Introducing AnomDB: An Unsupervised Anomaly Detection Method for CNC Machine Control Data

Lou Zhang[1], Sarah Elghazoly[2], and Brock Tweedie[3]

[1,2] *MachineMetrics, Northampton, MA 01060, United States*
lou.zhang@machinemetrics.com
sarah.elghazoly@machinemetrics.com
brock.tweedie@machinemetrics.com

ABSTRACT

We propose the application of unsupervised machine learning to automatically detect anomalous behavior on Computer Numerically Controlled (CNC) machines. We achieve this through an approach that utilizes Principal Component Analysis (PCA), time series feature extraction with the `anomalous` package in R, and Density Based Scanning of Applications with Noise (DBSCAN). We call this method *AnomDB*. Time series data collected from CNC machines may benefit from this technique due to its ability to consolidate noisy, multivariate data from CNC machine controls and detect anomalies without reliance on periodicity of signal. We perform experiments on CNC machine control data to demonstrate the effectiveness of this method in discovering anomalies over other commonly used methods of anomaly detection such as Interquartile Range (IQR) and k-means clustering. We show the effectiveness of this method through a case study of an actual machine anomaly, and then on a series of real machining data with synthetic anomalies injected.

1. INTRODUCTION

1.1. CNC Machine Control Data and The Anomaly Detection Problem

The integration of information technology in the discrete manufacturing industry is an active topic of research for applications in reducing waste, tracking of equipment, and automating and increasing efficiency. CNC machine control data is the in-process data collected during machine execution that can continually monitor work tasks, manufacturing resources, and operational status. Research using CNC machine control data has been conducted in quality assurance (Tiwari, Vergidis, Lloyd, & Cushen, 2008) (Ertekin, Kwon, & Tseng, 2003), improving and further automating process control (Kumar, Nassehi, Newman, Allen, & Tiwari, 2007), as well as tool condition monitoring and estimating remaining useful life (RUL) (Duan, Makis, & Deng, 2019) (Chen & Jen, 2000).

As the scale of Industrial Internet of Things (IIoT) applications in discrete manufacturing has exponentially expanded over the past several years, there is a growing opportunity for employing general-purpose diagnostic algorithms that can robustly operate on a wide variety of machines (vertical mills, horizontal lathes, grinders, stamping machines, etc.) that are manufacturing a wide variety of parts (Swiss turned parts such as fasteners, connectors, gears, etc., medical devices, aircraft components, firearm components, etc.). In particular, the basic question of "Is the machine operating normally?" is both universal and often of critical importance.

We therefore explore here, from a very general perspective, the possibility of a machine- and part-agnostic algorithm for anomaly detection using CNC machine control data. Such an algorithm can become an important component in any number of diagnostic applications, including helping to predict or categorize machine failures, flagging manufacturing defects, and spotting corruption within the data stream itself. Our focus will be on automated classification of completed part cycles as either "normal" or "anomalous," based on the behavior of other part cycles that occurred nearby in time.

Anomaly detection in time series data is not a new problem (Jinka, 2015). However, delineating "normal" versus "anomalous" behavior from raw CNC machine control data signals requires addressing a number of properties and constraints that, when taken in combination, become rather specific to this domain:

- *Real-time operation*. An ideal setup would alert operators to anomalous part cycles in (near) real-time for immediate follow-up, based on the recent history of a given machine. There may also be data storage limitations that

prohibit offline analysis and computationally expensive algorithms. Algorithms that minimize compute in the form of conserving processor and memory overhead are therefore preferred.

- *Multidimensional*. CNC machine control data can include quantities such as feed rate, tool positions, tool loads, and spindle speeds. What specific quantities are provided depend on the machine, and which quantities might be relevant for anomaly detection can change depending on the specific combination of machine and part type.

- *Repeated discrete periods of complicated activity*. Under ideal operation for a given manufacturing job, a CNC machine produces copies of the same part over and over through identically-repeated steps. This provides well-delineated and nominally-identical time series that can be compared across part cycles in the same job. The time series for each individual part cycle will typically follow a very complicated, erratic-looking path through the space of control data variables.

- *Irregularities between periods*. In reality, CNC operations are subject to a large number of environmental factors such as tool wear, small variations in materials, lubrication levels, temperatures, etc. The resulting "normal" control data time-series therefore contain a lot of part-to-part irregularities in quantities like the amount of time to complete individual steps, the magnitudes of torques, etc.

- *Commonly under-sampled*. A number of factors (discussed in more detail below) can limit the control data sampling rate to the O(1 Hz) level, whereas changes in the machine state often occur at rates $>$ 1 kHz. This limitation can heavily enhance the appearance of the intrinsic part-to-part irregularities. For example, a short-duration burst in spindle load might be sampled for one part cycle, but not for the next one. Figure 1 provides a demonstration of this.

In particular, the last two effects compromise methods that rely on signal periodicity or thresholds, as are commonly used for time series data analysis in other domains (e.g., finance).

As a concrete example, we show in Figure 2 the time series for three control variables over five adjacent part cycles, where the third part cycle exhibits a clear anomaly. In this example, an extended period of a single value across all three metrics in the third part signature can be seen. Although there are slight differences between each part signature (the exact timing of peak values, the duration and magnitude of load or speed, etc.), part signatures 1, 2, 4, and 5 all share signal similarities. Determining if any of these individual part cycle time series are "anomalous" requires contending with this severe distortion of the already-complicated underlying signals.

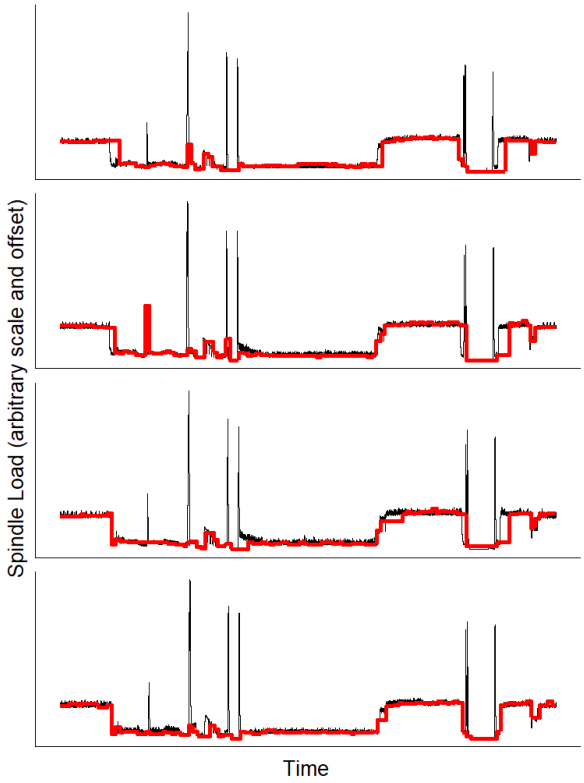

Figure 1. Aliasing occurring when high-frequency control data, in black (1 kHz) is sampled at a low rate, in red (1-4 Hz). Each plot shows spindle load for a single part cycle, which exhibits consistency in the high-frequency domain but irregularity in the low-frequency domain. Note the second part cycle, which samples a load spike while the other part cycles do not recognize it.

