INL/RPT-23-73933

Light Water Reactor Sustainability Program

Extending Data-Driven Anomaly Detection Methods to Transient Power Conditions in Nuclear Power Plants

August 2023

U.S. Department of Energy

Office of Nuclear Energy

DISCLAIMER

This information was prepared as an account of work sponsored by an agency of the U.S. Government. Neither the U.S. Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness, of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. References herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof.

INL/RPT-23-73933

Extending Data-Driven Anomaly Detection Methods to Transient Power Conditions in Nuclear Power Plants

Jacob Farber[1]
Ahmad Al Rashdan (Principal Investigator)[1]
Randall Reese[1]

Arvind Sundaram[2]
Hany Abdel-Khalik[2]

1. Idaho National Laboratory
2. Covert Defenses

August 2023

Prepared for the
U.S. Department of Energy
Office of Nuclear Energy
Light Water Reactor Sustainability Program

EXECUTIVE SUMMARY

Historically, nuclear power plants have operated predominantly at or near full power, meaning that most of the data collected are in this operating regime. Therefore, data-driven anomaly detection methods that are developed using this data can perform well at full power operations. This presents a challenge when the power drops (referred to as a transient) and may result in false alarms due to the lack of historical data at those new power levels. The current approach to handling this challenge is to turn the anomaly detection algorithms off during transients, causing missed detections.

The objective of this effort is to develop data-driven anomaly detection methods that can extend to transient conditions. Specifically, the research hypothesis tested is that anomaly detection methods can be modified and used during the data-poor transient conditions compared with baseline methods used in normal operation conditions. In the context of data-driven approaches, this means that the methods are trained on a mix of predominantly full power data and some sparse transient power data (collectively called the training data), and tested on exclusively transient power data (called the testing data). While the objective here focuses on full power and transient power conditions, the problem can be viewed more broadly as any situation where there are ample data for some condition but limited data for another similar condition.

Within anomaly detection, this study focuses on two common types of approaches: prediction based and feature based. Prediction-based methods often rely on self-supervised learning, where a subset of the data are used to predict another part, enabling learning to fill in the gaps. In this effort, this is performed by either withholding some data from the complete dataset and predicting that withheld data or compressing the complete data to some smaller dimension and using the compressed data to predict the full data. Detection is then based on prediction error between the real measurement and prediction, which for well-trained models is small during normal operations and larger during anomalies.

This effort implemented three prediction-based approaches to solve the transient problem. First, the covariate shift approach is used, which assumes there is a shift in the data distribution from the training data to the testing data, and the method tries to compensate for that shift. Second, this effort developed a new approach called the multiple models approach that calculates two isolated prediction models—one for all correlations except power and one for correlations just from power—and combines them. This is an example of a transfer learning approach, which separates the training data into an abundant source dataset (full power data) and a sparse target dataset (transient power data). Assuming the source and target datasets share some features or properties, the concept is to transfer knowledge learned from the source dataset to the target dataset. Third, another transfer learning approach based on autoencoder models is tested, called the frozen layers approach. In this approach, some of the model weights trained using the source dataset are fixed (frozen), and the target dataset is then used to fine-tune the rest of the weights. In addition to these three approaches that address the transient problem, this effort also implemented baseline approaches that use prediction-based models without accounting for the limited amount of transient data. These baseline approaches were used for comparing against other methods.

In contrast to prediction-based methods, feature-based methods try to directly extract features that are small during normal operations and larger during anomalies. This effort developed one feature-based approach that uses principal component analysis (PCA) to calculate the dominant features (i.e., those that are constantly varying and consequently would not make good anomaly detection features) for both full power and transient data separately, combines them to extract the dominant system features, and then finds the null space features (i.e., those that should be small during normal operations) to be used for anomaly detection. This method is called the combined null space approach. Like the prediction-based methods, similar baseline approaches were implemented for comparison.

To evaluate methods in a controlled environment, synthetic data generators were created and used. These data generators were based on spring mass damper (SMD) systems commonly found in mechanical engineering references. The full and transient power conditions were translated to the SMD simulator by having ample data for an SMD system with one mass held fixed, but the rest of the masses can move freely (called the base operating mode), and limited data for the same system but with all masses allowed to move freely (called the transient operating mode), respectively. The initial methods exploration showed that the methods were extremely sensitive to nonlinearity. To make this assessment broader, the nonlinearity of the data was quantified, and both linear and nonlinear versions of the methods were tested on data of varying nonlinearity.

Starting with linear datasets and linear anomaly detection methods, the methods were implemented on the SMD datasets, and the results showed a significant difference in anomaly detection performance between the developed methods and baseline methods. For particularly data-sparse applications, any of the covariate shift, multiple models, or PCA based methods (baseline or combined null space) provided a strong and comparable performance with very limited transient data.

By contrast, nonlinear methods were not well suited to the nonlinear transient problem without significant amounts of data; however, the linear methods applied to the nonlinear datasets showed some success. This implies that, even though the overall dynamics of the SMD datasets are nonlinear, there must be some linear patterns within the data that the methods are recognizing and learning. These patterns still hold when transferring from base to transient operating data. In addition, it appears the feature-based methods performed better than the prediction-based methods on these datasets when given very small amounts of transient data, although this advantage was not observed as more data were added. One possible explanation for this is that the methods are finding just the features that are linear and ignoring the other effects, while the prediction-based methods may not be able to extract just the linear features as accurately.

Combining all of this, it appears that, for linear datasets, the transient problem is solvable and multiple methods can achieve good results. For nonlinear datasets, the transient problem is much more difficult and, for very limited transient datasets, may only be solvable when some linear patterns exist that can be extracted.

ACKNOWLEDGEMENTS

The authors wish to thank the Light Water Reactor Sustainability program for funding this effort. They also wish to thank Florida Power and Light Company, which is part of NextEra Energy Inc., for collaborating on this effort.

CONTENTS

FIGURES

TABLES

ACRONYMS

ANN	Artificial neural network
AUC	Area under the curve
DPCA	Dynamic principal component analysis
FN	False negative
FP	False positive
KDE	Kernel density estimation
LOVO	Leave one variable out
ML	Machine learning
N	Negative
NPP	Nuclear power plant
P	Positive
PCA	Principal component analysis
PN	Predicted negative
PP	Predicted positive
PR-AUC	Precision-recall area under the curve
ReLU	Rectified linear unit
SMD	Spring mass damper
TN	True negative
TP	True positive

EXTENDING DATA-DRIVEN ANOMALY DETECTION METHODS TO TRANSIENT POWER CONDITIONS IN NUCLEAR POWER PLANTS

1. INTRODUCTION

The current approach for detecting and responding to process anomalies in nuclear power plants (NPPs) is primarily reactive in nature. This means plant operators do not search within time-series process data for subtle signs of anomalies but wait until alarms are generated by the anomalies once they become significant enough to exceed some predefined threshold. However, a proactive approach using automated anomaly detection tools could detect subtle signs of anomalies before they escalate into unexpected equipment failures, thereby affording the plant additional lead time in which to act (Figure 1). This would introduce significant cost savings to the plant.

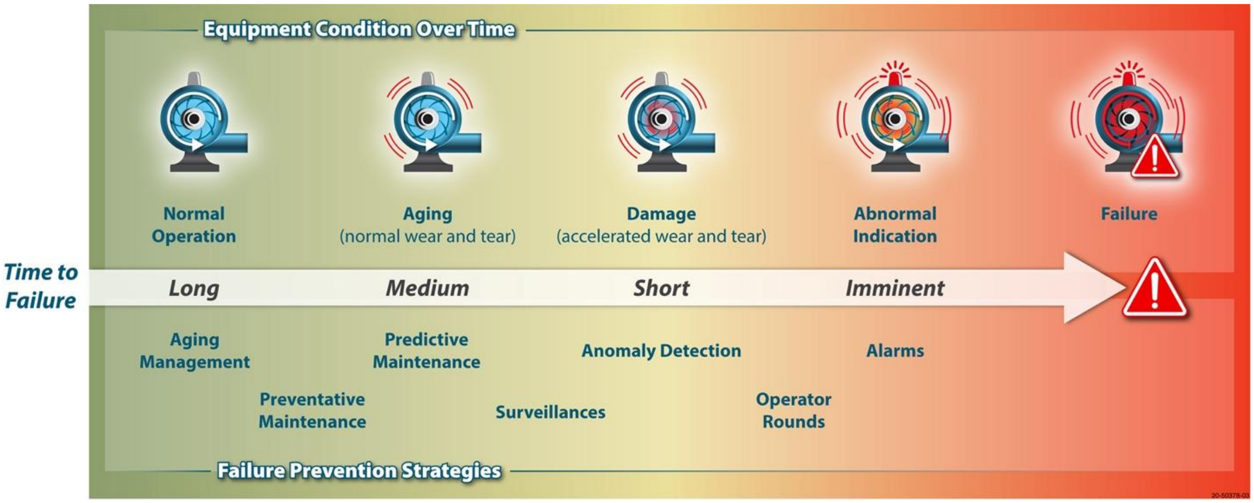

Figure 1. Equipment condition stages and strategies to prevent equipment failure.

To aid the nuclear power industry, the U.S. Department of Energy Light Water Reactor Sustainability program has been investigating machine learning (ML) methods for automated anomaly detection based on time-series data. This has included studies conducted on NPP test cases [1]; studies comparing and outlining when to use empirical, data-driven, and hybrid models [2]; studies investigating the incorporation of sparsely labeled known anomalous events into the anomaly detection methods [3]; and studies on methods to identify the root causes of anomalies [4]. Many of these studies have focused on using data-driven and ML methods due to their analytical power, scalability, and lack of required modeling investment. Those studies resulted in methods that are part of a multistage approach to detect anomalies with minimal false positives (Figure 2).

Historically, NPPs have operated predominantly at or near full power. As a result, existing archived data that could be used for training ML models will contain predominantly full power data, meaning that data-driven anomaly detection methods can likely perform well at full power operations. This presents a challenge when the power drops (referred to as a transient) and may result in false alarms due to the lack of historical data at those new power levels. The current approach to handling this challenge is to turn the anomaly detection algorithms off during transients when the power falls below some threshold. This would prevent false alarms during transients, but this also makes it impossible to use the algorithms to detect anomalies during these periods.

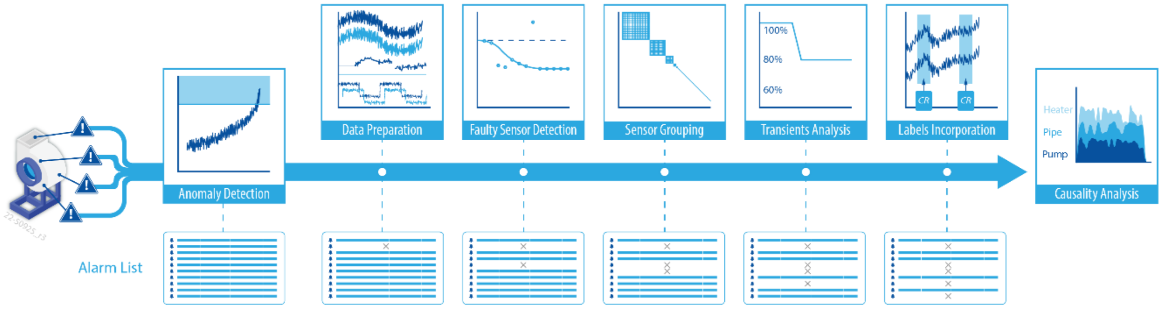

Figure 2. Methods developed to enhance anomaly detection while minimizing false alarms.

Because power plants could operate at reduced power levels for extended periods of time, the objective of this effort is to develop data-driven anomaly detection methods that can extend to transient conditions. Specifically, the research hypothesis tested here is that anomaly detection methods can be modified and used during the data-poor transient conditions compared with baseline methods used in normal operation conditions. In the context of data-driven approaches, this means that the methods are trained on a mix of predominantly full power data and some sparse transient power data (collectively called the training data), and tested on exclusively transient power data (called the testing data). While the objective here focuses on full and transient power conditions, the problem can be viewed more broadly as any situation where there are ample data for some condition but limited data for another similar condition.

Within anomaly detection, this study focuses on two common types of approaches: prediction based and feature based.

1.1 Prediction-Based Methods

Prediction-based methods often rely on self-supervised learning, where a subset of the data are used to predict another part, enabling learning to fill in the gaps. In this effort, this is performed by either withholding some data from the complete dataset and predicting that withheld data or compressing the complete data to some smaller dimension and using the compressed data to predict the full data. Detection is then based on the prediction error between the real measurement and prediction, which should be small during normal operations and larger during anomalies. A common example of a prediction-based anomaly detection algorithm involves using an autoencoder, in which the original data is compressed to some reduced dimension (often called a latent space) and then reconstructed. During training, the compression and reconstruction models learn the underlying patterns that correlate the sensor measurements during normal conditions. When those underlying patterns break, the reconstruction error will be large, and an anomaly is declared.

This effort implemented three prediction-based approaches to extend prediction-based anomaly detection methods to account for the limited transient data. The first approach is to treat it as a covariate shift problem (or sometimes referred to as imbalanced regression), described in Section 4.2.2. In the covariate shift problem, there is a shift in the data distribution from the training data to the testing data distributions (Figure 3). Approaches to address the covariate shift problem focus on synthetically altering the training data distribution. These approaches include tools to weight the samples when training models [5, 6] and tools to resample the available data [7, 8], both of which aim to approximate a distribution closer to the desired distribution. In the context of this effort, this would result in weighting transient training data higher than full power training data, or resampling to add additional transient training data instances. One advantage to these approaches is that they can be combined with previously developed anomaly detection algorithms, because they focus on data sampling, taking advantage of past work.

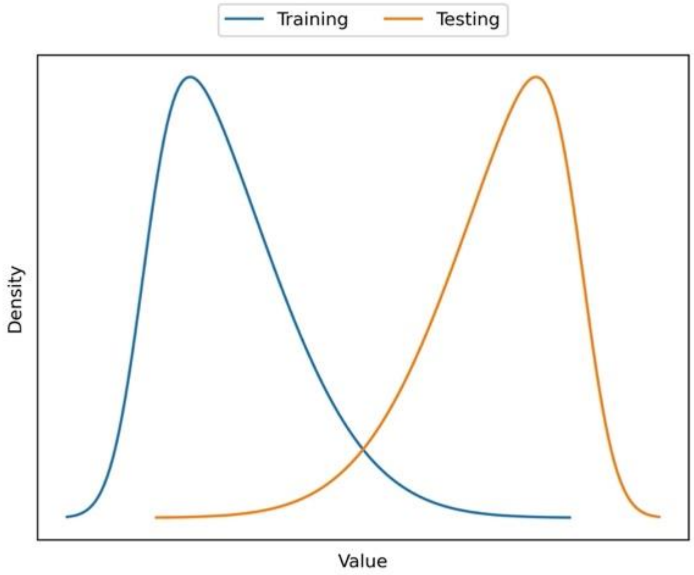

Figure 3. Illustration of the covariate shift problem, where there is a difference between the training and testing data distributions.

