# Manifold Alignment and Wavelet Analysis For Fault Detection Across Machines

Hala Mostafa, Soumalya Sarkar, George Ekladious
United Technologies Research Center
East Hartford, Connecticut
mostafh,sarkars,ekladigs@utrc.utc.com

## ABSTRACT

Fault detection and isolation (FDI) is of paramount importance in industrial settings that involve electric, mechanical, electronic or cyberphysical systems. Data-driven approaches to FDI typically use machine learning to classify a pattern of sensor readings as faulty or healthy. Despite the buzz around Big Data, data scarcity is still an issue in many situations, especially in industrial settings where data collection can be time consuming or require unavailable/expensive sensors. Exacerbating the data scarcity issue is the fact that differences between training and deployment settings preclude the direct application of learned models in new settings. Moreover, data from a new setting may be scarce, which precludes training a model from scratch. For example, we may have more data from a given machine, but need to do FDI for a different machine of the same family, or the same machine but in a different deployment environment. In this paper, we address the problem of FDI across multiple machines. We present a novel combination of 1) wavelet analysis to extract useful features from time series data from accelerometers mounted on a machine; and 2) manifold alignment, a well-known heterogeneous domain adaptation approach, to do transfer learning across different machines. Our results demonstrate that: 1) We can leverage data from different deployments and different machines to improve the accuracy of FDI in a new settings; 2) We can successfully learn across machines even if one of them has missing sensors and 3) We can improve learning accuracy by incorporating domain knowledge into the manifold alignment approach. All our experiments and reported results are based on sensor data from real instrumented machines.

**ACM Reference Format:**
Hala Mostafa, Soumalya Sarkar, George Ekladious. 2018. Manifold Alignment and Wavelet Analysis For Fault Detection Across Machines. In *MiLeTS '18, August 2018, London, United Kingdom.* ACM, New York, NY, USA, Article 4, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 INTRODUCTION

Despite the prevalence of Big Data, data scarcity is still an issue in many situations, especially in industrial settings. While in Information Retrieval and similar problems large volumes of data are available (e.g., image databases), data collection in industrial settings can be time consuming, require expensive or unavailable sensors or necessitate a well-trained work force.

Fault detection and isolation (FDI) is of paramount importance in industrial settings that involve electric, mechanical, electronic or cyberphysical systems. With the increasing ability to instrument these systems with a variety of sensors, FDI made the logical move from rule-based approaches to data-driven. Data derived from sensor readings obtained over time are processed in an attempt to distill patterns associated with healthy operation or a given fault. Unsupervised machine learning approaches operate without ground truth labels of the particular faults associated with the collected data and resort to clustering, whereby healthy behavior is assumed to cluster together in some feature space, and similarly for faulty behaviors. Supervised approaches require fault labels and treat the problem as that of learning a classifier capable of mapping data derived from sensor readings to the class corresponding to the fault exhibited in the system, if any.

In almost any realistic application, differences between training and test data preclude the direct deployment of learned models in new settings. For example, in industrial settings we may have more data from a given machine, but need to do FDI for a different machine from the same family, or the same machine but in different deployment environment. Bridging the gap between a *source* dataset and a *target* dataset is the focus of Transfer Learning [12]. The difficulty of the transfer depends on how different the source and target data distributions are. In covariate shift [6], one of the simplest settings of TL, the source and the target datasets differ in the distribution of the covariates, $x$ but agree on the conditional distribution of the label given the covariates $P(Y|X)$. *Domain Adaptation (DA)* goes a step further and drops the assumptions about how the joint distribution $P(X, Y)$ differs between the source and target. However, DA assumes that the feature or covariate space is the same across both datasets ([2, 4, 5].

*Heterogeneous Domain Adaptation (DA)* goes even a step further and allows transfer between datasets whose covariates belong to different feature spaces. HDA has been successfully used to do object recognition on diverse image datasets, in addition to sentiment classification and text categorization across multi-lingual corpora [1, 14, 15, 17]. To our knowledge, HDA has never been applied to time series data, or for the purpose of fault detection and isolation.

In this paper, we address the problem of Fault Detection and Isolation using time series sensor data. We operate on time series data from three accelerometers from two different machines we have access to, where each machine is observed in 2 different settings. From each accelerometer, we use wavelet analysis to extract features for FDI. We then apply transfer learning to accommodate and account for differences between training and deployment settings.

*MiLeTS '18, August 2018, London, United Kingdom*

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

The contributions of this paper are as follows:

(1) Using wavelets for feature extraction from time series data.
(2) Applying HDA to the FDI problem across different machines and/or different deployment settings. We demonstrate improved performance compared to only using data from the new setting.
(3) Demonstrating the practicality and improved FDI performance of HDA when some of the sensors readings are missing in the new setting.
(4) Incorporating simple domain knowledge into the HDA formulation and demonstrating the resulting benefits compared to the absence of domain knowledge.