The under-sampling issue can arise for several practical reasons. A basic one is the cost overhead in high-rate data acquisition and cloud computing. These costs accrue from establishing high-speed network infrastructure through purchasing data processing and storage resources. There are also concerns that high-frequency data collection from the CNC machines' computers can interfere with operations on some models (H. Atluru & Deshpande, 2009).[1]

[1]

> CNC controls are designed for motion control and coordination, and for holding tight tolerances over long service lives. Diverting excessive compute power away from core functions and to data acquisition can, in extreme cases, disrupt the core motion control function of a CNC. This 'data starvation' condition creates stuttering, jerky, or slowed motion and is also common when trying to run large part programs on older machines with insufficient RAM. If data collection is causing the starvation, it can be resolved by re-configuring to a reduced polling interval or sample rate.
>
> —Russell Waddell, Managing Director MTConnect Institute, July 17, 2019

An effective, truly general-purpose and economical part cycle anomaly detection algorithm for CNC machines will therefore need to rapidly and adaptively identify relevant part-to-part variations in the multivariate control data space while robustly dealing with sizable, erratic artifacts from environmental factors and limited sampling rate.

1.2. Overview of Our Approach

We structure the problem of anomaly detection for CNC machines in the context of unsupervised machine learning. Using such an approach allows for a high degree of flexibility by automating the identification of regular structures in a data set without explicit labels or strong assumptions about the data's content.

Unsupervised machine learning has many impressive applications within ecology, medical imaging (Solan, Horn, Ruppin, & Edelman, 2005), language processing, etc. (Litjens et al., 2017). It is also an attractive approach to organizing multivariate time series data, especially in the realm of anomaly detection (Zhang et al., 2018). For example, unsupervised machine learning for automated detection of anomalies in multivariate time series data has been used for predicting cyber attacks and ecological anomalies (Recknagel, 2001) (Kang, Hyndman, & Smith-Miles, 2017).

Our proposed algorithm for anomaly detection in CNC machine part cycles constitutes a sequence of feature extraction/transformation steps on the raw part cycle time series, followed by clustering of the higher-level features. Outliers from the identified clusters are classified as "anomalous," and all other part cycles as "normal." This approach derives from and modifies the strategy introduced in (Hyndman, Wang, & Laptev, 2015), where time series feature extraction and outlier identification were applied to identify unusual Yahoo mail server activity.

The specific tools that we employ include dimensionality reduction via Principal Component Analysis (PCA), time-series feature extraction using the `anomalous` package (Hyndman, Wang, & Laptev, 2019), and Density Based Scanning of Applications with Noise (DBSCAN) (Hahsler & Piekenbrock, 2018). In outline, the steps are:

- Selection of an ensemble of part cycles from the same job for comparison, ideally nearby in time (such as the most recent $O(10–100)$).
- PCA dimensionality reduction in the space of (normalized) control data variables, using the full set of instantaneous readings from all of the considered part cycles. Each part cycle's time series is then reduced to one dimension by projecting into the first principal component.
- Extraction of several higher-level features (spectral entropy, autocorrelation, etc) from each part cycle's time series using `anomalous`.

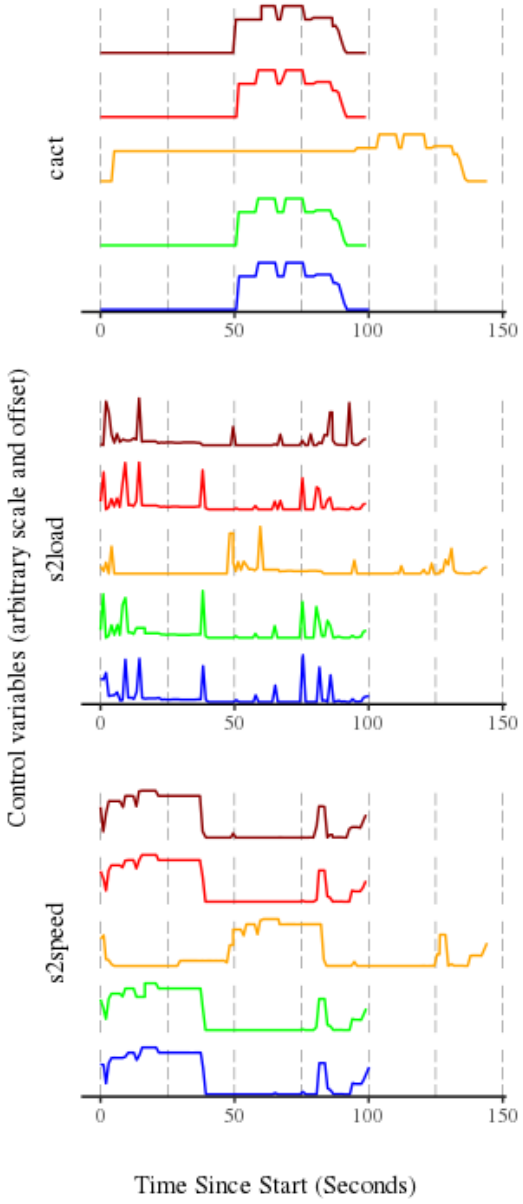

Figure 2. Examples of raw signal of C position, servo load and servo speed on a CNC machine (horizontal lathe), segmented into individual part cycles. An anomalous part cycle can be seen in orange (see main text). However, the signals exhibit significant variability between part cycles even during normal operating conditions. Methods of anomaly detection that rely on thresholds or periodic signals would face significant challenges in this context.

- A second PCA dimensionality reduction in the space of these (normalized) higher-level features, over the ensemble of part cycles.
- DBSCAN clustering of the retained principal component higher-level features over the ensemble of part cycles. Outlier points in the clustering are then identified as "anomalous" part cycles.