Second, a new approach called the multiple models approach was developed that calculates two isolated prediction models—one for all correlations except power and one for correlations just from power—and combines them (Section 4.2.3). This is an example of a transfer learning approach, which separates the training data into an abundant source dataset (full power data) and a sparse target dataset (transient power data). Assuming the source and target datasets share some features or properties, the idea is to transfer knowledge learned from the source dataset to be used with the target dataset. The covariate shift and multiple models approaches are agnostic to what type of prediction model is used, so this effort used the leave-one-variable-out (LOVO) model (Section 2.2 for the linear version and Section 2.3 for the nonlinear version) [4].

The third approach was the frozen layers approach using autoencoders, which is another example of transfer learning (Section 4.2.4). In this approach, some of the model weights trained using the source dataset are fixed (frozen), and the target dataset is then used to fine-tune the rest of the weights [9]. In this way, the autoencoder can learn features from the source data and then fine-tune them to the dynamics of the target data. This approach used the autoencoder prediction model (Section 2.4).

In addition to the extension methods, this effort implemented baseline approaches to compare the extension methods to (Section 4.2.1). The baseline approaches use prediction-based models without implementing extensions for the transient problem.

This review also considered another approach that treated the problem as a time-varying regression problem. In other words, the transients are simply the system dynamics naturally changing over time, and this process can be captured using time-varying anomaly detection models. This problem has been addressed using a windowed approach that trains models online. This means a past window of data is used to train a model to predict the next window of data, and these windows slide through time as more data is collected (Figure 4) [10, 11]. Like the covariate shift problem, an advantage to this approach is that it can build on or even directly use previously developed anomaly detection methods. However, it also has some potential limitations. First, it does not consider the entire set of available data, which could result in additional spurious false alarms from a lack of training data. Second, it is constantly updating the training data, so it would likely only detect anomalies that result in abrupt changes in process variables.

For slowly changing anomalies, the anomalous signal may be learned as normal before it becomes large enough to flag as an anomaly. Due to these limitations, the time-varying approach was not implemented.

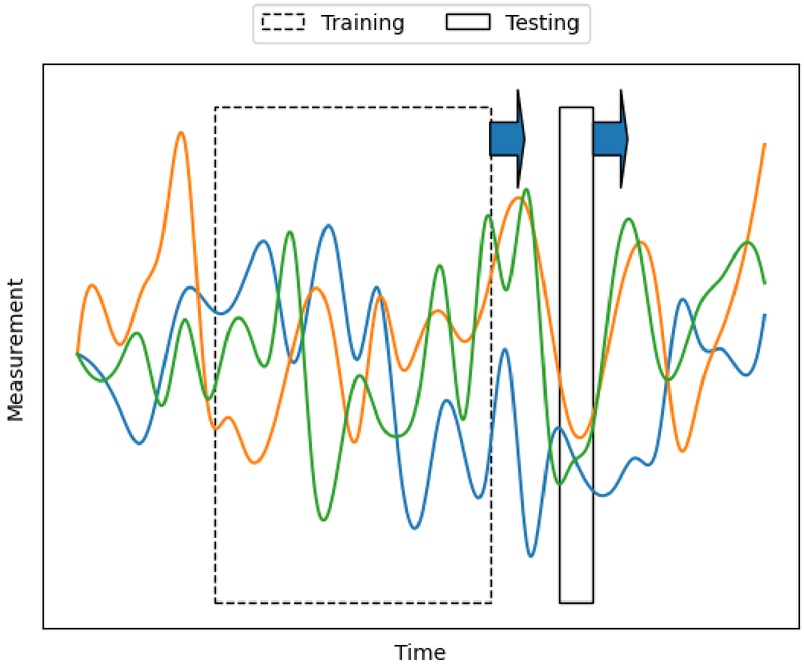

Figure 4. Illustration of treating the problem as a time-varying problem, where there are windows used to train a model and test new data that slide forward through time (the blue arrows).

1.2 Feature-Based Methods

In contrast to prediction-based methods, feature-based methods try to directly extract features that are small during normal operations and larger during anomalies. An example is using principal component analysis (PCA) (Section 2.1), which breaks data into orthogonal directions ordered by how much variance in the data each direction explains. The components that explain the least variance are often described primarily by noise and will be small during normal operations and large during anomalous operations [12].

Feature-based methods in the literature for extending to the transient problem often take a transfer learning approach. One method has used the more abundant source data to train an encoder that maps the sensor measurements into a feature space [9]. Then to transfer the knowledge, that same encoder is transferred to the target data, where the resulting source data features are used to detect anomalies. One challenge to using this approach for sensor data is that it assumes that the features will carry over from the source dataset to the target dataset. Prior work using this idea has often focused on image data, where features can represent general patterns present in many kinds of images.

This effort implemented one feature-based approach to extend feature-based anomaly detection methods to solve the transient problem. Due to the drawbacks of directly transferring features from source to target data, this effort has instead developed and implemented a new feature-based approach. This new approach, called the combined null space approach, calculates the dominant features (i.e., those that are constantly varying and consequently would not make good anomaly detection features) for both full power and transient data separately, combines them to extract the dominant system features, and then finds the null space features (i.e., those that should be small during normal operations, but can be used to detect anomalies) for anomaly detection (Section 4.3.2). These approaches make use of the PCA algorithm as detailed in Section 2.1.

Like the prediction-based methods, this effort also implemented baseline feature-based methods, which are similar in concept to the prediction-based baseline approach (Section 4.3.1).

To evaluate methods in a controlled environment, this effort created and used synthetic data generators (Section 3). These data generators were based on spring mass damper (SMD) systems commonly found in mechanical engineering references. The full and transient power conditions were translated to the SMD simulator by having ample data for an SMD system with one mass held fixed, but the rest of the masses can move freely (the “full power” condition, here called the base operating mode), and limited data for the same system but with all masses allowed to move freely (the “transient” condition, here called the transient operating mode). The results from the methods applied to these datasets can be found in the respective section for each method covered in this report. During the initial methods exploration, the methods were extremely sensitive to nonlinearity. To make this assessment broader, the nonlinearity of the data was quantified (Section 3.1), and the methods were tested on data of varying nonlinearity (Section 4).

2. ANOMALY DETECTION METHODS

This effort makes a distinction between general anomaly detection methods and those that focus on the transient problem. This section discusses the general methods and how they are used for anomaly detection. They are separated to emphasize the changes made to address the transient problem. In addition, several of the transient-specific methods are agnostic to the anomaly detection method used.

2.1 Principal Component Analysis

PCA is a popular algorithm that can separate data into orthogonal directions [13]. One of its most common uses is for dimensionality reduction, where it attempts to compress high-dimensional data while incurring minimal information loss. Traditional PCA, however, is only efficient when process variables are nearly time independent. To capture the autocorrelations that occur through time, the traditional PCA algorithm was modified to create dynamic PCA (DPCA) [14]. In anomaly detection methods, PCA and DPCA models are used to extract anomaly detection features.

This effort uses a DPCA model to capture the correlations between variables. To create this model, time-series data (vectors) are stacked to form a data matrix. Next, singular value decomposition is used to decompose the data matrix into features [13]. Note that this decomposition is a linear process, so it can only extract linear patterns from data. While the decomposition process does not necessarily imply the linear patterns are statistically independent, they can be treated as independent for many practical applications. Therefore, the decomposition permits one to view each pattern individually, ranked by their significance in the data.

Time-series data can be modeled as a sequence of measurement vectors $x_t \in R^m$, each consisting of m sensor measurements at sample times $t \in \{1,2,\ldots,n\}$, where n is the number of time steps in the data. In traditional PCA, the data samples would be stacked to create a large data matrix:

$$X = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix},$$

where $X \in R^{n\times m}$ (i.e., X is an $n \times m$ matrix of real numbers), and ^{T} denotes the transpose operator. Using this matrix, each data point can only capture correlations between variables in the same sample time. However, in dynamic processes, measurement samples may have an autocorrelation (i.e., dependence on information contained in previous sample times). Mathematically, this could mean an autoregressive system with $x_{t+1} = A_0x_t + A_1x_{t-1} + \cdots + A_px_{t-p}$ for a p-order dynamic system, where the A terms are coefficient matrices.

To capture this autocorrelation, DPCA uses augmented data sample $\mathrm{z_t} = [x_{t-s+1}^T \quad ... \quad x_t^T]^T$, each representing a window of data that includes both the measurement vector for that sample time and the measurement vectors for a fixed number of previous sample times. These samples are then stacked to form an augmented data matrix:

$$Z = \begin{bmatrix} x_1^T & \cdots & x_s^T \\ \vdots & \ddots & \vdots \\ x_{n-s+1}^T & \cdots & x_n^T \end{bmatrix} = \begin{bmatrix} z_s^T \\ \vdots \\ z_n^T \end{bmatrix},$$

where $Z \in R^{n-s+1\times ms}$ is the augmented data matrix, s is the window size of the augmented data sample, the rows contain the augmented data samples each containing data for a given window, and the columns contain the data for each variable at s different time steps. By incorporating multiple time steps within a window, the augmented data matrix can seek patterns both within the same time step and across neighboring time steps.

Once the augmented data matrix is created, it can be used with the standard PCA algorithm. This means it can be decomposed as a product of three matrices, namely the orthonormal matrix $U = [u_1 \, u_2 \dots u_{ms}]$, a positive semidefinite diagonal matrix S with elements $\{\sigma_1, \sigma_2, \dots, \sigma_{ms}\}$ along the main diagonal, and an orthonormal right singular matrix $V^T = [v_1 \, v_2 \dots v_{ms}]^T$:

$$Z = USV^T = \sum_{i=1}^{ms} u_i \sigma_i v_i^T,$$

where the left singular vectors are denoted by $u_i \in R^{ms}$, the right singular vectors by $v_i \in R^{n-s+1}$, and the singular values by $\sigma_i \in R$.

This decomposition can be used to generate a compressed approximation of the input data that captures most of the information (in the Frobenius norm-sense $|| \cdot ||_F$). Since the vectors are ordered by magnitude of the singular values (which represent their respective explained variances), a rank $k < ms$ approximation of the input data $\tilde{Z} \in R^{n-s+1 \times k}$ can be calculated. This approximation is obtained by projecting the data matrix on the first k left singular vectors (where k is often called the latent size):

$$\tilde{Z} = \sum_{i=1}^{k} u_i \sigma_i v_i^T.$$

Building a rank $k < ms$ DPCA model that captures most of the feature variance at a given time step is given by the first k left singular vectors $\{u_i\}_{i=1}^{k}$. Intuitively, this yields the top k patterns in the data that explain most of the trends and correlations observed between the measurement samples across time. The discarded left singular vectors $\{u_i\}_{i=k+1}^{ms}$ are considered uninformative and are typically assumed to be statistical noise in the time series.

A key observation in the present work is that, while the discarded left singular vectors $\{u_i\}_{i=k+1}^{ms}$ are uninformative when the samples are closely approximated by the DPCA model, this may not necessarily be the case when a sample is out of distribution (i.e., it cannot be approximated by the model). In other words, an error-like term can be calculated using the discarded feature(s) and consolidated into a single metric using their Euclidian norm (or another appropriate norm):

$$e_t = \sum_{i=k+1}^{ms} \left| u_i^T z_t \right|^2.$$

Nonlinear problems may require kernel PCA where the data are projected to a higher-dimensional space using nonlinear functions (also called kernels). In this space, the nonlinearly projected patterns are assumed to be linearly separable to enable the above analysis.

2.2 Linear Leave-One-Variable-Out Model

One challenge in implementing the DPCA method is the selection of the latent size. An alternative to the DPCA model is the LOVO model (developed as part of a prior effort [4]), which predicts each variable using regression, with all other variables serving as model inputs. This method does not compress the data, rendering the question of latent size moot.

As in the previous method, the augmented data samples z_t are used. However, rather than concatenating all the data into a single matrix, they are separated into input and output matrices. This is done for each variable, in rotation. Then a traditional regression approach can be taken, using the input matrices to predict the output matrices. As the method's name suggests, a model is created that leaves out one variable from the input data, then uses regression to predict that left-out variable. The input data include all the z_t-captured time steps in the window. For the output data, only the center sample time is used (meaning the window size for this method should be odd). This approach of only using the center point was selected because some variables cause future changes in other variables. For example, in an NPP, turning on a heater will eventually affect the temperature but will spark almost no immediate change. As such, if the model is trying to predict the heater power, it would likely be more accurate if it had access to both past and future temperature values within the window.

The input and output matrices are defined for each variable i in the vector x_t. The matrix definitions are simplified using the notations $x_{i,t}$ and $x_{-i,t}$, which represent variable i of x_t and all variables in x_t except for variable i, respectively. Note that when used with $z_{-i,t}$, this means that all instances (in the window) of variable i are removed. Then the input and output matrices for variable i are defined as:

$$X_i = \begin{bmatrix} x_{-i,1}^T & \cdots & x_{-i,s}^T \\ \vdots & \ddots & \vdots \\ x_{-i,n-s+1}^T & \cdots & x_{-i,n}^T \end{bmatrix} = \begin{bmatrix} z_{-i,s}^T \\ \vdots \\ z_{-i,n}^T \end{bmatrix},$$

$$Y_i = \begin{bmatrix} z_{i,s}^T \\ \vdots \\ z_{i,n}^T \end{bmatrix} = \begin{bmatrix} z_s^T \\ \vdots \\ z_n^T \end{bmatrix} J_i^T,$$

where X_i is the input matrix, Y_i is the output matrix, $J = [0 \quad I \quad 0]$, where the 0 terms in J are zero matrices with a number of columns equal to $\frac{s-1}{2}$, and J_i is row i of J. In the matrix J_i, the I in the center is used to select just the center sample time in the window, and row i is used to select only variable i of that center sample time. Using these matrices, each row of X_i is the input information used to predict each row of Y_i.