This paper is organized as follows: Section 2 gives a brief background on Heterogeneous Domain Adaptation, manifold alignment and time series feature extraction. This section also gives background on wavelet analysis and details our wavelet feature extraction approach. Section 4 details the manifold alignment formulation of the FDI problem with transfer across settings of the same machine, different machines, different sensor sets, as well as using domain knowledge to improve FDI performance. We then summarize our contributions and discuss future work.

## 2 BACKGROUND

### 2.1 Heterogeneous Domain Adaptation

Heterogeneous Domain Adaptation (HDA) addresses situations where we have 1 or more source datasets and the goal is to leverage them to improve learning on a target dataset, despite the difference in their feature spaces. One way of categorizing HDA approaches is according to their need for labeled target data, where some approaches require few labeled points [1, 9], while others are unsupervised and require only labeled points from the source datasets [14].

Another categorization of HDA approaches depends on how they calculate a) a transform of the source datasets that makes it "look like" it came from the target distribution; or b) a transform for each dataset, whether source or target, into a common space; or c) a machine learning model (e.g., classifier) without explicitly calculating or representing the data transforms.

The advantage of the first two types of approaches is that once transformed, the distinction between source and target datasets vanishes, allowing us to *apply standard ML to the new combined dataset* and to leverage labeled source data [5, 10, 16]. Finding the best transform amounts to solving an optimization problem. For example, subspace methods [5] find the best projection matrices to project source and target data to a common latent subspace. Sparse dictionary coding methods [10] represent a point as a sparse combination of codewords and transfer across domains by having a shared codebook. Examples of optimization criteria include preserving local geometry (unsupervised setting), preserving geometry with regards to labels (semi- and fully-supervised), and matching empirical source and target covariate distributions (unsupervised [5]).

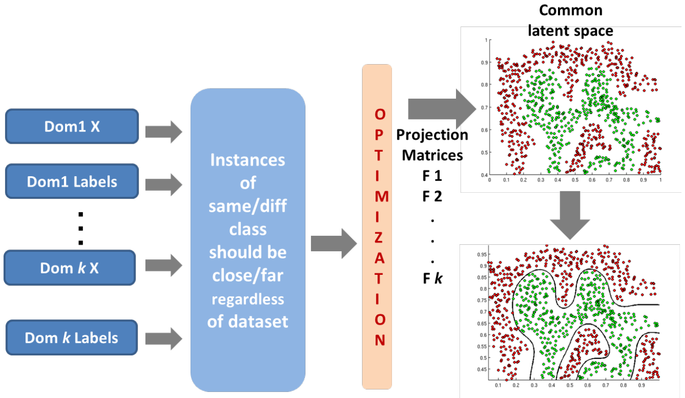

**Figure 1: HDA using MA to jointly project all datasets to a common latent space.**

### 2.2 Manifold Alignment

*MA for correspondence.* Manifold Alignment (MA) was introduced in 2003 [7] where it was used to learn correspondences between objects/points in different datasets with the aid of a low dimensional representation. The algorithm trains on data consisting of pairs of corresponding objects from 2 domains and during testing, it discovers unknown correspondences between objects in the test dataset. For example, one data set could consist of images of an object taken from multiple viewpoints, and another data set consists of images of a different object from different viewpoints. Simple regression does not work because of the high dimensionality of the original/raw feature space and the small number of given correspondences.

*MA for classification.* In 2011, Wang and Mahadevan extended MA for HDA, with the combined dataset used to do classification [16]. Unlike the use of MA for correspondence, in classification tasks have correspondences indicated by the class labels and the goal is not to learn new correspondences, but to discover a common space in which the downstream machine learning task (e.g., classification) can be performed. Figure 1 illustrates this process.

Wang formulates the problem of finding a common latent space to which the different datasets are projected as an optimization problem. The goal is to find projection matrices, one per dataset, that satisfy the following criteria:

(1) Preserving local neighborhoods within a dataset: if two points in a domain are close in the dataset's original feature space, their projections under the dataset's projection matrix should also be close.
(2) Class separability: if two points have different labels, they should be projected to points far away in the latent space. This should hold true regardless of the domains they belong to.
(3) Class homogeneity: if two points have the same label, they should be projected to points close together in the latent space. This should hold true regardless of the domains they belong to.