Taken together, these steps constitute the *AnomDB* algorithm.

Besides the change in domain of application (and especially the application to erratic and under-sampled time series), some new aspects that we explore beyond the original work of (Hyndman et al., 2015) include high dimensionality of the raw time series feature space and a density-based definition of outliers.

A review of the tools used in the algorithm, as well as some alternative tools, are given in Section 2. Specifics of the demonstration data set and parameter settings for the algorithm are then detailed in Section 3. *AnomDB*'s performance on a real data anomaly and in a variety of simulated anomaly scenarios is evaluated in Section 4. Finally, Section 5 contains concluding discussions.

2. REVIEW OF TOOLS

2.1. Dimensionality Reduction via PCA

Principal Component Analysis (PCA) is used on multidimensional data sets to capture an orthogonal set of statistically uncorrelated coordinates in the feature space. It allows for dimensionality reduction while retaining the dominant variations within a data set by projecting away all but the highest-variance coordinate directions (Hotelling, 1933) (Hyndman et al., 2015). The number of dimensions retained is a free parameter that can be specified depending on the problem.

For approximately periodic multivariate time series data, each sampled instant in time can be treated as an independent data point in the PCA computation. For the CNC machine control data, *AnomDB* uses the full set of instantaneous readings from all of the considered part cycles. Upon computing the highest-variance directions within the space of control variables, the part cycle time series values are then all projected into the subspace spanned by those directions.

The first step of PCA is the formation of a covariance matrix over all of the features. Since CNC machine control variables can be measured across a variety of conceptually distinct quantities with completely different units, each is scaled to unit variance. This is a common method for "standardizing" PCA. Eigenvalues and orthogonal eigenvectors of this matrix are then computed, and the eigenvalues ranked from highest to lowest. Smaller eigenvalues can be indicators of correlations between the (scaled) features, and their associated eigenvectors are therefore more likely to be associated with redundant information. Subsequently projecting out those directions reduces possible confusions in downstream machine learning pipelines and also reduces their computational overhead.

In addition to this PCA dimensionality reduction on the raw control data time series, *AnomDB* also includes a second PCA dimensionality reduction step in the space of higher-level time series features extracted using the `anomalous` package (discussed in the next subsection). The inclusion of this additional step was advocated for in (Hyndman et al., 2015). It not only further contains the complexity of the learning problem and helps avoid the "curse of dimensionality" (where all data points may appear in sparsely-populated regions when the number of features is large), but it also allows for great flexibility in what specific features are to be included. Determining the most relevant combinations of higher-level features then itself becomes part of the automated learning problem.

2.2. Time Series Feature Extraction via `anomalous`

Extraction of higher-level features is an important step in the processing of time series data. Even after the initial PCA step in the control variable space, each part cycle time series typically consists of hundreds of individual, highly-correlated but noisy measurements. For facilitating effective machine learning, the goal is to compute a handful of (mostly) non-redundant metrics that characterize the original signal data while aggregating redundant information and averaging-out noise effects. Such procedures effectively function as another layer of dimensionality reduction.

For *AnomDB*, we utilize the `anomalous` time series analysis package for R (Hyndman et al., 2019). This package provides computations of a number of higher-level features suggested in (Hyndman et al., 2015), ranging from simple mean and variance to more complicated features like spectral entropy. Not all of these are necessarily useful for detection of the types of anomalies encountered in CNC machine control data, and there may be other variables not included in `anomalous` that might also prove useful. For the present introductory study, we simply select a subset of `anomalous` variables based on empirical results (see Table 1), though inclusion of other features in *AnomDB* could be useful to explore.

2.3. Density Based Scanning for Outlier Detection

Density Based Scanning of Applications with Noise (DBSCAN) is a well-known clustering algorithm that produces arbitrarily-shaped clusters formed by maximal sets of density-connected points (Ester, Kriegel, Sander, & Xu, 1996) (Louhichi, Gzara, & Abdallah, 2014). DBSCAN can reveal possible latent classes within a dataset given user-specified parameters for the minimum number of objects in

a cluster ($minPts$) and the maximum separation between objects in a cluster (ϵ). It also identifies points in low-density regions that fall outside of any cluster, which turns it into a useful anomaly-detection tool.

As a preliminary step, DBSCAN performs a classification of points as either core, border, or outlier. This classification is determined using the ϵ-neighborhood of each point: the set of all points within a Euclidean distance ϵ, including itself. Core points are those whose ϵ-neighborhood contains at least $minPts$ points. Border points are non-core points whose ϵ-neighborhood contains a core point. Outliers are non-core points whose ϵ-neighborhood does not contain a core point.

DBSCAN subsequently identifies distinct clusters composed of core and border points, based upon a mutual density-reachable criterion. The outliers are otherwise set aside and not incorporated into clusters. For our anomaly-detection problem, we are only interested in identifying the outliers, and not the detailed cluster partitioning of the core and border points.

Note that the use of DBSCAN represents a departure from the corresponding anomaly-detection step in (Hyndman et al., 2015). There, methods were employed that function roughly as multidimensional generalizations of percentiles, allowing for identification of the "most outlying N points" or "most outlying $X\%$ of points". These require pre-specifying that a certain amount of the data will be categorized as "anomalous." With the density-based DBSCAN method, by contrast, anomalies may or may *not* be found in any given data set. This is more appropriate to a dynamic manufacturing environment, where anomalies are genuinely rare events of unknown frequency across different jobs.

Naive multidimensional generalizations of outlier detection based on interquartile range (IQR) could also be considered. In that approach, the data is iteratively projected onto each retained principal component feature. The IQR is computed as the distance between the 25% and 75% quartiles. Outliers are then identified as any points that lie $O(1)\times$ IQR below or above these quartiles, respectively, along any of the principal components. The method has the virtue of simplicity, but, unlike DBSCAN, its behavior is not necessarily robust to arbitrarily-shaped distributions of the data.[2]

2.4. Alternative Methods

Our focus here is on the chained application of PCA, time series feature extraction, and DBSCAN steps, which we have found to produce a powerful general-purpose anomaly-detector within the domain of CNC machine control data. However, several alternative methods were considered, some of which we use for comparisons in Section 4. We briefly outline these methods here.