Once the input and output matrices are created, they can be used to train a linear regression model for each variable i. The regression model is of the following form:

$$J_i\hat{z}_t = A_i z_{-i,t} + B_i,$$

where $A_i \in R^{1\times(m-1)s}$ and $B_i \in R$. The unknown coefficients were solved for by minimizing an objective function that included both the mean squared prediction error and elastic net regularization. Elastic net regularization places a penalty on large coefficients to prevent overfitting the model to the data [15].

Once all the individual variable models are solved for, they can be combined into a single LOVO model. Using matrix algebra, all these models for individual variables can be combined into a single model:

$$J\hat{z}_t = Az_t + B,$$

where $A = \begin{bmatrix} 0 & \cdots & A_1 \\ \vdots & \ddots & \vdots \\ A_m & \cdots & 0 \end{bmatrix} \in R^{m\times ms}$ is formed by stacking the A_i terms, with s zeros padded per row to ensure that variable i is not included in that row, and $B = \begin{bmatrix} B_1 \\ \vdots \\ B_m \end{bmatrix} \in R^{m\times 1}$ is formed by stacking the B_i terms.

For anomaly detection, the system models can be used to generate an anomaly score, which is a scalar value that quantifies the degree to which the augmented data are abnormal. When the score exceeds a certain threshold, the sample is considered anomalous.

To calculate the anomaly score, the prediction error e_t (often called residual) is first calculated as the difference between the measured and estimated values:

$$e_t = z_t - \hat{z}_t.$$

Then the score ϕ_t is calculated as the weighted sum of the squared error:

$$\phi(z_t) = e_t^T \hat{\Sigma}^{-1} e_t,$$

where $\hat{\Sigma}$ is a diagonal matrix containing the estimated variances of the error vector that normalizes the error statistics.

2.3 Nonlinear Leave-One-Variable-Out Model

In the linear LOVO approach, the data were split into an input and output dataset for each variable i to train m linear regression models. The individual models were then combined into a single model. This idea of training multiple models and combining them worked for linear models because they have low computational overhead and are fast to train. However, this would be much more computationally intensive using m artificial neural networks (ANNs).

By utilizing the flexibility of ANN models, this step can be simplified to use a single model that accomplishes the same goal. As such, an ANN was created following the input/output structure:

$$J\hat{z}_t = f(z_t),$$

where $f(\cdot)$ is the ANN function. The general architecture used in this effort is shown in Figure 5. In this figure, the input data are shown as a three-dimensional array, with the first dimension (labeled “None” as a placeholder for the number of windows) being a derivative of the Python programming language used here to imply that one or more windows can be transformed by the function in a single function call. The second dimension (equal to the window size) and third dimension (equal to the number of variables) are used for implementation purposes but could have been flattened and combined. In going through the ANN, the data are first split into m branches and flattened, with each branch omitting information from variable i. Conceptually, each of these branches can be considered its own ANN with a nonlinear transformation $f_i(\cdot)$. Each branch contains a nonlinear layer (in this effort, a densely connected layer with a rectified linear unit [commonly called ReLU] activation, though other ANN components and types could have been used) and a linear densely connected layer to reduce the dimensionality. All the $f_i(\cdot)$ functions are concatenated together to create the full $J\hat{z}_t$. In practice, this strategy of treating it as one model makes it more efficient. Then, anomaly scores are calculated in an identical manner to the linear LOVO model.

2.4 Autoencoder Model

Autoencoders [16] are a generalized nonlinear version of PCA deployed using a neural network architecture consisting of two networks—an encoder $E(\cdot)$ and a decoder $D(\cdot)$. Given samples z_t, the encoder network encodes the data into a low-dimensional space through a linear “bottleneck” layer, yielding $\tilde{z}_t = E(z_t)$. Effectively, while PCA finds a low-dimensional linear transformation of the input data, autoencoders find a low-dimensional linear transformation of nonlinear projections of the data. The differentiating feature of autoencoders from kernel PCA is that the autoencoder kernel, characterizing the high-dimensional projection, is not specified by the user, but is rather learned using a neural network that is continually trained on examples input by the user.

The decoder network performs the reverse operations, by finding a nonlinear projection of the embedding and finding a linear transformation of the data given by $\hat{z}_t = D(\tilde{z}_t)$. When combined, the objective of the neural network is to tune its weights and find encoder and decoder transformations that minimize the squared error loss. For time-series data, the kernel and bottleneck dimension may be treated as hyperparameters to be tuned on a validation set not part of the neural network training until the validation loss is minimized. Similar to the LOVO prediction models, the error can then be used to detect anomalies.

The decoder can also be deployed as a generative model with an understanding of the statistical variations of the low-dimensional representation. This is exploited in variational autoencoders where additional constraints are imposed on the bottleneck layer, such as having a multivariate Gaussian distribution, which allows the end-user to generate Gaussian random numbers to generate a variety of samples using the trained decoder as a generative model. No such architectures are explored in this work. The overall autoencoder architecture, inclusive of the encoder and decoder, is depicted in Figure 6.

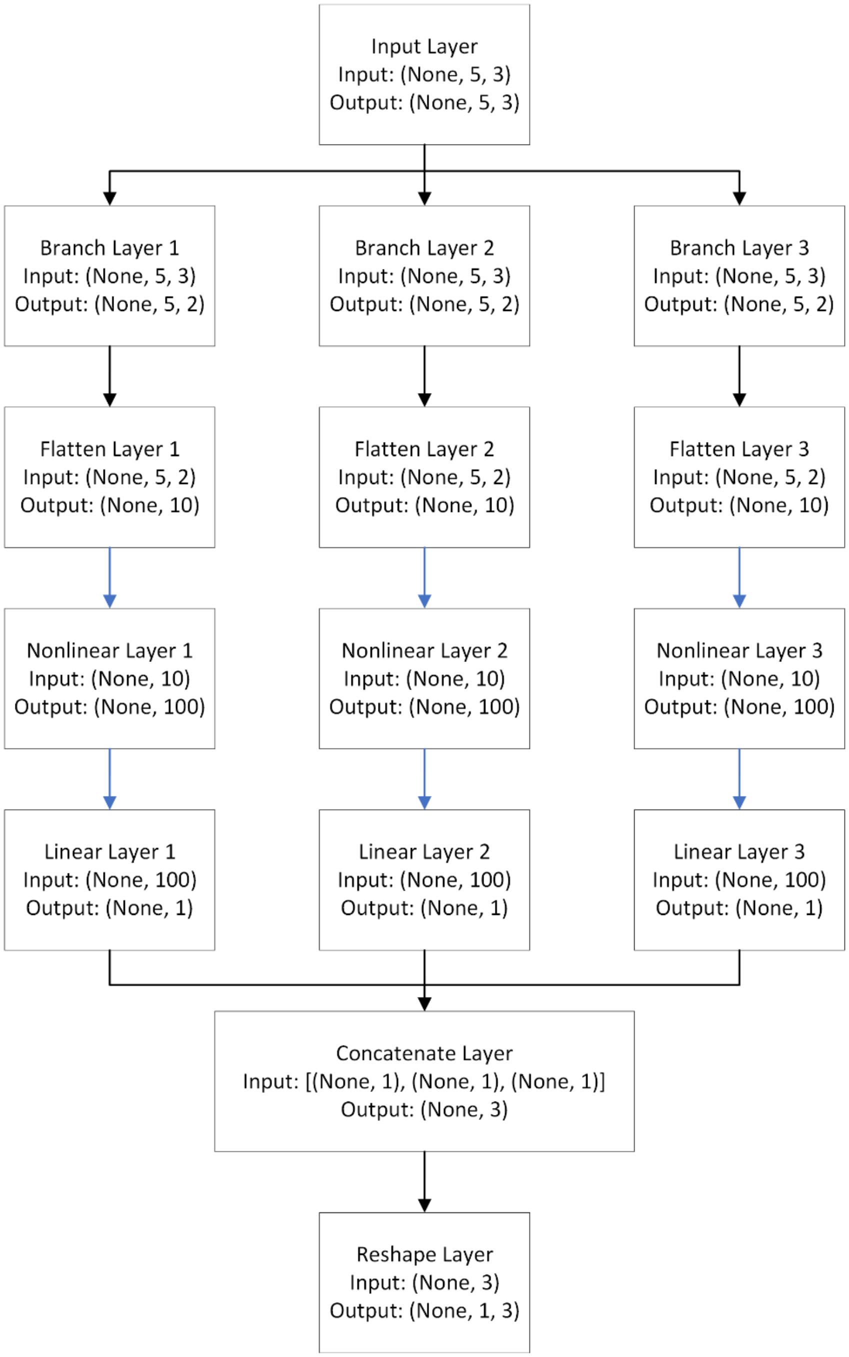

Figure 5. General ANN structure of the nonlinear LOVO model.

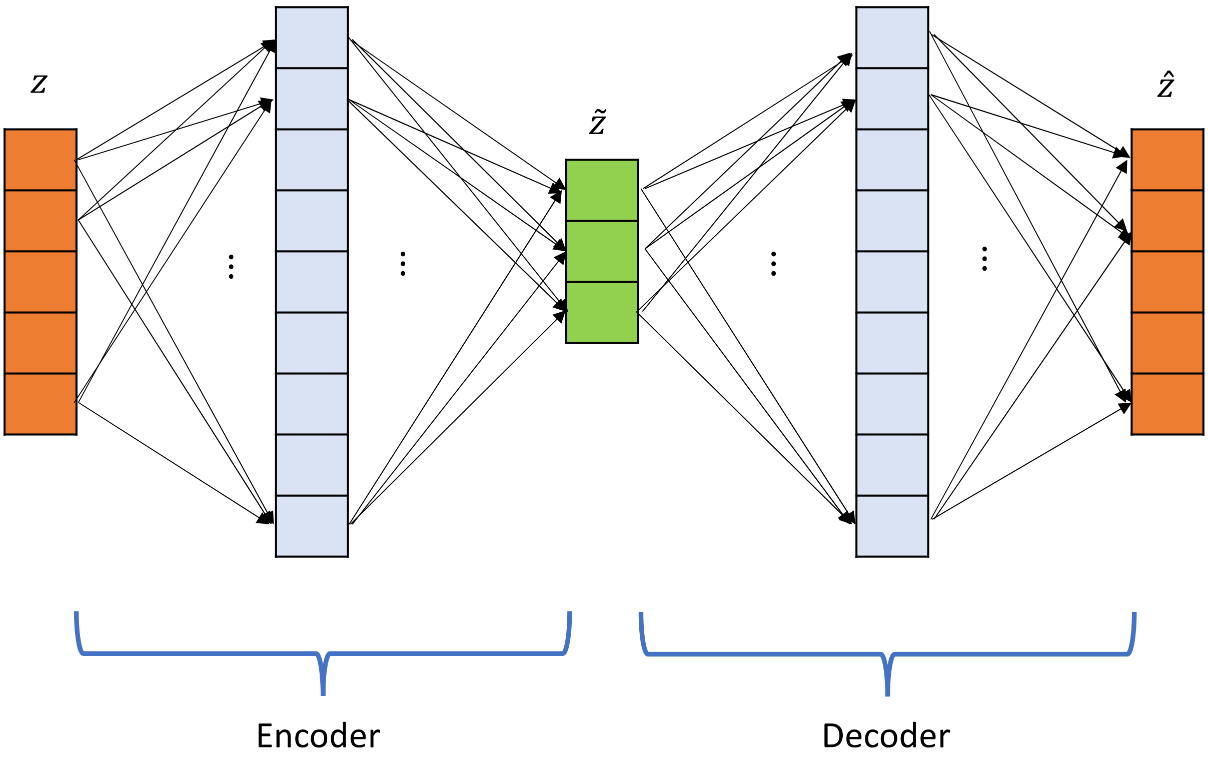

Figure 6. General ANN structure of the autoencoder model.

3. SYNTHETIC DATA

As previously mentioned, this work used synthetic data generators based on SMD systems. Due to the used method's sensitivity to nonlinearity, this was quantified.

3.1 Nonlinearity Quantification

This research anticipated that extending anomaly detection methods to transient conditions is more difficult for nonlinear systems than for linear systems. Therefore, this effort researched a nonlinearity measure for use with anomaly detection applications [17]. For the measure described in this section, the relative scales of the variables matter. Without loss of generality, it is assumed that all variables have been normalized to have zero mean and unit variance.

This effort started with a nonlinearity measure found in the literature that quantified nonlinearity in functions g of the form $y = g(x)$, where y and x can both be vector-valued functions. The general idea is that the nonlinearity measure should quantify the distance between $g(x)$ and its closest linear approximation L in the set of all linear functions $\mathcal{L}$ [17]. This amounts to finding the linear function L that reduces this measure:

$$M = \inf_{L \in \mathcal{L}} \sqrt{E[\|L(x) - g(x)\|_2^2]},$$

where inf is the infimum operator (similar to a minimization operator), E is the expected value, and $|| \cdot ||_2^2$ is the 2-norm squared. This measure can be normalized so that functions on different scales can be compared evenly. Their normalized nonlinearity measure is defined as:

$$\mathcal{M} = \frac{M}{\sqrt{E[\|g(x)\|_2^2]}}.$$

Intuitively, when $L(x)$ explains none of the data (i.e., when $L(x) = 0$), this normalized measure goes to one. When it fully explains the data (i.e., $L(x) = g(x)$), it goes to zero. As such, this measure falls between zero and one. Without the inf function, this could exceed one, as there are linear functions that would increase the numerator; however, the inf function prevents this from occurring.

Considering this measure further, it can identify two different phenomena as nonlinearities: actual nonlinear patterns between the data; and noise, which cannot be explained by a linear model. As such, there could be a true linear function $g(x)$ with sufficient noise that it provides a nonlinearity measure close to one. This is a reasonable result, as with so much noise, it is impossible to determine whether there could be small nonlinear functions superimposed within the noise.

To demonstrate the original method, the nonlinearity measure for some simple examples were calculated. The examples include the equations $y = x + \alpha x^2 + N$ (Figure 7), $y = x + \alpha x^3 + N$ (Figure 8), and $y = x + \alpha N$ (Figure 9), where x is a uniform random variable, N is Gaussian noise, and α is a multiplier that increases throughout the subplots.

For anomaly detection, there were two limitations with this nonlinearity measure as proposed. First, in anomaly detection, there is no inherent input/output data but rather simply a set of data that can be used to detect anomalies. To overcome this limitation, the LOVO framework is employed to calculate the linear model L, with $g(x)$ replaced with x:

$$\mathcal{M} = \sqrt{\frac{E[\|(A - I)x + B\|_2^2]}{E[\|x\|_2^2]}},$$

where A and B are calculated using the LOVO approach.