The MA approach can therefore be seen as a preprocessing step that acts on the datasets (both covariates and labels) by plugging them into an optimization problem whose output is a set of projection matrices, one per domain. The original datasets are projected using their respective matrices to give a combined dataset in the latent space. This combined dataset is then used to train traditional ML classifiers. During testing, points from the target dataset are projected using the corresponding learned projection matrix into the latent space, where the trained classifier operates on it.

Formally, the MA optimization problem has an objective function made up of the following quantity:

$$\begin{array}{llll} A = & \mu\Sigma_k & \Sigma_{ij}\|F_k^T x_i - F_k^T x_j\|^2 & W_k(x_i, x_j) \\ B = & \Sigma_{kl} & \Sigma_{ij}\|F_k^T x_i - F_l^T x_j\|^2 & W_s(x_i, x_j) \\ C = & \Sigma_{kl} & \Sigma_{ij}\|F_k^T x_i - F_l^T x_j\|^2 & W_d(x_i, x_j) \end{array}$$

For each domain $k$, $F_k$ is its projection matrix and $W_k$ is the similarity matrix defined over points in it. $W_s$ is a matrix defined for every pair of points, across domains, indicating whether they belong to the same class.

$$W_s(x_i, x_j) = \mathbb{I}_{c(x_i)==c(x_j)}$$

$W_d$ is a similarly defined matrix indicating whether a pair of points belong to different classes.

$$W_d(x_i, x_j) = \mathbb{I}_{c(x_i)\neq c(x_j)}$$

Quantity $A$ iterates over every domain $k$ and tries to preserve local neighborhoods. Quantities $B$ and $C$ iterate over every pair of domains $k$ and $l$ and promote inter-class separability and intra-class homogeneity. The goal is to minimize $A$ and $B$ while maximizing $C$. The objective function is therefore

$$\min_{F_1,..F_m} (A + B)/C$$

With simple manipulation, it can be seen that the above can be solved in closed form where the optimal $F = [F_1; F_2; \ldots F_m]$ is the solution to the generalized eigenvalue problem

$$F(\mu L + L_s)F^T x = \lambda F L_d F^T x \tag{1}$$

Where $L = W - D$ is the combinatorial Laplacian matrix defined over all pairs of points in all domains and $L_s$ and $L_d$ are Laplacians obtained from $W_s$ and $W_d$, respectively.

*Latent space dimensionality.* In the MA approach, the user needs to provide the dimension of the resulting latent space (i.e., the dimension of the space that the matrices $F_k$ project the data to). For a chosen dimensionality $d$, we construct the projection matrices from the first $d$ eigenvectors of the eigenvector matrix $F$. Different values of $d$ will yield representations in different latent spaces, with different ease of separability for the subsequent classification step. Choosing the best value of $d$ a priori is an open research problem.

*MA vs PCA.* It is useful to compare the MA-based HDA (Figure 1) to an HDA approach based on PCA (Figure 2) to project each dataset down to the same dimension. This comparison highlights the shortcomings of PCA compared to MA:

- PCA calculates the projection for each dataset independently of the others, as opposed to simultaneously optimizing over all projection matrices as in MA.

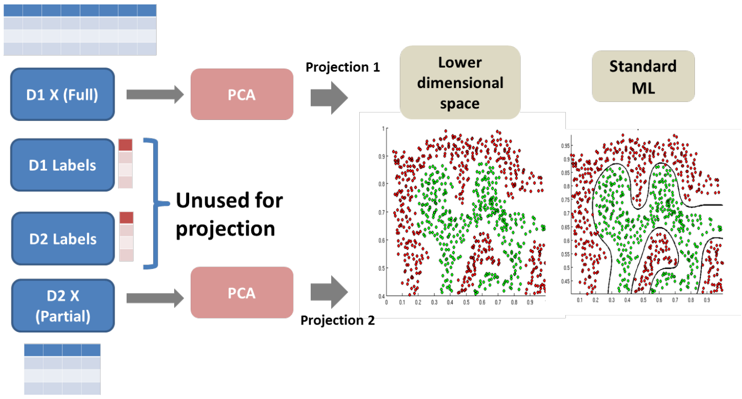

**Figure 2: HDA using PCA to project all datasets down to the same dimensionality.**

- PCA does not make use of the labels when projecting, potentially resulting in projections that obscure, rather than enhance, differences between classes.

## 3 WAVELET FEATURE EXTRACTION

In the following sub-sections, we give some background material on wavelets and the wavelet transform and briefly discuss their use in related applications and the challenges therein. We then outline how we extracted wavelet features for FDI.