Perhaps one of the simplest and most commonly-used anomaly detection methods for time series data is to set upper and lower thresholds on the raw signal based on the interquartile range (IQR). As noted in the previous subsection, for multivariate data this could be implemented by setting thresholds in each individual feature, though again some care needs to be taken when working in a high number of dimensions. (This is another place where PCA can be helpful.) For time series that cover a large range of values under "normal" conditions, windowing in time can also be employed. For periodic processes like repetitive part production, ideally we could instead compare different part cycles at the same moment of progress toward part completion. Throughout this paper, we employ IQR as defined by the `anomalize` package in R (Dancho & Vaughan, 2018) using default settings, which conducts IQR on the residuals of the input signal after Seasonal and Trend decomposition after LOESS (STL).

A major difficulty of the IQR approach with CNC machine control data traces back to the issues discussed in Section 1.1, namely the presence of high-frequency jumps, irregularities, and perturbations in step completion times, combined with the limitation of low-frequency sampling. This results in an uneven or choppy signal that can easily be confused with anomalous behavior based on simple thresholds. The problem is common both to comparing across different times for a given part cycle and to comparing across different part cycles at the "same" progress time. Another basic limitation is that anomalous machine behavior can include (quite frequently, in fact) long pauses or stops in the part cycle, which if within tolerances could be classified as "normal."

Another popular machine learning method that can be adapted for anomaly detection is k-means clustering. In a broad sense, k-means clustering differs from DBSCAN in that it is based on a global criterion of number of clusters k, rather than local criteria related to density. In (Chawla & Gionis, 2013), a generalization of k-means was proposed wherein also a given number ℓ of the data points are to be excluded from the clustering as outliers. These ℓ outliers are then identified dynamically via an optimization routine called k-means-- (which has been implemented in R as `kmodR` (Howe, 2015)). The entire set of part cycle time series can be decomposed into their individual multivariate control data measurements, and k-means-- applied to this aggregated set of measurements without regard to time-ordering. Individual instants in time for individual part cycle are then flagged as outliers, and any part cycle containing an outlier measurement as "anomalous." Appropriate values for k and ℓ can be determined algorithmically, for example using the method described in (Ray & Turi, 1999), which is based on finding "elbows" in the intra-cluster variation. Unlike

[2]For example, DBSCAN also has additional robustness to operating on mixtures of part types, in situations where information on part type is not available.

IQR, the k-means-- approach can adapt to muti-modal data, which is more characteristic of CNC machine controls.

A final method that we considered was Sequence Motif Discovery. Sequence motifs are conserved sequences of similar or identical patterns that reveal themselves in time series (Das & Dai, 2007). We found that sequence motifs were not able to perform at adequately low latencies for use in a live anomaly detection application, often taking 10-100x longer than *AnomDB* to surface anomalies. Thus, we do not include this in our comparison of methods.

In general, many other advanced tools for signal processing might be considered for anomaly detection in the context of highly erratic CNC machine part cycles. Their absence from this study should not be taken to constitute a judgement of their efficacy. However, we take the relative simplicity, speed, and performance of *AnomDB* as establishing a good baseline approach for this complicated domain-specific problem.

3. METHODOLOGY

3.1. Data Collection

For the development and evaluation of *AnomDB*, we have used a data set collected from approximately 600 CNC machines operating between June 2018 and June 2019. Control data was sampled at 1–4 Hz, depending on the machine. Control data was extracted with proprietary adapters as developed by MachineMetrics, Inc.

AnomDB has also been implemented in a production environment since June of 2018 (with settings as in the next subsection). The algorithm is run by machine on streaming data using all part cycles of common part type collected over the previous 4-24 hours. Typically, this involves approximately 10–500 part cycles, each containing several hundred instantaneous observations of O(10's) of control variables.[3] We present a case study of one example anomalous part cycle drawn from this data set in Section 4.1. A small subset of the production data is also used for algorithm performance evaluations and comparisons in Section 4.2, via the injection of synthetic anomalies.

3.2. *AnomDB* Parameter Settings and Feature Selection

AnomDB involves a number of steps and higher-level feature definitions that require choosing parameter values. Which features to use from `anomalous` must also be decided. We summarize here the choices and motivations made for the present study.

For the initial stage of PCA, used to reduce the dimensionality of the raw CNC machine control data variables, we have found that taking the first principal component suffices to achieve good performance while dramatically reducing processing time. On average, the time it takes to run *AnomDB* retaining all principal components is 16 times longer than when retaining only one. Generally, adding more control data principal components into the analysis degrades performance in the evaluations of Section 4.2.

For the feature set chosen from `anomalous`, we selected seven features that either appear to correlate well with human-tagged anomalies, that appear to vary significantly between different part types, or that empirically improve discrimination of known anomalies. We list these features with some basic descriptions in Table 1.

Table 1. Higher-level time series features extracted using `anomalous`. Unless otherwise indicated, we use default package parameters, where relevant.

Spectral Entropy	Describes the complexity of a signal in terms of its Shannon entropy $-\sum_i p_i \ln p_i$, where p_i represents the normalized power of a signal's discrete Fourier components
First-Order Autocorrelation	Measures the correlation between a time series and its one-step lagged series
Level Shift	Partitioning the time series into ten equal-sized blocks, compute the maximum absolute difference in mean between consecutive blocks
Variance Change	Partitioning the time series into ten equal-sized blocks, compute the maximum absolute difference in variance between consecutive blocks
Curvature	Strength of curvature determined from a global quadratic fit
Spikiness	Smoothing the signal with LOESS, compute the set of leave-one-out variances amongst the residuals, and then take the variance of those variance estimates
Flat Spots	Discretizing the observation variable into (at most) ten equal-sized intervals, the maximum number of consecutive observations within any one of those intervals

Two of the chosen features (Level Shift and Variance Change) require partitioning the time series into blocks and measuring changes between the blocks. For these, we take ten blocks as a default. We have verified that performance is only weakly sensitive to this choice. Otherwise, we use `anomalous` package defaults for any other parameters needed in the feature calculations.

The second stage of PCA operates within the space of these `anomalous` features. We choose the first two principal components as our default. We have found that adding more

[3] 3,117 anomalies were caught over 600 machines in this time (5.1/machine/year), resulting in dozens of customer-reported preventions or early-indications of machine failure. The actual number of failures prevented is difficult to measure because robust data collection methods for feedback have not been established.

principal components can potentially further improve performance, but for the algorithm's development and for the present studies we have restricted to two in order to aid in visualizations.