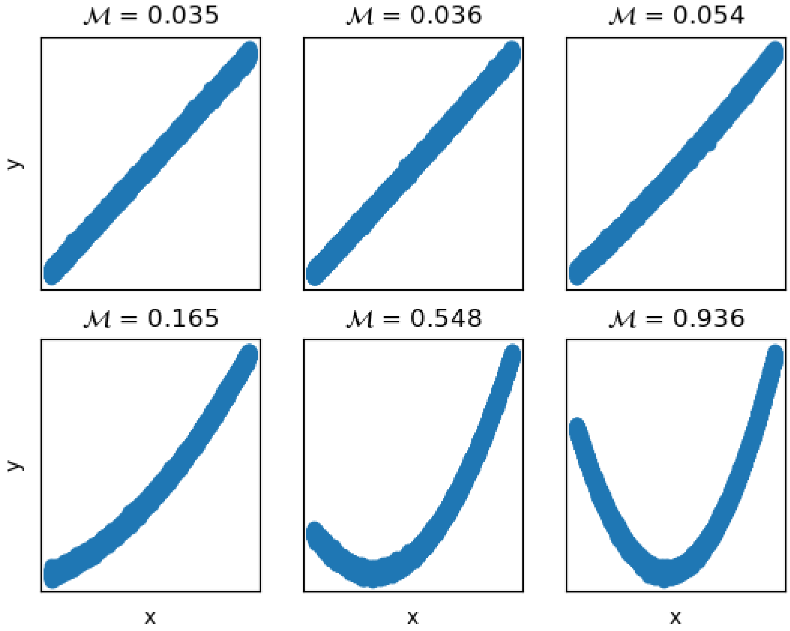

Figure 7. Nonlinearity measure for a second-order polynomial with increasing nonlinearity.

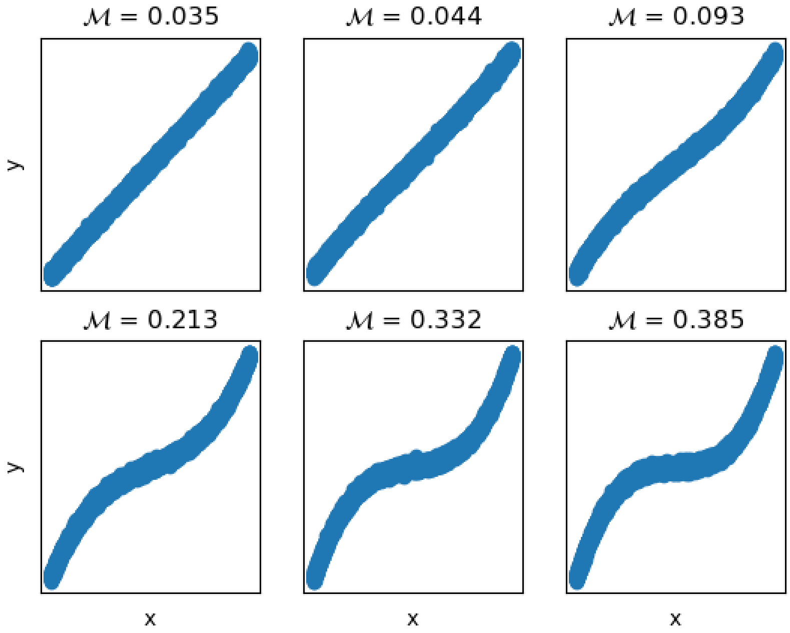

Figure 8. Nonlinearity measure for a third-order polynomial with increasing nonlinearity.

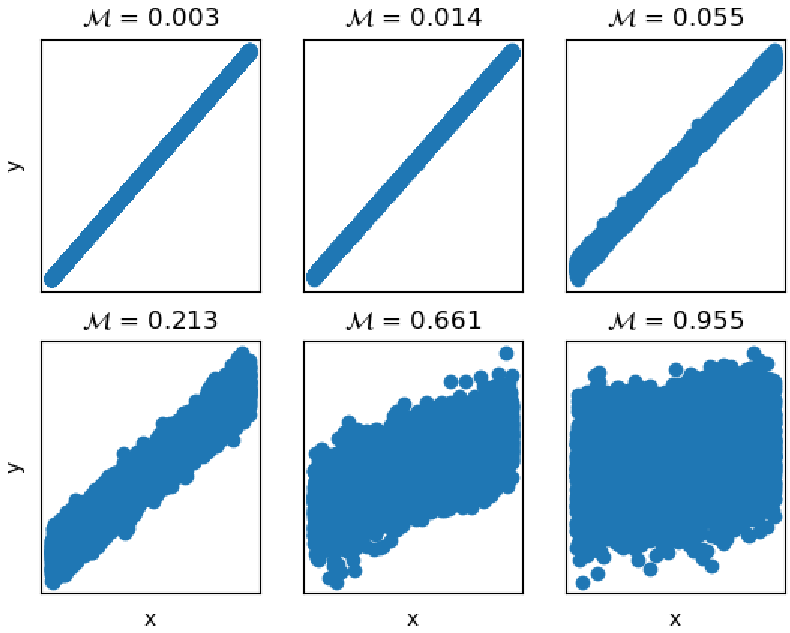

Figure 9. Nonlinearity measure for a first-order polynomial (line) with increasing noise.

The second limitation is that, in general, the generative function (i.e., $g(\cdot)$) is unknown and only sample data are known. In the original method [17], it was proposed that, for complicated nonlinearities, the measure could be calculated numerically. In this effort, this is the natural choice because there are data and not functions, such that the expected values can be replaced by sums:

$$\mathcal{M} = \sqrt{\frac{\sum_{i=1}^{n}\left((A-I)x_i + B\right)^T\left((A-I)x_i + B\right)}{\sum_{i=1}^{n} x_i^T x_i}}.$$

In summary, this metric can empirically measure nonlinearity for multivariate anomaly detection data with a result that ranges from zero to one. It is also worth noting that this can be done for each variable in x (or y if using the original formula), which can add insight.

An example is shown in Figure 10, where x_0 and x_1 are uniform random variables and $x_2 = x_0^2 + 10x_1$ and no noise is added. From these plots, x_1 and x_2 have some linearity (although are still not linear), and x_0 is highly nonlinear. As a result, the combined nonlinearity measure shows that this system of equations is highly nonlinear.

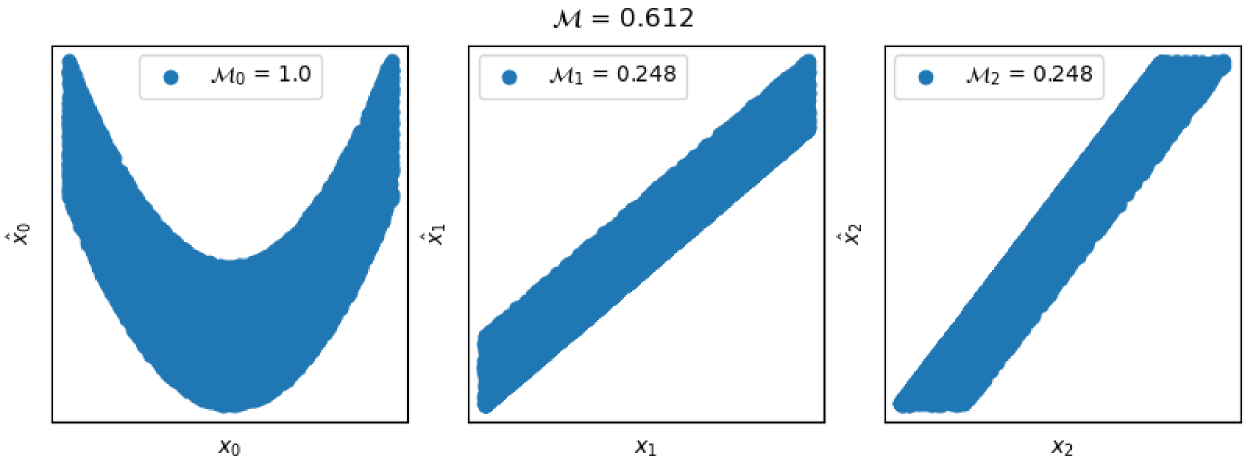

Figure 10. Nonlinearity measure in the LOVO framework.

3.2 Spring-Mass-Damper Data

The simulator used the idea of the SMD system, commonly seen in mechanical engineering references. The basic building blocks of this system are springs, masses, dampers, sensors, and actuators: the masses respond to forces, the springs apply restorative forces to the mass, the dampers apply damping forces to the mass, the sensors measure the position of the mass, and the actuators apply forces directly to the mass. This simulator was used in previous efforts, and additional details on equations of motion and differential equations are provided by Farber et al. [3]. Among its key features are its abilities to:

- Emulate both linear and nonlinear differential equations
- Simulate SMD components in different configurations, quantities, and layouts
- Incorporate process and measurement noise to make more realistic data
- Inject multiple types of anomalies.

The SMD system was selected for this research for several reasons. First, though it is conceptually simple and features just a few basic components, those components can be combined to generate high-order systems with coupled variables. Second, the system is easily scalable to include many sensors and actuators. These two characteristics are important for emulating the large-scale, high-order systems in NPPs. Third, the system allows for a straightforward incorporation of sensor anomalies (by directly modifying sensor measurements) and process anomalies (by modifying system parameters—here, the spring stiffness and damping coefficients).

The objective of this effort is to extend anomaly detection methods to transient conditions. This problem was translated to the SMD simulator as having ample data for an SMD system with one mass held fixed, but the rest of the masses can move freely (the “full power” condition, here called the base operating mode), and limited data for the same system but with all masses allowed to move freely (the “transient” condition, here called the transient operating mode).

The SMD configuration used in this effort (see Figure 11) had four masses (labeled with m) connected in series, with the two end masses also connected to fixed reference points (commonly called grounds). Position sensors (labeled with s) were placed on each mass, and actuators that applied forces (labeled with f) were placed on each mass except the last one. This last actuator was omitted to emulate only having one variable change when going from base to transient modes. However, while only one variable changes, the dynamics change, which consequently affects the entire system.

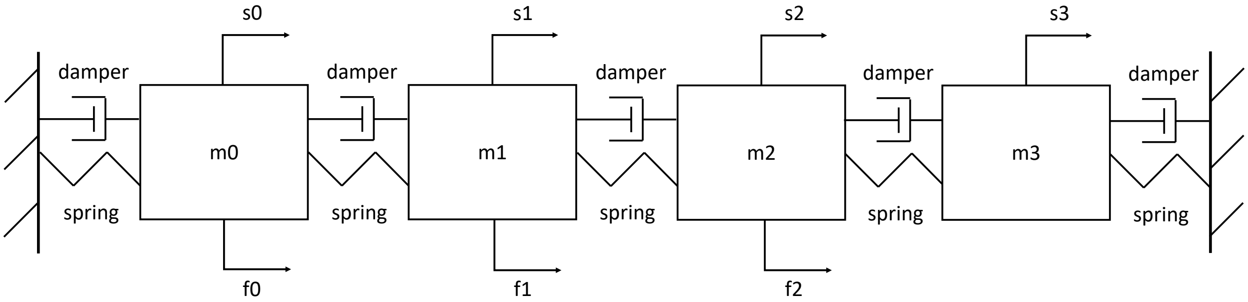

Figure 11. Sketch of the three-mass SMD system (without m1 grounded).

Within this configuration, both linear and nonlinear simulations were run. Here, linearity refers to the spring and damper components: the linear simulations used linear spring and damper components, meaning that their forces were linearly proportional to their relative displacements and velocities, respectively, and the nonlinear simulations used nonlinear spring components, meaning that their respective forces were a nonlinear function (third-order polynomial) of their relative displacements. For the nonlinear simulations, multiple values of the nonlinearity were assessed to study whether the "amount" of nonlinearity affected the results. The values of the nonlinearity measure used are shown in Table 1.

Table 1. Summary of the SMD datasets.

Number	Name	Nonlinearity $\mathcal{M}$
1	Linear	0.02
2	0.05 nonlinear	0.05
3	0.1 nonlinear	0.1

For each nonlinearity measure, five datasets were generated to capture the natural stochasticity of the SMD simulator. Each dataset represented 10 years' worth of the data, where the first year was base mode training data, the second year was transient mode training data, and the last 8 years were transient mode testing data. The methods were only tested on transient mode data as this was the focus of the effort. Examples of the base and transient mode data are shown in Figure 12. These two examples are both of normal (i.e., non-anomalous) operations, so their differences are not indicative of an anomaly.

When assessing the studied methods applied to these datasets, a natural question is how much transient data should be used in conjunction with the base mode data to train the anomaly detector. Rather than attempt to select a ratio or quantity, this effort turned this question into part of the experiment. As such, the detection algorithms were allowed to use all the base mode training data and an incremental amount of transient training data until the last detector trained was allowed to use all the transient training data. Results are shown as a function of this parameter.

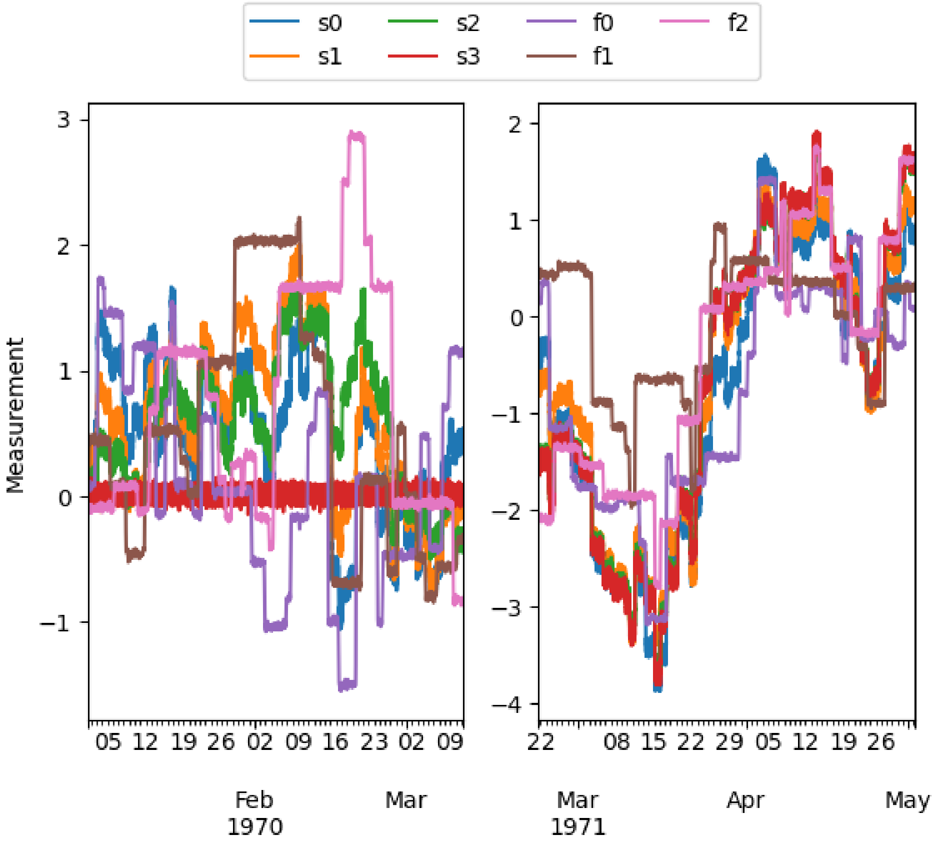

Figure 12. Example data from the base (left) and transient (right) operating conditions, where the primary difference is in whether the mass m3 is allowed to vary.