### 3.1 The wavelet transform

The wavelet feature extraction method belongs to the linear time-frequency representations family, where signals are decomposed into a weighted sum of a series of bases localized in both time and frequency domains [8, 11]. For example, the wavelet Short-Time Fourier Transform (STFT) represents the signals in a time-frequency-energy space so that the constituent frequency components and their time variation features can be revealed. Unlike the STFT approach, the *wavelet transform* employs wavelets, instead of sinusoidal functions, as the basis, so it has a zooming and adaptive windowing capability which makes it effective for time-frequency localization, and is suited to transient signal analysis.

For a signal $x(t)$, the wavelet transform is defined as

$$WT_x(t, a) = \frac{1}{\sqrt{a}} \int x(\tau)\psi\left(\frac{\tau - t}{a}\right) d\tau$$

where wavelet $\psi\left(\frac{\tau - t}{a}\right)$ is derived by dilating and translating the wavelet bases $\psi(t)$, $a$ is the scale parameters, $t$ is the time shift and $1/\sqrt{(a)}$ is a normalization factor to maintain energy conservation.

### 3.2 Wavelet features for FDI

Whenever machines are running under time-varying conditions, nonstationary signals are being produced where the task of FDI becomes more challenging since the fault signatures differ over time. Time-frequency analysis can be used to identify the constituent components of signals and their time variation, and thus reveal the time variant features of the nonstationary signals [3]. In other words, a segmented short duration signal usually does not change too much and hence can be assumed to be stationary.

Wavelet analysis has been widely used in machinery fault diagnosis. The wavelet transform was used to analyze the transient features, extract the impulse characteristics or suppress the background noise of vibration signals to diagnose faults of turbo-machinery, gearboxes, and internal combustion engines [19]. Peng and Chu surveyed wavelet analysis and its applications in mechanical vibration signal analysis [13].

For the impulse detection issue in localized fault diagnosis of bearings and gears, the wavelet transform might be an effective approach, because the energy of an impulse mainly concentrates in higher frequency band and wavelet transform has fine time localization in higher frequency band [3].

Despite the success of using wavelet features, there are some challenges [3]. One of the main challenges is that the wavelet transform suffers from a trade-off between time localization and frequency resolution. The higher the resolution in time, the lower the resolution in the frequency domain and vice versa. Due to the trade-off between time localization and frequency resolution, the resolutions in time and frequency domains cannot reach their highest levels concurrently.

The variable time localization and frequency resolution enables the wavelet transform to zoom and adapt its window to suit non-stationary signals. Wavelet transform iteratively decomposes the approximation signals of lower frequency, but does not further work on the detail signals of higher frequencies. For higher frequency components, wavelet transform has a better time localization but a lower frequency resolution. For lower frequency components, the frequency resolution is higher whereas the time localization is worse.

In order to mitigate this limitation, the multi-resolution wavelet analysis was introduced, where multiple transforms with different resolution tradeoffs are employed and combined to capture the signal characteristics along the time and frequency scales [11].

## 3.3 Wavelets features from our accelerometers

For variety of fault types, accelerometer signals show both temporally local and global characteristics. To capture this type of multi-time scale behavior in a compressed fashion, we use *multi-resolution wavelet analysis* (MRA) [11] for feature extraction from signals collected by 3-axis accelerometers. A 2-level MRA generates one approximate and 2 detail levels of coefficients that encapsulate low frequency and high frequency contents from accelerometer signal segments. As the final step of feature extraction, the mean and variance of the top coefficients from each of the three levels (1 approximate and 2 detail) are calculated and a six-dimensional feature is obtained from a channel of 3-axis accelerometers. **Therefore each data point in this paper consists of 18 features**.

# 4 WAVELETS AND MA FOR FDI

## 4.1 Data, metrics and baselines

**Data**

Our time series sensor data was collected from multiple sensors on our machines. Wavelet transform was applied to the raw sensor data as discussed in Section 3. The resulting 18 wavelet features are what we use to do fault detection and isolation.

**Table 1: Sizes of data sets for each machine and setting.**

| Machine | Setting | Number of points |
|---|---|---|
| M4 | 13 | 151 |
| M4 | 15 | 157 |
| M5 | 13 | 121 |
| M5 | 14 | 123 |

The machines we have are called M4 and M5. M4 has data from 2 different settings; M4-13 and M4-15 while M5 has settings M5-13 and M5-14. The number of data points from each machine and setting is given in Table 1.

**Faults:** Each data point is labeled as having no fault, which we also refer to as F0, or having fault F3 or F5. The meanings of these faults are specific to our machines and will be omitted.

Fault detection and isolation in our particular setting presents the following challenges:

- The machines can have varying degrees of fault severity for each type of fault. There is thus potentially large variation within points having the same label, some of which may look like baseline (fault-free) points.
- Different machines have different baseline behaviors and may exhibit faults differently, precluding the straightforward use of data across machines, even if they are equipped with the same number and type of sensors.