Finally, the anomaly-detection stage using DBSCAN has two free parameters: $minPts$ and ϵ. Our goal with this step is to identify the "bulk" of the data as one or a few clusters, and identify outliers as points that are isolated from both these main clusters and from other points. To achieve this behavior, we choose an ϵ that is of order the intrinsic spread of the data in the principal component dimensions, and $minPts \gtrsim 1$. For the former, we parameterize $\epsilon \equiv n_\sigma \sigma_{PCA1}$, where σ_{PCA1} is the standard deviation of the first (dominant) principal component. The value of n_σ has a strong effect on which points are to be classified as anomalies, and will be scanned below. However, in production we have found that $n_\sigma = 3$ works well. For $minPts$, we fix a value of five. Since anomalies are anyway to be associated with sparse regions of feature space, the results are not very sensitive to the specific value of $minPt$.[4]

4. RESULTS

4.1. Case Study

We provide here a representative example of the method drawn from live streaming data. The control data was collected from a horizontal lathe producing the same part continuously over a four hour period, and with twelve control variables including feedrate, position, load magnitude, and spindle speed. This data actually precedes (by several minutes) a catastrophic tool failure, and includes a number of "anomalous" part cycles identified by *AnomDB* that are nearby in time. For illustration, we focus on only one of these.

A handful of control variable time series for this illustrative part cycle, as well as some nearby "normal" part cycles, appeared already in this paper in Figure 2 in the Introduction. The first stage of the *AnomDB* pipeline is to project out only the first principal component of these control variables, which we now show in Figure 3. Even restricting to just this one linear combination of control variables clearly exhibits unusual behavior in this part cycle.

The anomaly-detection stage then proceeds over the space of `anomalous` feature variables, projected into their first two principal components. The data for this collection of part cycles, as represented in this feature space, is shown in Figure 4. The "anomalous" part cycles identified as DBSCAN outliers (with $minPts = 5$ and $n_\sigma = 3$) are indicated in green, and the chosen illustrative "anomalous" part cycle in red.

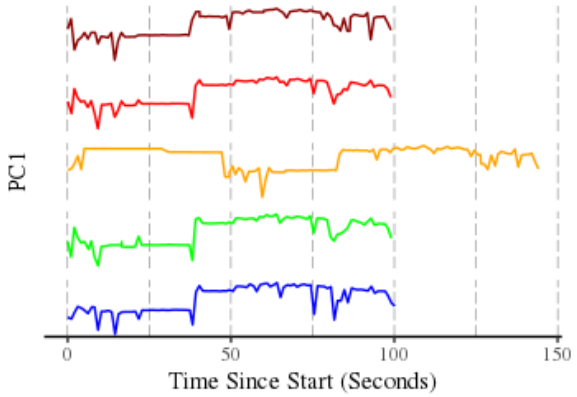

Figure 3. Principal Components for the anomaly shown in Figure 2 divided into individual part signatures.

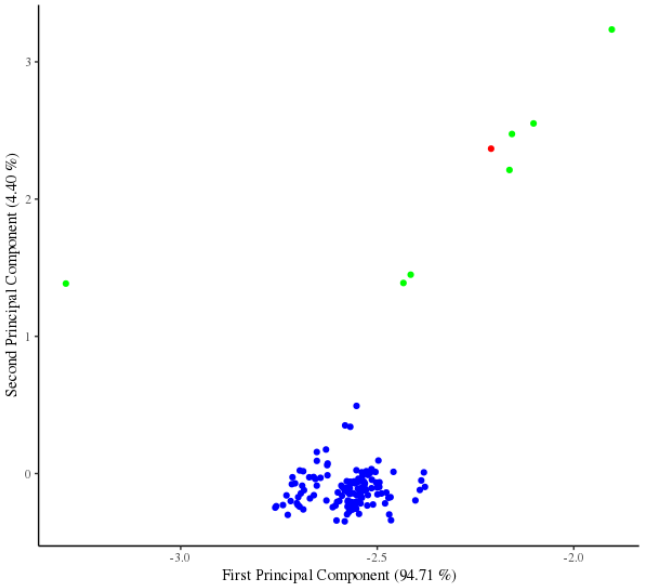

Figure 4. Clustered time series containing the anomaly of Figure 2 shown in red and other anomalies detected within the period sampled shown in green. The anomaly highlighted in red was the most recent anomaly before machine failure, and the one illustrated in the case study. The clustered points that represent normal part cycles are shown in blue.

[4]Note that these parameters could require some additional tuning in very high-dimensional principal component spaces.

4.2. Performance on Synthetic Anomalies

Because production anomalies are by definition rare events, the statistics for evaluating the performance of *AnomDB* on real data are small. The forms and contexts of individual anomalies are also highly heterogeneous. To make a more statistically meaningful and controlled evaluation, we have generated data sets containing synthetic anomalies.

To do this, we start with a small subset of the real machine data discussed in Section 3.1. We consider three specific machines: a horizontal lathe, a vertical mill, and a Swiss CNC machine. From each of these, 50 part cycles that appear "normal" by eye have been selected.

Synthetic anomalies of three types are considered, which can be used to replace a random 1/4 of an existing part cycle's time series across all control variables simultaneously:

- A 'drop' in the signal value to zero
- A flatline 'stop' in the signal
- A linear ramp 'hike' in the signal value from the minimum to the maximum within the replaced cycle segment.

These classes of anomalies were chosen based on our visual inspection of human-tagged production anomalies in the full data set. An example of the synthetic anomaly insertion for these three classes is shown in Figure 5.

For a given machine type and anomaly type, we choose a single random part cycle in which to introduce anomalies, and correspondingly a random time segment to replace. We construct sets of trials by repeating this procedure 50 times. In evaluating anomaly-detection performance using statistical metrics such as precision, etc., we average over the trials.

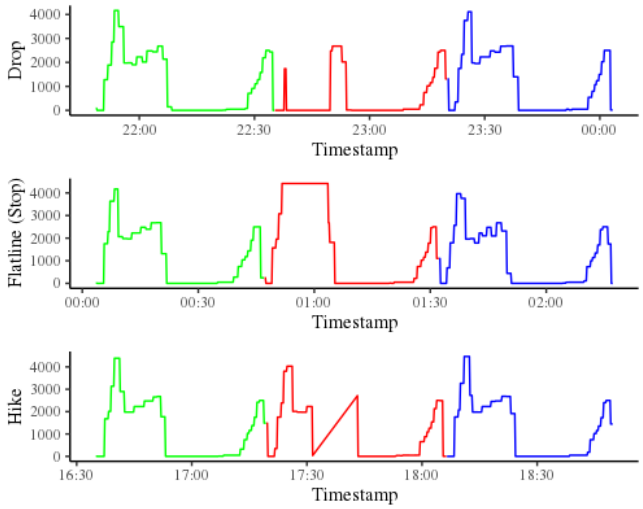

Figure 5. Example of three types of synthetic anomalies inserted into a non-anomalous streaming time series metric.

