prognostics and health management society 2011.*

Shahidi, P., Maraini, D., & Hopkins, B. (2016). Railcar diagnostics using minimal-redundancy maximum-relevance feature selection and support vector machine classification. *International Journal of Prognostics and Health Management: Special Issue Big Data and Analytics*.

Shao, R., Hu, W., Wang, Y., & Qi, X. (2014). The fault feature extraction and classification of gear using principal component analysis and kernel principal component analysis based on the wavelet packet transform. *Measurement*, *54*, 118–132.

Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T., & Zeileis, A. (2008). Conditional variable importance for random forests. *BMC bioinformatics*, *9*(1), 307.

Strobl, C., Boulesteix, A.-L., Zeileis, A., & Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC bioinformatics*, *8*(1), 25.

Verma, N. K., Gupta, V. K., Sharma, M., & Sevakula, R. K. (2013). Intelligent condition based monitoring of rotating machines using sparse auto-encoders. In *Prognostics and health management (PHM), 2013 IEEE conference on* (pp. 1–7).

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on machine learning* (pp. 1096–1103).

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, *11*(Dec), 3371–3408.

Yan, W. (2012). On reducing feature dimensionality for partial discharge diagnosis applications. In *Prognostics and system health management (PHM), 2012 IEEE conference on* (pp. 1–7).