**Classifier**

For the downstream FDI classification task, we used a weighted KNN classifier with K=5. We set the weight of a training point depending on its class. The goal of the weighting is to help rectify the imbalance in the data where the number of faulty points far exceeds the number of no-fault points. As such, each baseline point was weighted at 1.5.

**Metrics**

In keeping with the HDA literature, the quality of an HDA approach is measured by the performance of the downstream ML task. Since FDI is a classification task, we use the following standard metrics:

- False alarm rate is the number of baseline points classified as faulty. This measures the fault detection performance.
- Classification score is the fraction of correctly classified points across all fault types and non-faulty points. This measures the fault isolation performance.

**Other approaches**

As discussed earlier, comparing MA-based HDA and PCA-based HDA gives an appreciation of the importance of the supervised joint projection done by MA, so we compare these 2 approaches. Both PCA and MA have performance that depends on $d$, the user-provided dimension of the latent space. As such, we ran experiments that explore the FDI performance of the downstream classifier across the entire range of $d$, which is 1 to the maximum dimensionality (18).

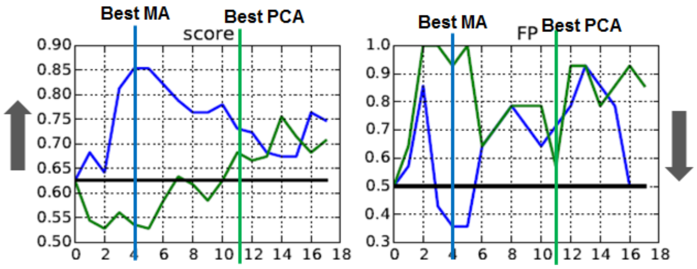

**Figure 3: Classification accuracy and false positive rate as a function of latent space dimension. Train on M5-13 and test on M5-14.**

We also compare to an baseline approach which we refer to as *raw space* baseline where we train the classifier on a dataset obtained from naively combining the source datasets in their original spaces without any transformation. This is of course only possible when the datasets have the same feature spaces. Essentially, this baseline makes the naive assumption that all machines exhibit faults, or lack thereof, in a similar manner.

### 4.2 Same machine, same setting

In this set of experiments, we investigate whether MA and PCA can project data to a space where classification is easier. We train and test on the same machine, but different settings.

Figures 3 and 4 show performance (classification score and false positive rate) at different values of the dimension $d$ for machines M4 and M5. The black horizontal line shows the performance of the classifier in the raw feature space where it is trained on the original wavelet features of the training set and applied as-is to the test set without projecting either set into any intermediate space.

We consistently found it harder to detect and isolate faults on M4 than on M5, as can be seen by the lower score and higher false positive rates in the figures. For M4 (Figure 4), FDI in the raw space has 100% false alarm rate, which renders it useless. At $d = 1$, MA projects the dat from M4-13 and M4-15 down to 1 dimension where the classification accuracy drops to 50% but the false alarm rate drops to 45%, which makes it a usable, albeit not very accurate, approach.

For M5 (Figure 3), the best accuracy-false positive tradeoff is at $d = 4$ for MA and $d = 11$ for PCA. As can be seen, MA raises fault isolation accuracy from around 63% in the raw space to 85% in a 4D space while reducing fault detection false positive rate from 50% in the raw space to 33% in a 4D space.

From the figures, we can see that MA finds latent spaces that:

- Have much lower dimensionality (MA achieves its best performance with only a 4D latent space)
- Enhance the distinction between baselines and the different faults

As can be seen, performance is not monotonically increasing in $d$, which makes the choice of $d$ a priori impossible. In the rest of this paper, we present the results of the “best” dimension for each approach, where best is determined by manual inspection of a reasonable trade-off between the 2 evaluation metrics. We will report results of MA vs. PCA as in Figure 5.

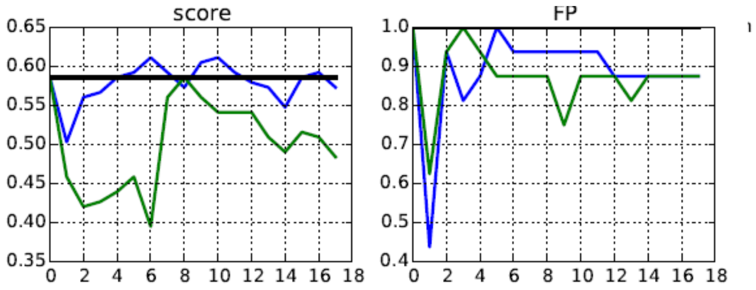

**Figure 4: Classification accuracy and false positive rate as a function of latent space dimension. Train on M4-13 and test on M4-15.**