We use this clean set of synthetic anomalies to perform some evaluations of *AnomDB* and to compare it against two other time series anomaly-detection methods discussed in Section 2.4: IQR and k-means--. Each of these alternative anomaly-detection algorithms is also run on the control data time series projected onto their first principal component in order to make the comparisons more direct.

To evaluate and compare the three approaches, we generate ROC curves for true-positive versus false-positive rates for detection of our synthetic anomalies, scanning over respective algorithm parameters that strongly correlate with anomaly detection thresholds.[5] For *AnomDB*, this is the scale of the DBSCAN ϵ. For IQR, it is the number of IQR lengths (a real number) below/above the first/third quartile. For k-means--, this is the number of instantaneous outliers, but we also optimize over $k \sim O(1)$.

Comparison of the relative performance of *AnomDB* on the time series data from different machine types, as well as for the different classes of synthetic anomalies are shown in Figures 6, 7, and 8.

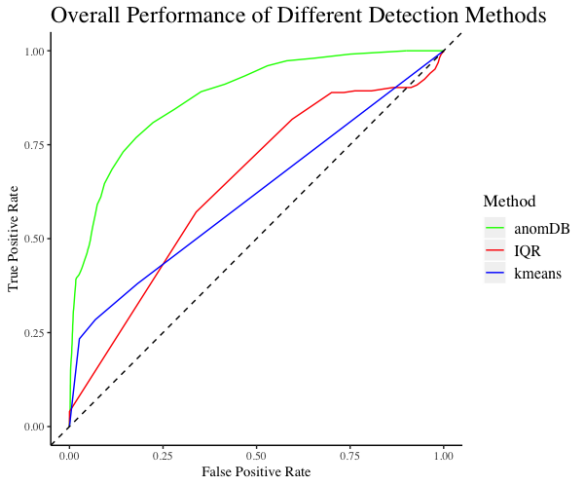

Figure 6. ROC Curve comparing the classifiers *AnomDB*, k-means--, and IQR, averaged over considered machine types and anomalies.

[5] We emphasize that anomaly detection is not a uniquely-defined problem, and that true-positive/false-positive rates are not universally well-defined concepts in this regard. Nor is the identification of anomalies generally to be thought of as a labeled learning problem. Our purpose here is to evaluate the algorithms' ability to flag certain classes of known anomalous patterns, not to "learn" these specific patterns from one data set and "predict" them in another.

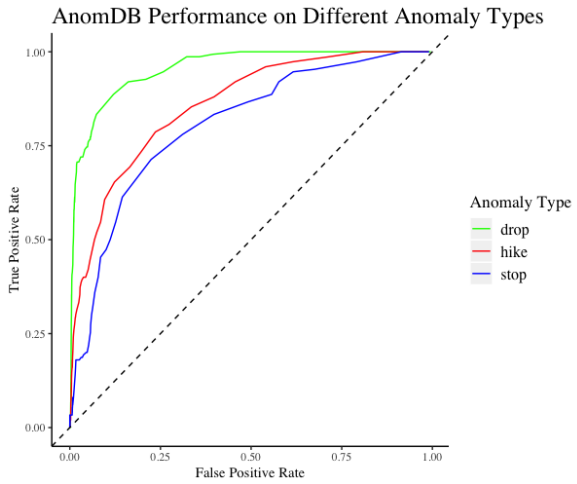

Figure 7. ROC Curve for *AnomDB* performing on different types of anomalies, averaged over considered machine types.

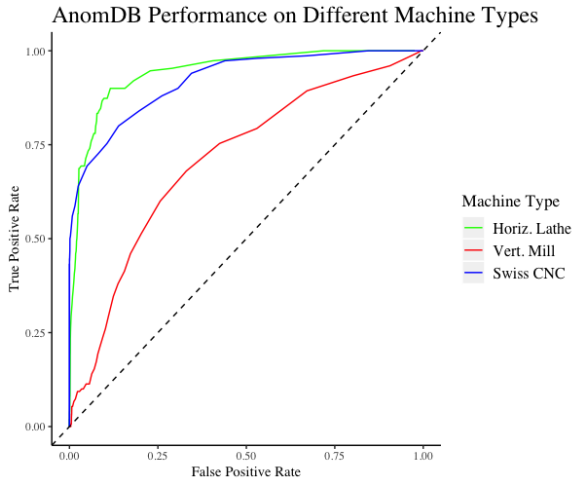

Figure 8. ROC Curve for *AnomDB* performing on different CNC machine types, averaged over considered anomalies.

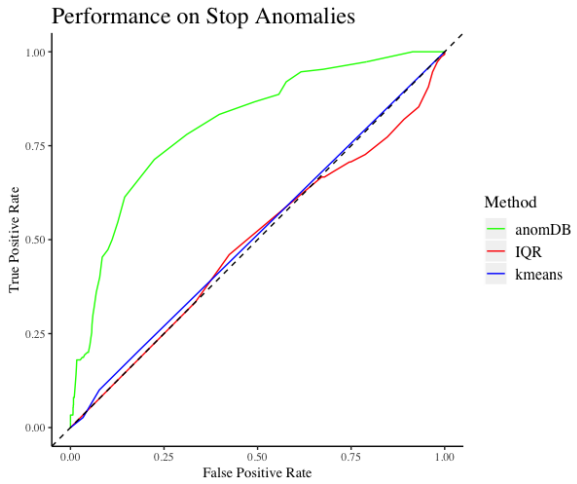

Figure 10. ROC Curve comparing the classifiers *AnomDB*, k-means--, and IQR on 'stop' synthetic anomalies.