4. EXTENDING DETECTION METHODS TO TRANSIENTS

This section describes the developed prediction-based (Section 4.2) and feature-based (Section 4.3) methods. For each of these two categories, there are baseline and transient-specific methods. In addition, to report the results for the methods in a consistent and succinct way, this work used a common set of assessment metrics (Section 4.1).

4.1 Assessment Metrics

Assessment metrics are a number or set of numbers that describe the performance of some algorithm and can be used to compare methods. In this effort, the assessment metric should measure anomaly detection performance. Because there were multiple datasets, methods, and amounts of transient data used in the training data, it was beneficial to come up with a single number that summarized the detection performance for each of these experiments.

Many anomaly detection methods consist of two parts, a scoring algorithm that assigns an anomalous score to a data sample and a classifier that classifies the data as either normal or anomalous based on the score (e.g., scores above some threshold are classified as anomalous, whereas scores below are classified as normal). Because the scoring algorithm is inherent in the classifier, the classifier performance can be used to assess the detection methods. As such, this section uses the following conventions from the classification algorithms: positive (P), predicted positive (PP), true positive (TP), false positive (FP), negative (N), predicted negative (PN), true negative (TN), and false negative (FN). Here, a time-series sample is classified as positive if it is anomalous and negative if it is normal.

In selecting an assessment metric, it needed to be independent of two factors: the ratio of normal to anomalous data, and the threshold used to classify whether data are normal or anomalous. Starting with the first factor, a common classification assessment metric is accuracy, defined as (TP+TN)/(P+N). The challenge with this metric is that for example, if the data consist of 990 negative samples and 10 positive sample (i.e., 99% normal), a classifier that predicts all data as normal would give an accuracy of 99%. However, this classifier is obviously not useful when it is important to detect the anomalies within the data.

To overcome this first factor, this effort used precision and recall rather than accuracy. Precision and recall are defined as TP/(TP+FP) and TP/(TP+FN), respectively, and are commonly used to report metrics for imbalanced classification. Returning to the example above, the precision would be undefined, and the recall would be 0%, clearly showing that the classifier is not working well. The above example (Example 1) along with a second classifier (Example 2) that labels everything as positive are shown in Table 2.

Table 2. Accuracy, precision, and recall of two classifiers for an imbalanced dataset.

Example	PP	PN	TP	TN	FP	FN	Accuracy	Precision	Recall
1	0	1,000	0	990	0	10	0.99	Undefined	0
2	1,000	0	10	0	990	0	0.01	0.01	1

Moving to the second factor, using the examples from Table 2, Example 1 is equivalent to selecting a threshold above the maximum score (i.e., naively classifying everything as normal), and Example 2 is equivalent to selecting a threshold below the minimum score (i.e., naively classifying everything as anomalous). In both examples, the scoring algorithm is completely ignored by the classifiers. This shows that precision and recall are very sensitive to the method used to select the threshold. During actual detection problems, a threshold needs to be selected, but this is less important when comparing methods against each other.

To overcome this second factor, a metric called the area under the curve (AUC) was used. Parameterized by varying threshold values, precision and recall can be plotted against each other to show the tradeoff between them. Then, the AUC metric calculates the area under this curve, thereby removing the threshold from the calculation entirely. The ideal performance is precision and recall both equal to one. An example precision-recall curve is shown in Figure 13 with an AUC of 0.83. Based on these factors, the precision-recall AUC (PR-AUC) metric was used to compare different detection approaches.

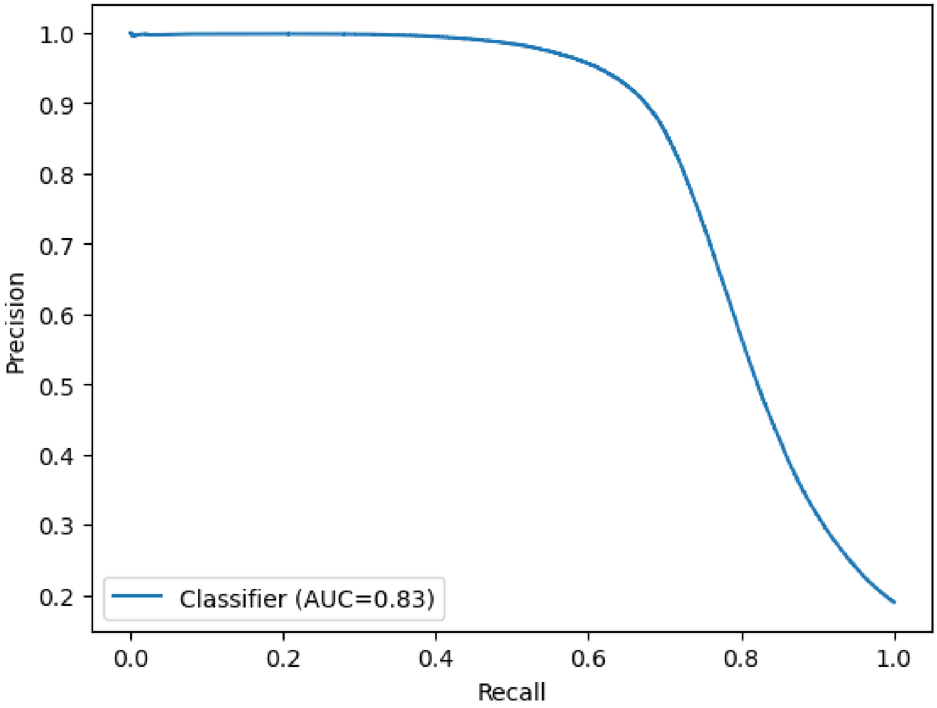

Figure 13. Example precision-recall curve with AUC calculated.

In this effort, the anomalies are inserted as ramp functions, meaning the effects of the anomalies (called anomaly magnitude) start at zero, and slowly increase to their maximum effect. As such, in the beginning of every anomaly, the magnitude is so small that the anomaly is impossible to detect. In other words, it is not expected that any detection algorithm will achieve both a precision and recall close to one. This approach was taken because it emulates anomalies of many different magnitudes, providing more opportunities to distinguish between the better and worse algorithms (assuming better algorithms will detect anomalies earlier in the ramp function).

4.2 Prediction-Based Methods

The prediction-based methods include the baseline methods (Section 4.2.1), the covariate shift method (Section 4.2.2), the multiple models method (Section 4.2.3), and the frozen layers method (Section 4.2.4). The baseline, covariate shift, and multiple models methods each used the LOVO prediction model (Section 2.2 and 2.3), and the frozen layers method used the autoencoder model (Section 2.4). The results of each of these methods applied to the SMD datasets are presented and compared in Section 4.2.5.

4.2.1 Baseline Methods

This effort implemented two baseline approaches, where baseline here implies that the prediction models were used without regard to the transient problem. In other words, these methods have access to the full power training data and part of the transient power data and simply train prediction models to be applied to the transient power testing data, using the reconstruction error to calculate anomaly scores and then AUC values. The first baseline approach used both the available full power and transient power data. This approach enabled an evaluation of how much the methods below impact detection performance. The second baseline approach used only the available transient power data and enabled an evaluation of whether the methods were able to transfer knowledge from the full power data to the transient problem.

4.2.2 Covariate Shift

As mentioned previously, the covariate shift problem assumes there is a shift in the data distribution from the training data to the testing data distribution. This assumption is valid, as the training data is predominantly full power data, but the testing data is exclusively transient power data.

The solution taken here is to synthetically alter the training distribution to get it closer to the desired distribution. This effort achieves this solution through weighting the data samples, often called importance weighting [5]. Changing the weights from the default uniform distribution effectively changes the training distribution.

Calculating importance weights needs two pieces of information. First, the training distribution is needed. When this distribution is unavailable, it can be estimated from the data using kernel density estimation (KDE) [18], which is a nonparametric technique for estimating probability density functions over a set of variables:

$$p(x) = \frac{1}{n}\sum_{i=1}^{n} K(x - x_i),$$

where $K(\cdot)$ is a kernel smoothing function that smooths the data distribution as a function of a user-selected bandwidth. An example of KDE applied to a data distribution with the Gaussian kernel and three different bandwidths is shown in Figure 14. In this example, the bandwidth of 0.01 is noisy, suggesting it is too small, the bandwidths of 1 and 3 reduce the peaks, suggesting they may be too large, and the bandwidth of 0.1 seems to capture the distribution but with less noise than the bandwidth of 0.01, suggesting it may be the most appropriate here. This process becomes more challenging in higher dimensions. In this effort, the Gaussian kernel smoothing function was used, and the bandwidth was selected using Scott's rule of thumb, which is a suggested bandwidth based on the size of the data, the number of variables, and the scale of the data [19].

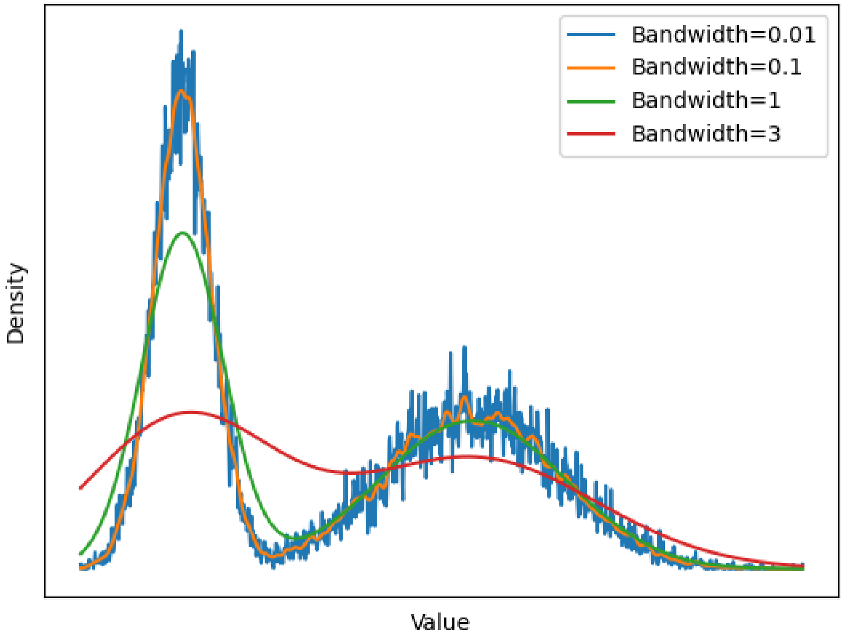

Figure 14. Illustration of KDE with three different bandwidths.

The second piece of information needed is the testing data distribution. In general, this distribution is unknown, and there are very limited data with which to produce this estimate. As such, this effort assumed that all power levels and variables are equally likely; in other words, the testing distribution is uniform. This ensures that the model does not bias any power level over another.

Given these two distributions, the importance weights can be calculated as [5]:

$$w(x_i) = \frac{p_{test}(x_i)}{p_{train}(x_i)},$$

where p_{train} and p_{test} are the training and testing distributions. This can be simplified further because p_{test} is a uniform distribution, which has constant likelihood, and because the weights are normalized within regression algorithms, so only relative values matter:

$$w(x_i) = \frac{1}{p_{train}(x_i)}.$$

The result of this procedure is a set of importance weights for the data that are applied to each sample error during training. An example from one of the SMD datasets is shown in Figure 15, where the top plot shows the sensor data, and the bottom plot shows the importance weights. In this example, the transient operating data make up just the last 1% of the time-series data but account for 5.4% of the total weight of the data, and each data sample in the transient data is weighted on average 5.8 times more than each data sample in the base operating data. Here, the weights are calculated for and applied to the entire measurement vector at each time stamp (as opposed to individual sensors).

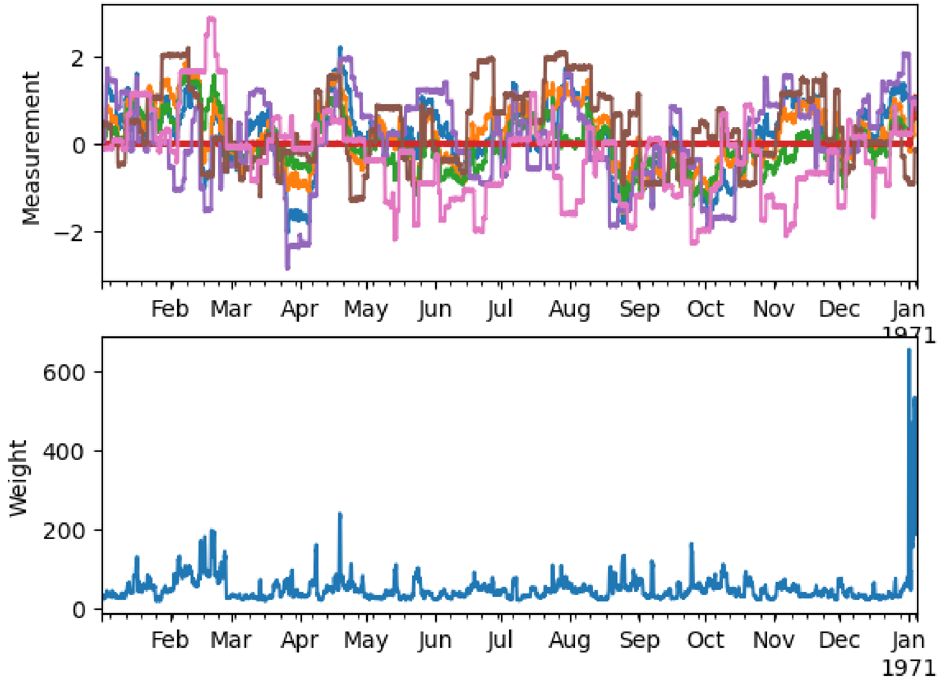

Figure 15. Example SMD data with corresponding sample importance weights.