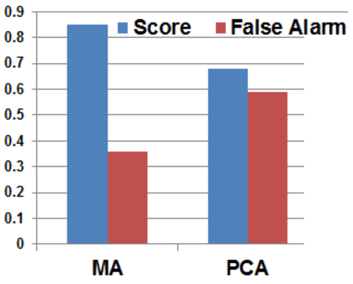

**Figure 5: Classification accuracy and false positive rate. Train on M5-13 and test on M5-14.**

### 4.3 Transfer across machines equipped with the same sensors

Given the challenging nature of learning an FDI model machine M4, we investigated whether including some data from machine M5 can improve learning performance. Specifically, our experiment compares learning performance when testing on M4-15 data after training on M4-13 only vs. training on M4-13 and M5-13.

Doing MA on M4-13 gives projection matrix $P^A_{4-13}$. Doing MA on M4-13 and M5-13 gives projection matrices $P^B_{4-13}$ and $P_{5-13}$. Figure 6 shows the process of combining data. We then apply $P^A_{4-13}$ and $P^B_{4-13}$ to data from M4-15 and compare FDI classifier performance. The results are shown in Figure 7. Despite the difference in machines, including data from M5 significantly reduced the false alarm rate, essentially taking a model that was unusable (70% false alarms) and making it usable.

### 4.4 Transfer across machines equipped with different sensors

In the next set of experiments, we demonstrate the ability of our approach to leverage data that has a different feature space. This is particularly useful for transfer across machines fitted with different sensors, or in the case of some machines losing one or more sensors. In our setting, we investigate whether we can leverage data

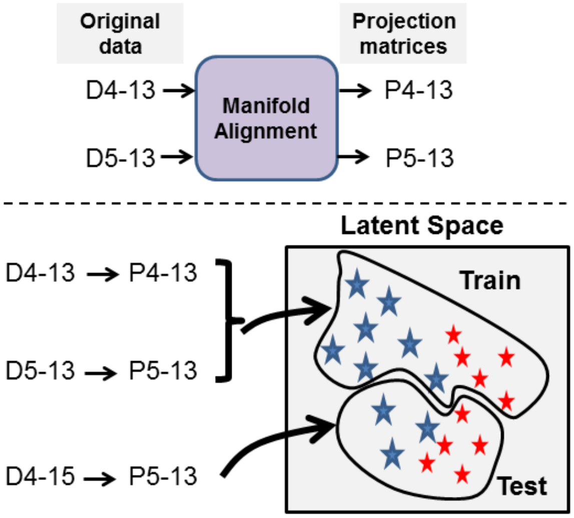

Figure 6: Training and testing on data combined from multiple machines through manifold alignment

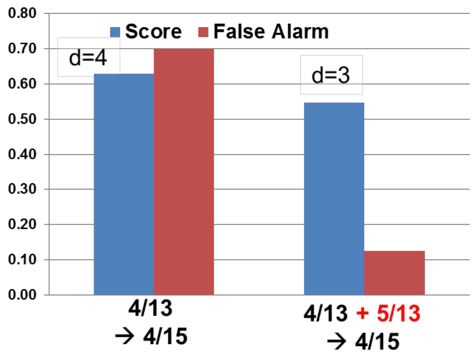

Figure 7: Effect of transfer across different machines on performance. Training on M4 only (left) vs. training on M4 and M5 (right)

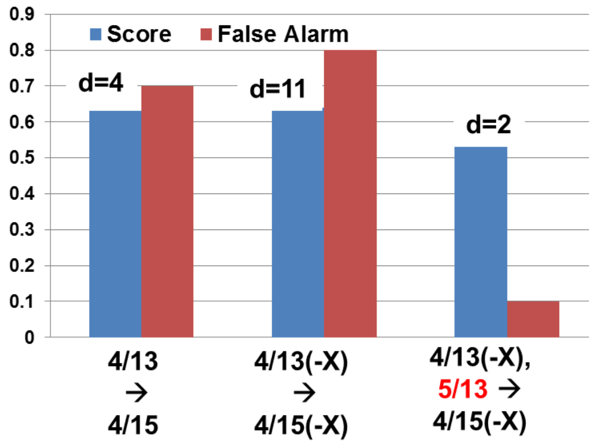

Figure 8: Effect of transfer across different machines with different sensors on performance.