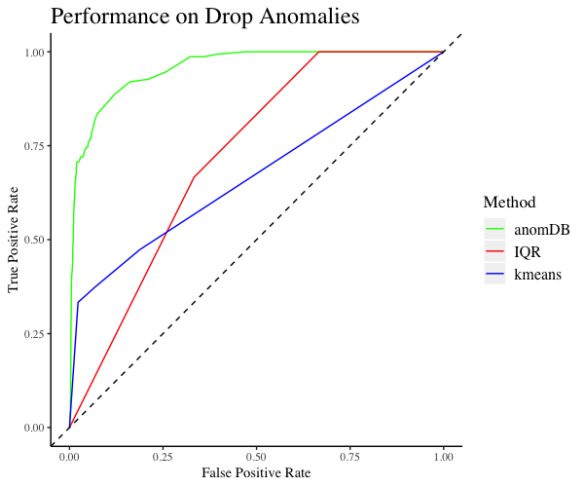

Figure 9. ROC Curve comparing the classifiers *AnomDB*, k-means--, and IQR on 'drop' synthetic anomalies.

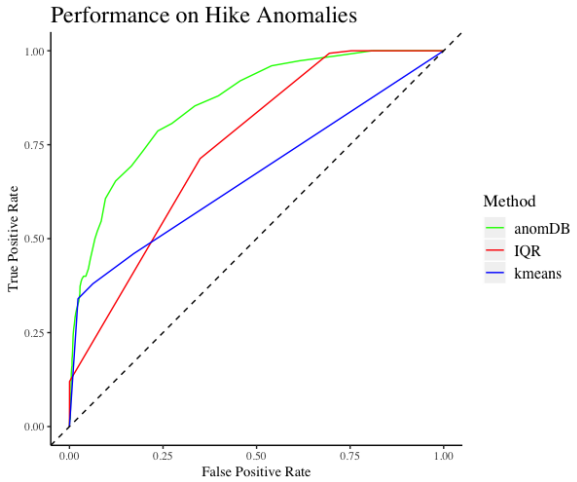

Figure 11. ROC Curve comparing the classifiers *AnomDB*, k-means--, and IQR on 'hike' synthetic anomalies.

In the ROC curve showing the performance of *AnomDB* at categorizing synthetic anomalies on different machine types 8, the curves for horizontal lathes and Swiss CNCs closely adhere to the left hand border and top border which can be interpreted as evidence of *AnomDB*'s high performance for these machine types. Figure 7 also indicates that *AnomDB* has a high discrimination for time series containing drop synthetic anomalies. Using the Area Under the Curve (AUC) as a measure of diagnostic accuracy, *AnomDB* overall outperforms k-means-- and IQR with an AUC of 86.92 %, as compared with k-means-- with an AUC of 61.65 % and IQR with an AUC of 64.34 %.

We further show that across different types of machines and synthetic anomalies, *AnomDB* outperforms other methods significantly and has a more consistent ROC curve. This is important in demonstrating its power as a general-purpose

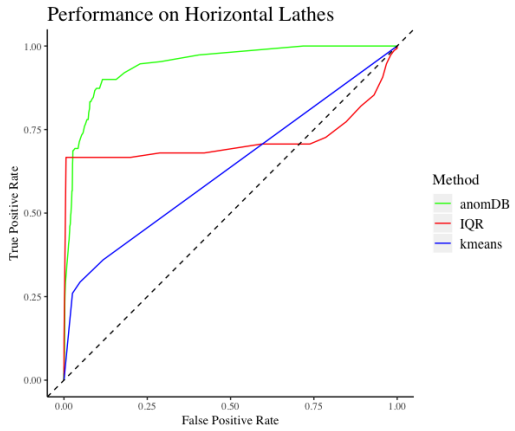

Figure 12. ROC Curve comparing the classifiers *AnomDB*, k-means--, and IQR on horizontal lathe time series.

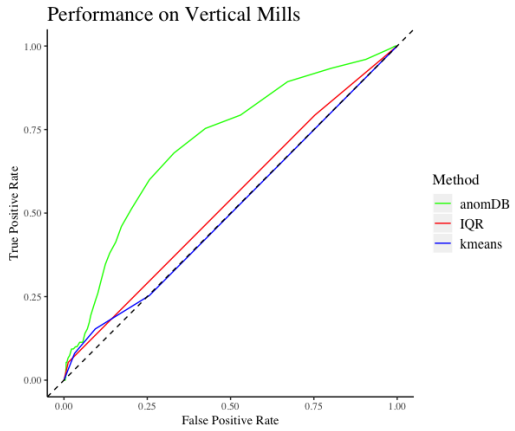

Figure 13. ROC Curve comparing the classifiers *AnomDB*, k-means--, and IQR on vertical mill time series.

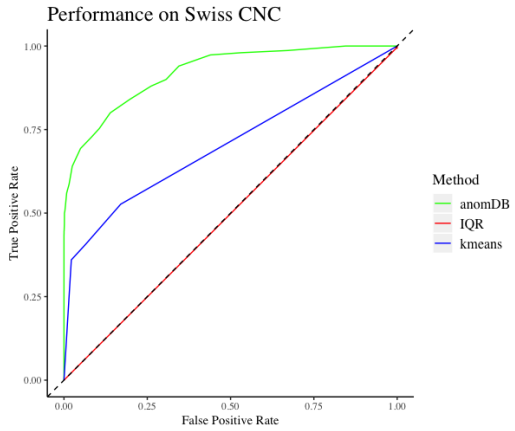

Figure 14. ROC Curve comparing the classifiers *AnomDB*, k-means--, and IQR on Swiss CNC time series.

anomaly-detection algorithm on CNC machine control data. These ROC curves can be seen in Figures 9–14. Particularly in the cases of 'stop' synthetic anomalies, k-means-- and IQR as shown in Figure 6 come close to the diagonal line in the ROC space, indicating a lower overall accuracy that performs comparably with random selection, whereas *AnomDB* has significantly higher discrimination ability.

5. DISCUSSION

We propose *AnomDB* as a novel algorithm for detection of anomalous part cycles in CNC machines using their native control data in a live production setting. This approach, inspired by (Hyndman et al., 2015), can effectively determine certain broad classes of anomalous machine behavior using a density-based outlier strategy, while eliminating many of the false positives associated with simple alternative approaches that we explored based on IQR and k-means clustering. Specifically, time series patterns typical of CNC machines may benefit from this technique due to its ability to combine distinct, noisy, and possibly under-sampled control data streams.

Our initial work toward this goal leaves a number of interesting open issues. A major one is in how anomalous part signatures identified by *AnomDB* correlate with specific machine events or can provide information about tool/machine issues or data stream issues. Further studies along these lines will facilitate the transformation of online condition monitoring into actionable information for intelligent preventative maintenance.