4.2.3 Multiple Models

In this approach, the idea is to calculate separate contributions to an overall model from the full and transient power conditions. This is accomplished by using the full power data to train a corresponding model that extracts the full power contributions (i.e., the correlations of variables excluding the power) and using the transient power data to train a corresponding model that extracts the transient power contributions (i.e., the correlations just from power after removing the full power part). The final anomaly detection model is the combination of the two models. This approach will be described in the context of the LOVO framework; although it can be generalized to other prediction models. In the LOVO framework, models take the form $\hat{x} = f(x)$.

The data are separated into full power and transient power data X_F and X_T, respectively. Because the full power data contain no transients, they are equivalent to the full power contribution data $X_F^* = X_F$ (where * indicates contribution data) and are used to train the corresponding contribution model:

$$\hat{x}_F = M_F(x),$$

where M_F is the full power contribution model, and $\hat{x}_F$ is the full power contribution estimate.

The full power contribution model is then used to calculate the transient power contribution by subtracting the full power contribution estimate from the transient power data:

$$x_T = x - M_F(x),$$

where x_T is the transient power contribution. This transformation is only applied to transient power data. Applying this transformation to the transient power data results in the transient power contribution data $X_T^* = X_T - M_F(X_F)$. When this transformation is applied to full power data, the result is a zero-mean residual sequence (because full power data only contains full power contributions). Finally, the transient power contribution data are used (along with a modification, described below) to train the corresponding contribution model:

$$\hat{x}_T = M_T(x_T),$$

where M_T is the transient power full model. The required modification is that, because the full power contribution data has been mapped to a zero-mean process, the transient power contribution model must also map full power to zero. For linear models, this is done by setting the y-intercept equal to zero. Nonlinear models (including the nonlinear LOVO model) in general do not have an equivalent constraint. To get around this problem for nonlinear models, the origin can be added to transient power contribution data with a high sample weight to ensure it passes through (or close to) the origin.

With these two contribution models defined, the full model can be defined as the sum of the two estimates:

$$\hat{x} = \hat{x}_T + \hat{x}_F = M_T\big(x - M_F(x)\big) + M_F(x).$$

This entire process is applied to SMD data and is shown in Figure 16. In Figure 16 (a)–(d), the first year (1970) is base operating data, and the second year (1971) is transient operating data. In this Figure, the base operating contribution estimates (b) are roughly equal to the raw data (a) for the base operating data (i.e., left half), but very inaccurate for transient operating data (i.e., right half); and the transient operating contribution estimates are zero mean for the base operating data and nonzero for the transient operating data. The error plot (d) is the difference between the raw data (a) and the combination of base and transient contribution estimates (b and c).

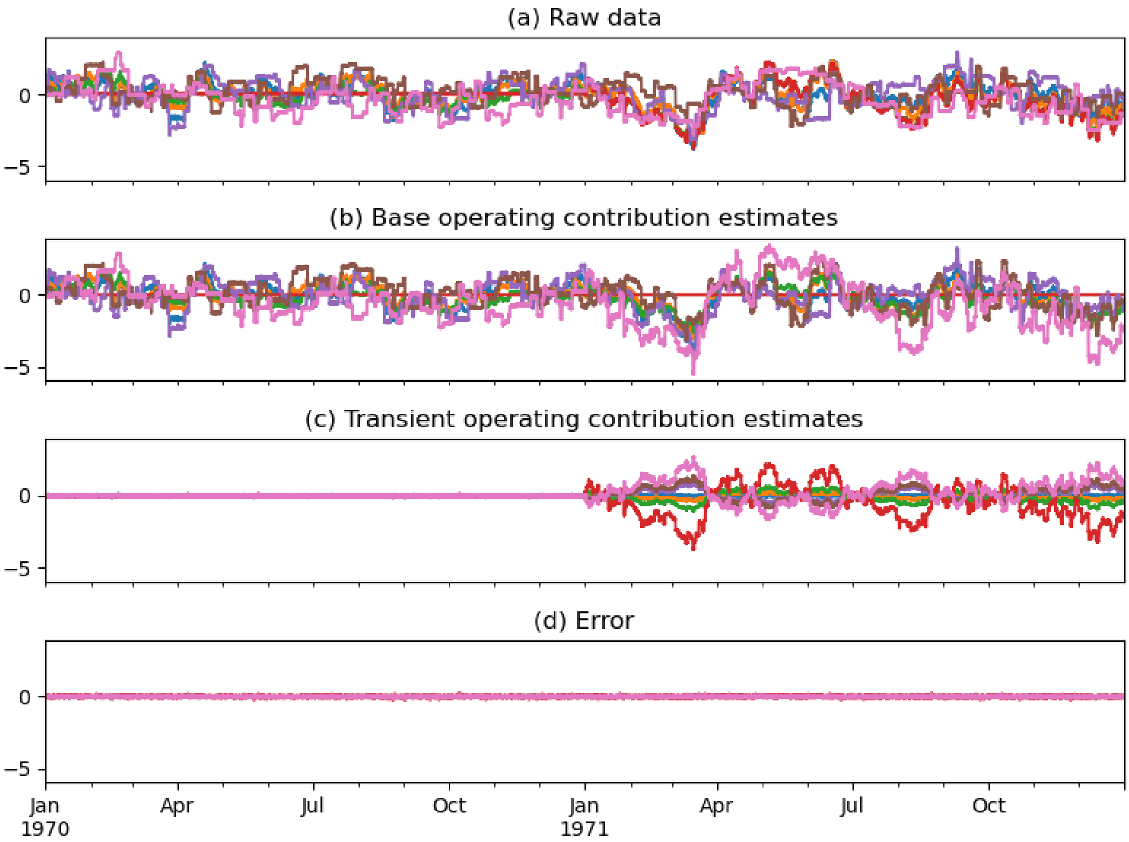

Figure 16. Example showing the data during the different steps of the multiple models approach.

4.2.4 Frozen Layers

Transfer learning with neural network architectures (including the autoencoder) is performed by training the network on the more abundant source dataset (here, full power data), freezing the weights of the initial layers of the neural network, and fine-tuning subsequent layers to the sparser target dataset (here, transient power data). This avoids having to train a neural network from scratch on the sparse target data only, while also providing it the flexibility to fine-tune the weights in the final few layers to the target dataset.

Expanding on this, the autoencoder is initially trained on the X_F data, after which the learning rates of the initial layer(s) consisting of the kernel projection and potentially subsequent layers are set to zero ("freezing"). The remaining weights are then fine-tuned to the X_T data. Since the initial high-dimensional projection is identical, identifying the features that "transfer" over may be observed using the weights of the neural network after the frozen layer(s). In this effort, just the encoder was frozen, as shown in Figure 17. This limited section was frozen because this represents the projection from the nonlinear input data to a higher-dimensional space on which the features become linear (i.e., the next transformation to the latent space is linear). As such, this seemed an appropriate place to freeze the network.

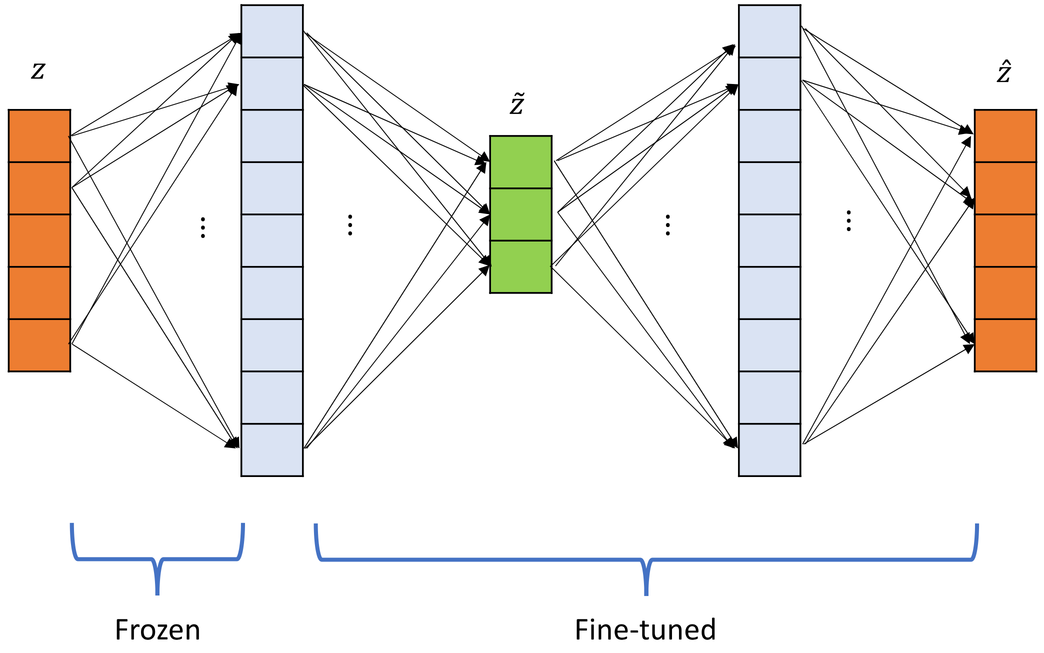

Figure 17. Diagram showing which layers are frozen and which are fine-tuned.

4.2.5 Results

This section shows results for the prediction-based methods applied to the different datasets shown in Table 1. Throughout this effort, there are different detection methods and extension methods; for clarity, the full set of methods and datasets analyzed is summarized in Table 3. In plotting results, the nonlinear LOVO methods are grouped with the autoencoder method because they are all nonlinear methods.

Table 3. Summary of datasets, detection methods, and extension methods analyzed for prediction methods.

Datasets	General Detection Method	Transient Extension Method
Linear datasets 0.05 nonlinear datasets 0.1 nonlinear datasets	Linear LOVO	Baseline: transient only Baseline: base and transient Covariate shift Multiple models
0.05 nonlinear datasets 0.1 nonlinear datasets	Nonlinear LOVO	Baseline: transient only Baseline: base and transient Covariate shift Multiple models
	Autoencoder	Frozen layers

For each dataset, the PR-AUC metric was calculated over a range of transient power training data added until the last detector trained was allowed to use all the transient power training data. Then, the median, first quartile, and third quartile (over the five datasets) are calculated to report the range of PR-AUC values.

For the linear dataset, the results for the two baseline, covariate shift, and multiple models methods are shown in Figure 18. Starting with the baseline: transient only case, the PR-AUC starts near 0.2 when adding the 0.1% of the transient data (which is the value that would be calculated by a naïve detector that assigns scores randomly). This metric stays there until adding roughly 1–2% of the transient data, at which point it jumps sharply to just over 0.8. As a reminder, it was not expected to achieve PR-AUC values very close to 1.0 because the anomalies are inserted as ramp functions, meaning the anomalies with near-zero magnitude are still considered anomalies (but are nearly undetectable). Moving to the baseline: base and transient case, the PR-AUC starts in the range of 0.4–0.55, which immediately shows that the base operating data is adding information to the detector. By around 1% of the transient data, this method has also achieved PR-AUC values of 0.8. Finally looking at the two methods focused on transient operating anomaly detection, both methods' performance start around 0.8 even with just 0.1% of the transient data added and remain relatively constant throughout. Based on these results, the two methods show clear benefit for the linear datasets.

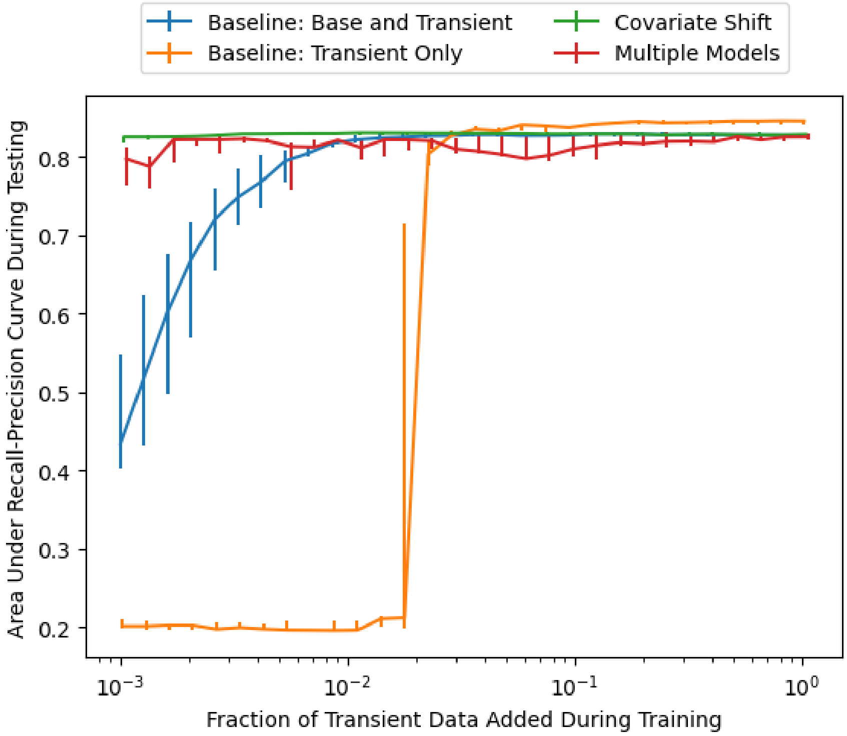

Figure 18. PR-AUC results for the linear dataset and linear prediction methods.

Looking at the 0.05 nonlinear datasets, the results for the nonlinear methods are shown in Figure 19. These results look very different from the results for the linear dataset. Up until roughly 10% added transient data, all the methods perform comparably to a naïve detector. This is very different from the results for the linear datasets, where just a small amount of transient data resulted in a significant performance increase. Starting at the 10% mark, all the methods except the multiple models approach follow a steep upward trajectory, before reaching a maximum value of roughly 0.8. Along this path, the baseline: base and transient, covariate shift, and frozen layers approaches appear to perform slightly better than the baseline: transient only case. This still suggests that having the base operating data provides some additional information but not as much as for the linear datasets and methods.