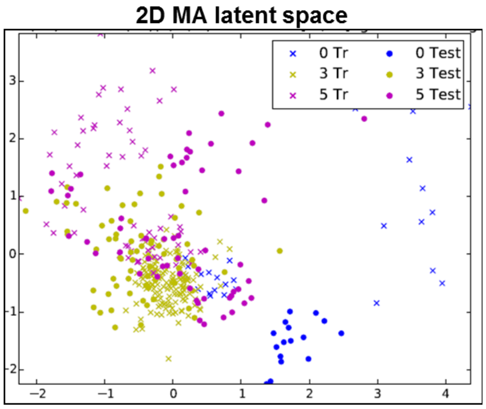

Figure 9: Data from the cross-machine setting plotted in the 2D latent space found by manifold alignment.

from machines that have a more âĂIJcompleteâĂİ set of sensors to augment data from, and improve performance on, machines that lack these sensors.

In order to test this, we simulated sensor loss by removing the 6 wavelet features associated with x-axis accelerometer on machine M4, which leaves us with 12 out of the original 18 features. We append the name of a dataset that is missing its x-axis accelerometer features with (-X).

As can be expected, performance dropped. The left and center sets of bars in Figure 8 show that without the x-axis features, the false alarm rate rose from 70% to 80%. Including data from M5-13, which has a full set of sensor features, greatly increased learning accuracy, bringing down the false alarm rate to 10% at the expense of some loss in accuracy.

**Low dimensional space:** Because MA can achieve good performance in a 2D latent space, we can visualize our training and test data in this new space as in Figure 9. This low dimensionality is a very useful by product of using MA that can shed light on which faults are likely to be difficult to distinguish and how the training and test datasets differ. Additionally, by inspecting the entries in the 2 eigenvectors used in the projection, we can also understand which features are most indicative of the underlying fault. In our setting, for example, the first of the 6 z-axis feature had a significantly larger entry than the rest, indicating its importance in learning a representation that aids in the classification task.

## 4.5 Transfer with domain knowledge

In many settings, domain experts or physics-based models can provide insights about how a new learning task differs from previous ones long before any data can be collected from it. In our previous work, we demonstrated increased data efficiency and improved learning performance from incorporating different types of high-level domain knowledge into a transfer learning framework that focused on the covariate shift and functional change settings [18].

Going back to the challenging learning task posted by machine M4, we investigated whether there are any domain-specific "hints" we can give the MA algorithm to help it project the data to a space where classification performs well.

One important observation is that differences between labels are not all equal. In our setting, the engineers and maintenance crew of the machines care very much about distinguishing baseline from faulty units, but perhaps not as much about distinguishing the different types of faults. As such, we can manipulate the penalty terms in the objective function to place more emphasis on projecting baseline points further away from all the faults. Note that this is different from, and serves an orthogonal purpose to, the weighting applied to the instances when running the KNN classifier. In the latter, baseline points are given more weight to compensate for the class imbalance resulting from most of the data coming from faulty machines.

Mathematically, we originally have the following terms in the objective function:

$$C = \Sigma_{kl}\Sigma_{ij}||F_k^T x_i - F_l^T x_j||^2 W_d(x_i, x_j)$$

where $W_d(x_i, x_j) = \mathbb{I}_{c(x_i) \neq c(x_j)}$ is a matrix of binaries.

Based on domain knowledge, we changed an entry $W_d(x_i, x_j)$ to depend on the labels of $x_i$ and $x_j$. For points $x_i, x_j$ (potentially from different datasets), we define $W_d(x_i, x_j) = c(x_i) == c(x_j)$ if both points are labeled as faults and $W_d(x_i, x_j) = v > 1$ if one point is baseline and the other is a fault. The above associates a higher penalty $v > 1$ for projecting a baseline point close to a faulty point and a lower penalty of 1 for projecting two points with different faults close to each other.

We experimented with the above modified formulation using $v = 3$. The results shown in Figure 9 demonstrate that placing more emphasis on separating baseline and faulty points decreased the false alarm rate to below 10% while slightly increasing classification accuracy.

The advantage of the MA approach is that it admits this type of manipulation of the penalty terms in the objective function to incorporate domain knowledge. For example, if there are certain types of points that we want to distinguish, we can customize the construction of the difference matrix $W_d$ even further.

## 5 CONCLUSION

In this paper, we address Fault detection and isolation (FDI) in an industrial settings where differences between training and test data preclude the direct deployment of learned models in new settings. Moreover, data from a new setting may be scarce, which precludes training a model from scratch.

We operate on time series data from accelerometers mounted on machines deployed in different settings. The goal is to learn a model that detects whether a fault and classify it if it exists. We show how we extract wavelet features from the time series data using multi-resolution wavelet analysis. We then show how we applied manifold alignment, a well-known heterogeneous domain adaptation approach, and demonstrated successful leveraging of data from different deployments and different machines to improve the accuracy of FDI in a new settings. We also showed the possibility of doing this across machines with different sensor. Finally, we showed that incorporating domain knowledge into the manifold alignment approach leads to further improvement in performance. All our experiments and reported results are based on sensor data from real instrumented machines.