ACKNOWLEDGMENT

This research was supported by MachineMetrics. We thank our colleagues from MachineMetrics for guidance and insight that enabled this research.

REFERENCES

Chawla, S., & Gionis, A. (2013). k-means-: A unified approach to clustering and outlier detection. In *Sdm*.

Chen, S.-L., & Jen, Y. (2000). Data fusion neural network for tool condition monitoring in cnc milling machining. *International Journal of Machine Tools and Manufacture*, *40*(3), 381 - 400. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0890695599000668` doi: https://doi.org/10.1016/S0890-6955(99)00066-8

Dancho, M., & Vaughan, D. (2018). anomalize: Tidy anomaly detection [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=anomalize` (R package version 0.1.1)

Das, M. K., & Dai, H.-K. (2007). A survey of dna motif finding algorithms. In *Bmc bioinformatics*.

Duan, C., Makis, V., & Deng, C. (2019). Optimal bayesian early fault detection for cnc equipment using hidden semi-markov process. *Mechanical Systems and Signal Processing*, *122*, 290 - 306. Retrieved from http://www.sciencedirect.com/science/article/pii/S088832701830760X doi: https://doi.org/10.1016/j.ymssp.2018.11.040

Ertekin, Y. M., Kwon, Y., & Tseng, T.-L. B. (2003). Identification of common sensory features for the control of cnc milling operations under varying cutting conditions. *International Journal of Machine Tools and Manufacture*, *43*(9), 897 - 904. Retrieved from http://www.sciencedirect.com/science/article/pii/S0890695503000877 doi: https://doi.org/10.1016/S0890-6955(03)00087-7

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the second international conference on knowledge discovery and data mining* (pp. 226–231). AAAI Press. Retrieved from http://dl.acm.org/citation.cfm?id=3001460.3001507

Hahsler, M., & Piekenbrock, M. (2018). dbscan: Density based clustering of applications with noise (dbscan) and related algorithms [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=dbscan (R package version 1.1-3)

H. Atluru, S., & Deshpande, A. (2009, 01). Data to information: Can mtconnect deliver the promise?

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, *24*(6), 417–441. Retrieved from https://doi.org/10.10372Fh0071325 doi: 10.1037/h0071325

Howe, D. C. (2015). kmodr: K-means with simultaneous outlier detection [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=kmodR (R package version 0.1.0)

Hyndman, R. J., Wang, E., & Laptev, N. (2015, Nov). Large-scale unusual time series detection. In *2015 ieee international conference on data mining workshop (icdmw)* (p. 1616-1619). doi: 10.1109/ICDMW.2015.104

Hyndman, R. J., Wang, E., & Laptev, N. (2019). anomalous: Unusual time series detection [Computer software manual]. (R package version 0.1.0)

Jinka, P. (2015). *Anomaly detection for monitoring: A statistical approach to time series anomaly detection*. O'Reilly Media. Retrieved from https://books.google.com/books?id=DnnuuQEACAAJ

Kang, Y., Hyndman, R. J., & Smith-Miles, K. (2017). Visualising forecasting algorithm performance using time series instance spaces. *International Journal of Forecasting*, *33*(2), 345 - 358. Retrieved from http://www.sciencedirect.com/science/article/pii/S0169207016301030 doi: https://doi.org/10.1016/j.ijforecast.2016.09.004

Kumar, S., Nassehi, A., Newman, S. T., Allen, R. D., & Tiwari, M. K. (2007). Process control in cnc manufacturing for discrete components: A step-nc compliant framework. *Robotics and Computer-Integrated Manufacturing*, *23*(6), 667 - 676.

Litjens, G., Kooi, T., Ehteshami Bejnordi, B., Setio, A., Ciompi, F., Ghafoorian, M., ... I. Snchez, C. (2017, 02). A survey on deep learning in medical image analysis. *Medical Image Analysis*, *42*. doi: 10.1016/j.media.2017.07.005

Louhichi, S., Gzara, M., & Abdallah, H. B. (2014, Jan). A density based algorithm for discovering clusters with varied density. In *2014 world congress on computer applications and information systems (wccais)* (p. 1-6). doi: 10.1109/WCCAIS.2014.6916622

Ray, S., & Turi, R. H. (1999). Determination of number of clusters in k-means clustering and application in colour segmentation. In *The 4th international conference on advances in pattern recognition and digital techniques* (pp. 137–143).

Recknagel, F. (2001). Applications of machine learning to ecological modelling. *Ecological Modelling*, *146*(1), 303 - 310. Retrieved from http://www.sciencedirect.com/science/article/pii/S0304380001003167 doi: https://doi.org/10.1016/S0304-3800(01)00316-7

Solan, Z., Horn, D., Ruppin, E., & Edelman, S. (2005). Unsupervised learning of natural languages. *Proceedings of the National Academy of Sciences*, *102*(33), 11629–11634. Retrieved from https://www.pnas.org/content/102/33/11629 doi: 10.1073/pnas.0409746102

Tiwari, A., Vergidis, K., Lloyd, R., & Cushen, J. (2008, 10). Automated inspection using database technology within the aerospace industry. *Proceedings of The Institution of Mechanical Engineers Part B-journal of Engineering Manufacture - PROC INST MECH ENG B-J ENG MA*, *222*, 175-183. doi: 10.1243/09544054JEM938

Zhang, C., Song, D., Chen, Y., Feng, X., Lumezanu, C., Cheng, W., ... Chawla, N. V. (2018, November). A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data. *arXiv e-prints*, arXiv:1811.08055.

BIOGRAPHIES

Lou Zhang is the Lead Data Scientist at MachineMetrics, Inc. in Northampton, MA, where he develops applications of predictive analytics and machine learning for CNC machine tools. Previously, he worked as a data analyst in finance and econometrics at IHS Markit. He received his B.B.A. degree in Finance from the College of William and Mary in Virginia in 2012. His research interests include time-series analysis and applied machine learning for both finance and manufacturing.

Sarah Elghazoly is a Data Science Intern at MachineMetrics, Inc. She received her B.A. in Physics at Smith College in Northampton, MA in 2019. Her research interests include signal processing and quantum thermodynamics.

Brock Tweedie is a Data Science Intern at MachineMetrics, Inc. He received his Ph.D. in theoretical particle physics from the University of California, Berkeley, in 2007. His post-doctoral physics work was focused on developing strategies for the discovery and characterization of new particles at the CERN Large Hadron Collider. His current work is in private-sector applications of data science.