Originally, it was assumed that the linear methods would be applied to the linear datasets and that the nonlinear methods applied to the nonlinear datasets. However, given that the nonlinear methods did not extend to transients as well as the linear methods, this effort also tested the linear methods on the nonlinear datasets. The results for the linear methods are shown in Figure 20. These results look markedly different from the nonlinear methods; up until that same 10% added mark, the covariate shift and multiple models approaches perform significantly better than the baseline: transient only approach and slightly better than the baseline: base and transient approach. At and beyond that 10% mark, the covariate shift, multiple models, and baseline: base and transient approaches see slight improvements, but generally remain constant, while the baseline: transient only approach rises past the other three methods and reaches relatively high performance levels.

Finally, looking at the 0.1 nonlinear datasets, the results for the nonlinear and linear methods are shown in Figure 21 and Figure 22, respectively. Starting with the nonlinear methods, these results are similar to the results for the 0.05 nonlinear datasets, except that these results show more clearly that the baseline: base and transient, covariate shift, and frozen layers methods outperform the baseline: transient only method for small amounts of transient data added. In other words, they show that the base operating data does add value. However, the three methods mentioned that do contain base operating data all perform fairly similarly to each other. Moving to the linear methods, again the trends are similar, with the covariate shift approach performing better with less transient data added but the baseline: transient only method performing best past a certain amount of data.

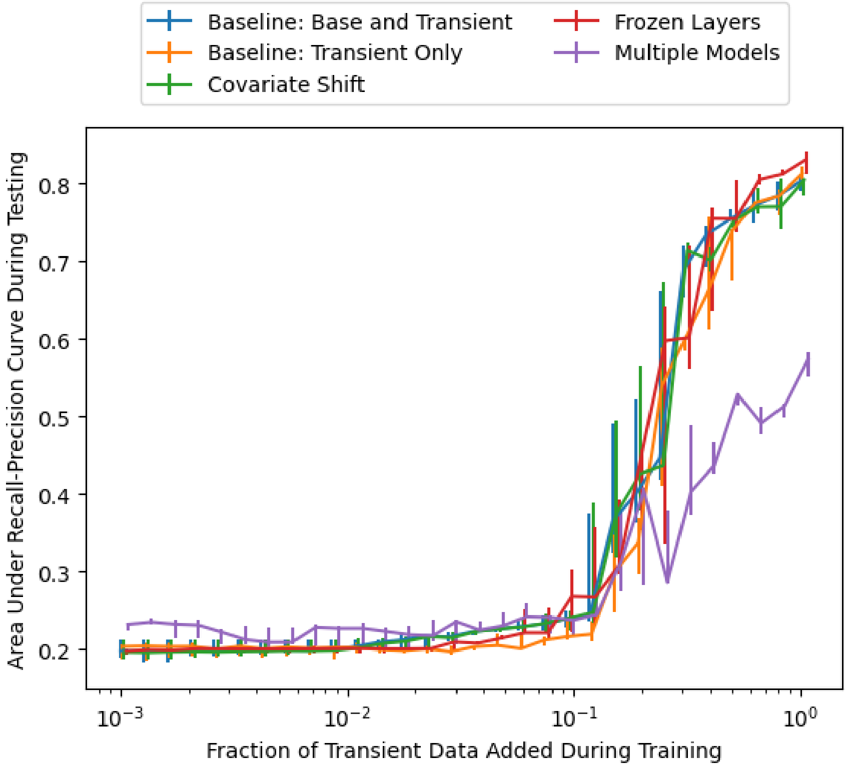

Figure 19. PR-AUC results for the 0.05 nonlinearity dataset and nonlinear prediction methods.

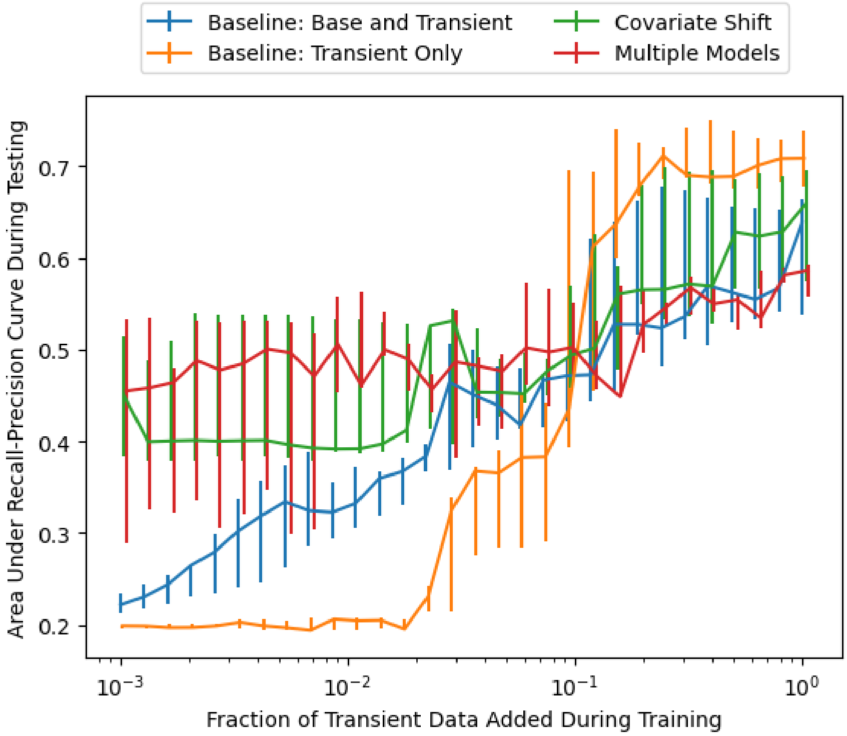

Figure 20. PR-AUC results for the 0.05 nonlinearity dataset and linear prediction methods.

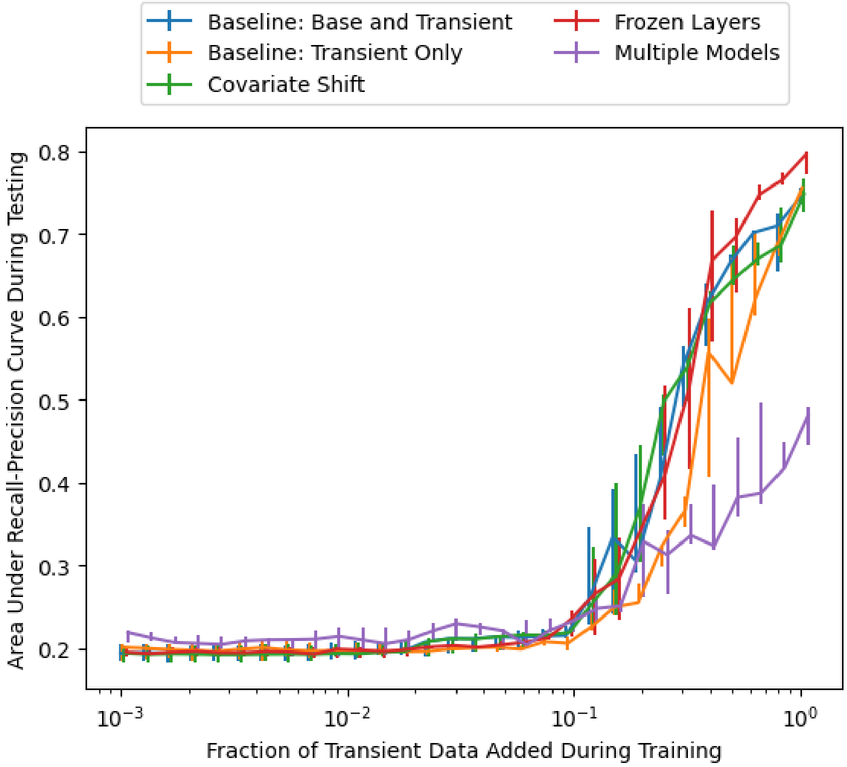

Figure 21. PR-AUC results for the 0.1 nonlinearity dataset and nonlinear prediction methods.

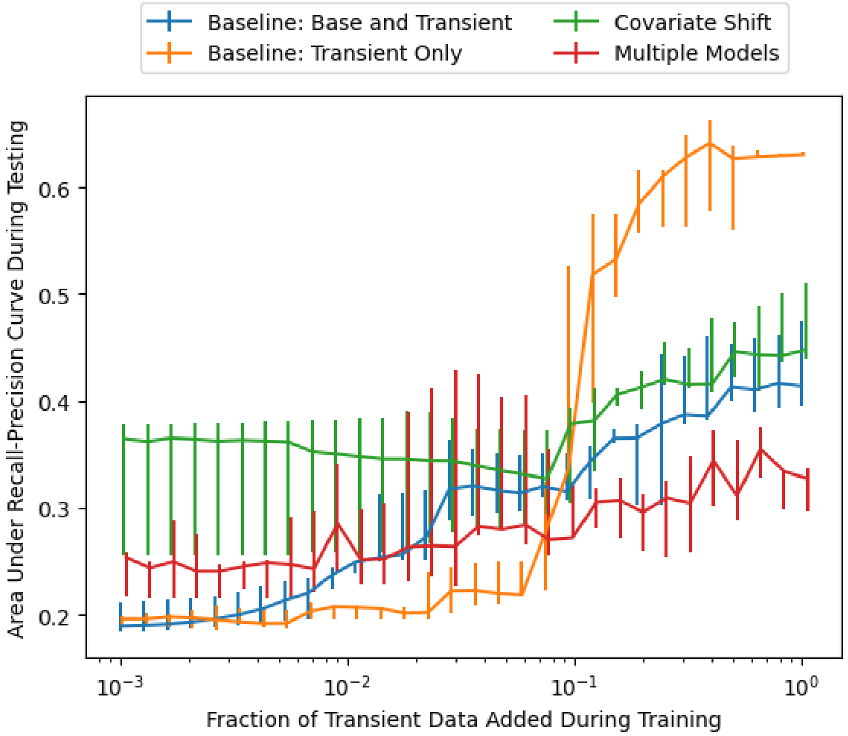

Figure 22. PR-AUC results for the 0.1 nonlinearity dataset and linear prediction methods.

4.3 Feature-Based Methods

The feature-based methods include the baseline methods (Section 4.3.1) and the combined null space method (Section 4.3.2). The results of these methods are presented and compared in Section 4.3.3. These methods all make use of the PCA model (Section 2.1).

4.3.1 Baseline Methods

The feature-based methods implemented the same two baseline approaches as the prediction-based methods. The first baseline method used both the available full power and transient power data, and the second used only the available transient power data. However, these baseline approaches used the PCA model for anomaly detection instead of the LOVO models.

4.3.2 Combined Null Space

As mentioned previously, this effort developed and implemented a new feature-based transfer learning approach to anomaly detection when there are sparse transient data. The general idea is to identify features that transfer from the source data to the target data, with the objective of finding those feature that are uninformative under normal operations yet show larger values during anomalous conditions. Intuitively, the developed approach searches for common patterns between the two datasets, measured using cosine similarity.

Similar to the multiple models approach, the data are separated into full power and transient power data X_F and X_T, respectively. Then, the PCA procedure of Section 2.1 is applied to each dataset, resulting in compressed approximations and sets of k_1 and k_2 left singular vectors, $\{u_{F,i}\}_{i=1}^{k_1}$ and $\{u_{T,i}\}_{i=1}^{k_2}$, for the two datasets, respectively. In essence, each set of left singular vectors describes most of the variance in its respective dataset. The intersection of the two sets $\{u_{F,i}\}_{i=1}^{k_1} \cap \{u_{T,i}\}_{i=1}^{k_2}$ forms the orthonormal basis (i.e., feature set) that transfers from the source to target datasets, whereas the union $\{u_{F,i}\}_{i=1}^{k_1} \cup \{u_{T,i}\}_{i=1}^{k_2}$ forms the basis that captures relevant variations in the data in both datasets. Here, the union is the relevant set and is calculated from the perspective of vector spaces by finding the cosine similarity c_i between each vector in the target domain $\{u_{T,i}\}_{i=1}^{k_2}$and the subspace spanned by the source domain $\{u_{F,i}\}_{i=1}^{k_1}$ using the projection operator $U_F U_F^T$:

$$c_i = \frac{U_F U_F^T u_{T,i}}{\left\| U_F U_F^T u_{T,i} \right\|_2}.$$

Then starting with all vectors in the source domain, vectors are added from the target domain if they are below a user-defined similarity threshold (i.e., they are not similar enough to be a duplicate and thus should be in the union).

Once the union is calculated, it can be used in a similar manner to the standard PCA detection described in Section 2.1 where uninformative left singular vectors can be used to look for anomalous patterns. If the union is defined as $\{u_i\}_{i=1}^{k} := \{u_{F,i}\}_{i=1}^{k_1} \cup \{u_{T,i}\}_{i=1}^{k_2}$, the uninformative left singular vectors may be constructed using the null space [20] of the union set, $\mathcal{N}\left(\{u_i\}_{i=1}^{k}\right)$, spanned by a basis of $ms - k$ orthonormal vectors, $\{u_i\}_{i=k+1}^{ms}$. This null space operation is equivalent to finding a discarded set of features that is uninformative with respect to trends in both datasets but highly informative for detecting anomalies. This is because the union operator combines patterns that explain trends in both sets, and the corresponding null space therefore contains patterns that do not explain trends but the error instead. As discussed in Section 2.1, this error-like term can then be used to flag anomalous data.

4.3.3 Results

This section shows results for the feature-based methods applied to the different datasets. The full set of analyzed methods and datasets is summarized in Table 4.

Table 4. Summary of datasets, detection methods, and extension methods analyzed for feature methods.

Datasets	General Detection Method	Transient Extension Method
Linear datasets		Baseline: transient only
0.05 nonlinear datasets	PCA	Baseline: base and transient
0.1 nonlinear datasets		Combined null space

For the linear datasets, the results for the two baseline and combined null space methods are shown in Figure 23. Like the prediction methods, the baseline: transient only case starts low and increases rapidly after obtaining more training data. However, unlike the prediction methods, the baseline: base and transient case shows strong performance across the full range of transient data added. This is also true for the combined null space approach.

Looking at the results for the 0.05 and 0.1 nonlinear datasets (Figure 24 and Figure 25), the plots show similar trends to the prediction methods. The methods that include base operating data show better results with very limited transient data compared with the nonlinear prediction method results (Figure 19 and Figure 21), but that performance does not increase with more transient data. In addition, the baseline: transient only method starts out with very poor performance, but increases past the other methods with additional transient data.