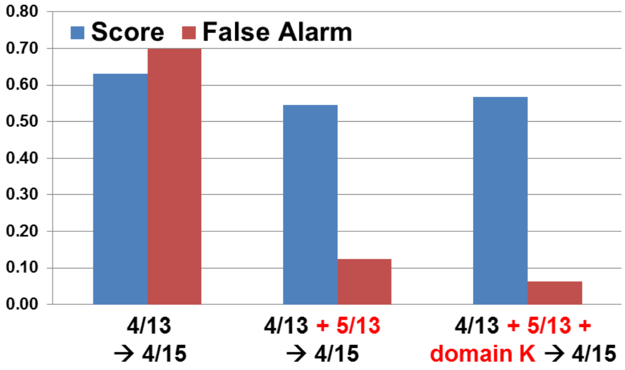

**Figure 10: Effect of including of leveraging both domain knowledge and data from M5 on learning performance for M4.**

For future work, we can explore the use of non-linear manifolds, which relaxes the assumption that the datasets, when mapped to a new latent space, must lie on a linear manifold, and can therefore broaden the applicability of the method. We will also explore incorporating different types of domain knowledge into the learning framework, either through regularization terms in the objective function, or constraints.

## REFERENCES

[1] W. Li et al. 2014. Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. In *IEEE transactions on pattern analysis and machine intelligence.*

[2] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. 2007. Analysis of representations for domain adaptation. In *Advances in neural information processing systems.* 137–144.

[3] Zhipeng Feng, Ming Liang, and Fulei Chu. 2013. Recent advances in timeâĂŞfrequency analysis methods for machinery fault diagnosis: A review with application examples. *Mechanical Systems and Signal Processing* 38, 1 (2013), 165–205.

[4] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. 2013. Unsupervised visual domain adaptation using subspace alignment. In *Computer Vision (ICCV), 2013 IEEE International Conference on.* IEEE, 2960–2967.

[5] B. Gong, Y. Shi, and F. Sha. 2012. Grauman and Kristen, "Geodesic flow kernel for unsupervised domain adaptation," in CVPR. In *CVPR.*

[6] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Sch"olkopf. 2009. Covariate shift by kernel mean matching. In *Dataset shift in machine learning.*

[7] Ji Hun Ham, Daniel D Lee, and Lawrence K Saul. 2003. Learning high dimensional correspondences from low dimensional manifolds. (2003).

[8] Nikolaj Hess-Nielsen and Mladen Victor Wickerhauser. 1996. Wavelets and time-frequency analysis. *Proc. IEEE* 84, 4 (1996), 523–540.

[9] Yao-Hung Hubert Tsai, Yi-Ren Yeh, and Yu-Chiang Frank Wang. 2016. Learning cross-domain landmarks for heterogeneous domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 5081–5090.

[10] M. Long, G. Ding, J. Wang, J. Sun, Y. Guo, and P. S. Yu. 2013. Transfer sparse coding for robust image representation. In *CVPR.*

[11] StÃľphane Mallat. 1999. *A wavelet tour of signal processing.* Academic press.

[12] S. J. Pan and Q. Yang. 2010. A survey on transfer learning. In *IEEE Transactions on knowledge and data engineering.*

[13] ZK Peng and FL Chu. 2004. Application of the wavelet transform in machine condition monitoring and fault diagnostics: a review with bibliography. *Mechanical*

*systems and signal processing* 18, 2 (2004), 199–221.

[14] Xiaoxiao Shi, Qi Liu, Wei Fan, S Yu Philip, and Ruixin Zhu. 2010. Transfer learning on heterogenous feature spaces via spectral transformation. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 1049–1054.

[15] C. Wang and S. Mahadevan. 2011. Heterogeneous domain adaptation using manifold alignment. In *IJCAI*.

[16] Chang Wang and Sridhar Mahadevan. 2011. Heterogeneous domain adaptation using manifold alignment. In *IJCAI proceedings-international joint conference on artificial intelligence*, Vol. 22. 1541.

[17] C. Wang and S. Mahadevan. 2013. Manifold Alignment Preserving Global Geometry. In *IJCAI*.

[18] Matthew O Williams and Hala Mostafa. 2016. Active Transfer Learning Using Knowledge of Anticipated Changes. In *IJCAI Workshop on Interactive Machine Learning*.

[19] et al Z. He, Y. Zi. 2001. *Fault Diagnosis Principles of Nonstationary Signal and Applications to Mechanical Equipment.* Higher Education Press, China.