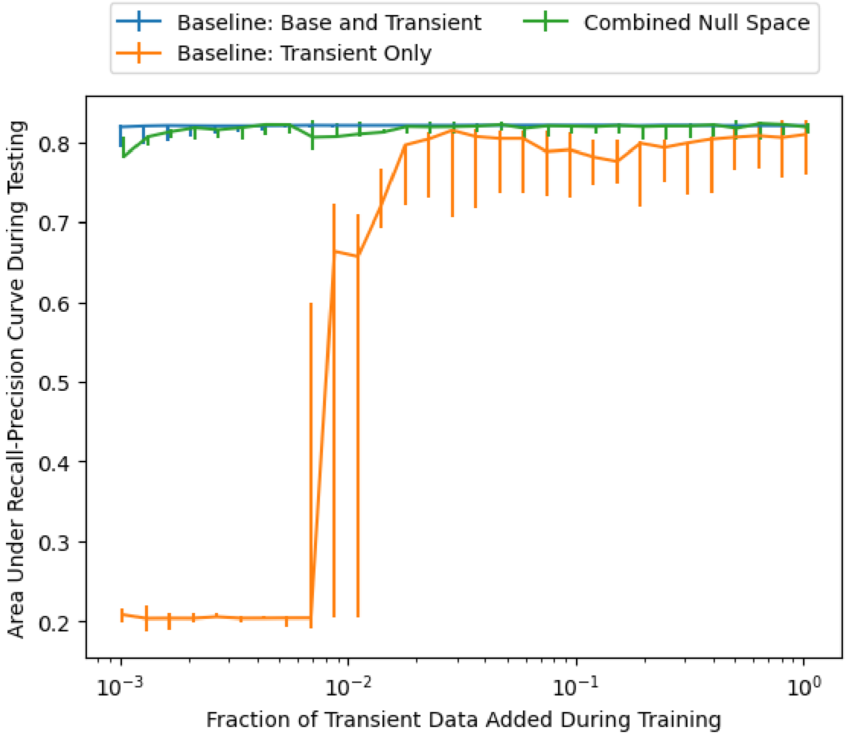

Figure 23. PR-AUC results for the linear dataset and linear feature methods.

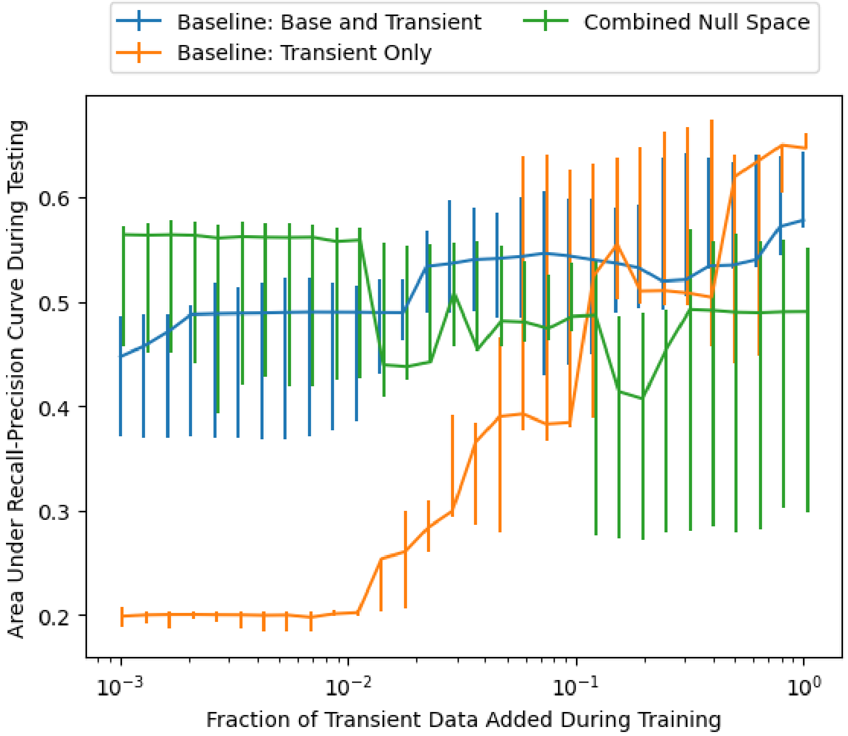

Figure 24. PR-AUC results for the 0.05 nonlinear dataset and linear feature methods.

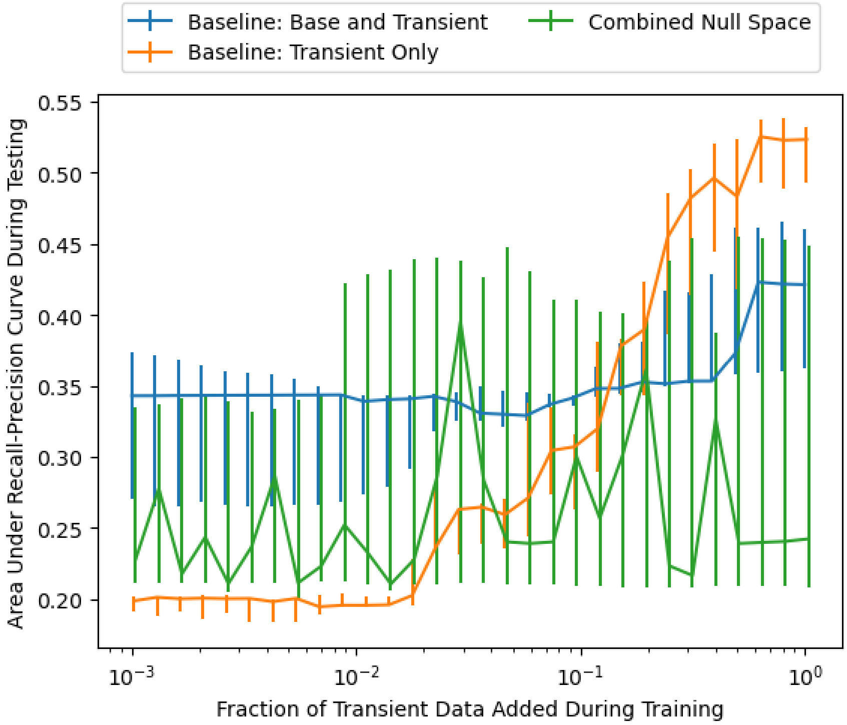

Figure 25. PR-AUC results for the 0.1 nonlinear dataset and linear feature methods.

5. CONCLUSIONS

This effort tested the hypothesis that anomaly detection methods can be extended to improve performance during the data-poor transient conditions compared with baseline methods that are trained without regard to the limited transient data. To accomplish this, it compared three prediction-based methods (covariate shift, multiple models, and frozen layers) and one feature-based method (combined null space) to the baseline approaches. These methods were all tested on SMD datasets over a range of nonlinearity measures and amounts of transient data included in the training data.

Starting with the linear datasets and prediction-based methods, the methods that included the base operating data performed significantly better than the baseline approach that did not include base operating data (i.e., they required significantly less transient data to reach comparable performance). This showed that there was noticeable benefit in including the base operating data. Comparing the extension methods with the baseline approach that did include base operating data, the covariate shift and multiple models approaches both showed significant improvement over the baseline. These two methods that accounted for the transient were able to reach peak performance (AUC over 0.8) with just 0.1% of the transient data added, whereas the baseline with transient and base required nearly 10× more data to achieve comparable performance. Moving to the feature-based methods, the extension method performed similarly better than the baseline approach without base operating data. However, here the baseline with base data and combined null space methods achieved very similar performance (AUC over 0.8) to each other with just 0.1% of the data.

Based on these results, including base operating data made a significant difference in anomaly detection methods for linear datasets. For particularly data-sparse applications, any of the covariate shift, multiple models, or PCA-based methods (baseline or combined null space) provided strong and comparable performance with very limited transient data.

Moving to the nonlinear datasets and nonlinear prediction-based methods, the nonlinear methods did not benefit as much from including the base operating data compared with the linear datasets and linear methods. Here, all the methods achieved results equivalent to a naïve detector up until 10% of the transient data was added. After this there appeared to be some benefit from including base operating data; although, it is less noticeable compared with the linear datasets. Moving to the linear prediction and feature based methods, these methods showed a significant performance improvement over the nonlinear methods at the lower end of the transient data added. Compared with the baseline with transient only method, all the methods that included base operating data achieved better performance (AUC of 0.4–0.5 compared with an AUC of 0.2). The covariate shift and multiple models approaches both performed better than the baseline with the base operating data method. The two feature-based methods performed even better (AUC of 0.45–0.56) than the prediction-based methods.

Based on these results, nonlinear methods were not well suited to the nonlinear transient problem without significant amounts of data; however, the linear methods applied to the nonlinear datasets showed some success. This implies that, even though the overall dynamics of the SMD datasets are nonlinear, there must be some linear patterns within that the methods are recognizing and learning. And these patterns still hold when transferring from base operating data to transient operating data. In addition, it appears the feature-based methods performed better than the prediction-based methods on these datasets when given very small amounts of transient data, although this advantage was not observed as more data were added. One possible explanation for this is that the methods are finding just the features that are linear and ignoring the other effects, while the prediction-based methods may not be able to extract just the linear features as accurately.

Combining all of this, it appears that, for linear datasets, the transient problem is solvable and multiple methods can achieve good results. For nonlinear datasets, the transient problem is much more difficult and, for very limited transient datasets, may only be solvable when some linear patterns exist that can be extracted.

6. REFERENCES

1. Al Rashdan, A., M. Griffel, R. Boza, and D. Guillen. 2019. "Subtle Process Anomalies Detection Using Machine-Learning Methods." INL/EXT-19-55629, Idaho National Laboratory. https://lwrs.inl.gov/Advanced%20IIC%20System%20Technologies/Subtle_Process-Anomalies_Detection_Using_Machine-Learning_Methods.pdf.

2. Al Rashdan, A., H. Abdel-Khalik, K. Giraud, L. M. Griffel, D. Guillen, and A. Varuttamaseni. 2020. "An Applied Strategy for Using Empirical and Hybrid Models in Online Monitoring." INL/EXT-20-59688, Idaho National Laboratory. https://lwrs.inl.gov/Advanced%20IIC%20System%20Technologies/Applied_Strategy_Using_Empirical_Hybrid_Models.pdf.

3. Farber, J., A. Al Rashdan, H. Abdel-Khalik, Y. Li, M. Abdo, D. Mandelli, and D. Huang. 2021. "Process Anomaly Detection for Sparsely Labeled Events in Nuclear Power Plants." INL/EXT-21-64303, Idaho National Laboratory. https://lwrs.inl.gov/Advanced%20IIC%20System%20Technologies/ProcessAnomalyDetectionSparselyLabeled.pdf.

4. Farber, J., et al. 2023. "Anomaly Detection and Identification Using a Leave-One-Variable-Out Method." NPIC HMIT 2023, Knoxville, TN, July 15–21, 2023.

5. Sugiyama, M., M. Krauledat, and K.-R. Müller. 2007. "Covariate Shift Adaptation by Importance Weighted Cross Validation." Journal of Machine Learning Research 8.5 (May): 985–1005. https://jmlr.org/papers/volume8/sugiyama07a/sugiyama07a.pdf.

6. Yang, Y., K. Zha, Y. Chen, H. Wang, and D. Katabi. 2021. "Delving into Deep Imbalanced Regression." International Conference on Machine Learning. PMLR, 139: 11842–11851. https://proceedings.mlr.press/v139/yang21m.html.

7. Torgo, L., R. P. Ribeiro, B. Pfahringer, and P. Branco. 2013. "SMOTE for Regression." Progress in Artificial Intelligence: 16th Portuguese Conference on Artificial Intelligence. EPIA 2013, Lecture Notes in Computer Science, 8154: 378–389. http://dx.doi.org/10.1007/978-3-642-40669-0_33.

8. Torgo, L., P. Branco, R. P. Ribeiro, and B. Pfahringer. 2015. "Resampling strategies for regression." Expert Systems 32(3): 465–476. http://dx.doi.org/10.1111/exsy.12081.

9. Wu, J., Z. Zhao, C. Sun, R. Yan, and X. Chen. 2020. "Few-shot transfer learning for intelligent fault diagnosis of machine." Measurement 166: 108202. https://doi.org/10.1016/j.measurement.2020.108202.

10. Lee, M.-C., J.-C. Lin, and E. G. Gran. 2021. "SALAD: Self-Adaptive Lightweight Anomaly Detection for Real-Time Recurrent Time Series." IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC). IEEE, Madrid, Spain, July 12–16, 2021. https://doi.org/10.1109/COMPSAC51774.2021.00056.

11. Khalastchi, E., G. Kaminka, M. Kalech, and R. Lin. 2011. "Online anomaly detection in unmanned vehicles." The 10th International Conference on Autonomous Agents and Multiagent Systems. IFAAMAS 2011, 1: 115–122. https://dl.acm.org/doi/10.5555/2030470.2030487.

12. Reris, R., and J. Brooks. 2015. "Principal Component Analysis and Optimization: A Tutorial." 14th INFORMS Computing Society Conference. ICS2015, 212–225. http://dx.doi.org/10.1287/ics.2015.0016.

13. Johnson, R. A., and D. W. Wichern. 2023. *Applied Multivariate Statistical Analysis* (6th edition). London: Pearson.

14. Ku, W., R. Storer, and C. Georgakis. 1995. “Disturbance detection and isolation by dynamic principal component analysis.” Chemometrics and Intelligent Laboratory Systems 30(1): 179–196. https://doi.org/10.1016/0169-7439(95)00076-3.

15. Zou, H., and T. Hastie. 2005. “Regularization and Variable Selection via the Elastic Net.” Journal of the Royal Statistical Society Series B: Statistical Methodology 67(2): 301–320. https://doi.org/10.1111/j.1467-9868.2005.00503.x.

16. Zhai, J., S. Zhang, J. Chen, and Q. He. 2018. “Autoencoder and Its Various Variants.” 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE, Miyazaki, Japan, October 7–10, 2018. https://doi.org/10.1109/SMC.2018.00080.

17. Li, X. R. 2012. “Measure of nonlinearity for stochastic systems.” 2012 15th International Conference on Information Fusion. IEEE, Singapore, July 9–12, 2012. https://ieeexplore.ieee.org/document/6289928.

18. Silverman, Bernard W. 1986. *Density Estimation for Statistics and Data Analysis*. Monographs on Statistics and Applied Probability 26. CRC Press.

19. Scott, David W. 2015. *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons. https://www.doi.org/10.1002/9781118575574.

20. Hefferon, Jim. 2018. *Linear Algebra*, Third edition. https://joshua.smcvt.edu/linearalgebra/book.pdf.