Denoising with Singular Value Decomposition for Phase Identification in Power Distribution Systems

Nicholas Zaragoza and Vittal Rao, *Senior Member, IEEE*

Abstract—Phase identification is the problem of determining what phase(s) that a load is connected to in a power distribution system. However, real-world sensor measurements used for phase identification have some level of noise that can hamper the ability to identify phase connections using data-driven methods. Knowing the phase connections is important to keep the distribution system balanced so that parts of the system are not overloaded, which can lead to inefficient operations, accelerated component degradation, and system destruction at worst. We use Singular Value Decomposition (SVD) with the optimal Singular Value Hard Threshold (SVHT) as part of a feature engineering pipeline to denoise data matrices of voltage magnitude measurements. This approach reduces Frobenius error and increases the average phase identification accuracy over a year of time series data. K-medoids clustering is used on the denoised voltage magnitude measurements to perform phase identification.

Impact Statement—Phase identification is a popular problem to apply data-driven methods in power distribution systems. These methods allow phase identification to be performed without the costly requirements of installing specialized hardware across the distribution system or using unscalable human labor to visually inspect all electrical connections. However, data from smart meters are inherently noisy and can reduce the ability of phase identification algorithms to be effective. In this work, we present a data-driven approach to reduce the amount of noise in smart meter data. In some cases, this approach can increase the average accuracy of phase identification from 45.47% to 98.92% using a simple phase identification algorithm. It can enable electric utility companies to narrow the trade-off gap between the cost and accuracy of smart meters installed in a power distribution system.

Index Terms—SVD, SVHT, machine learning, smart grid, AMI, phase identification

I. INTRODUCTION

Phase identification is the act of determining what phase of a multi-phase power distribution system that a load (such as a home, business, etc.) is connected to. Knowing the phase connectivity of loads is important for keeping a distribution system operating under balanced conditions, where the amount of power flowing on each phase is as close to equal as possible [1]. An unbalanced system can result in issues such as lower operating efficiency, component degradation (such as that of secondary distribution transformers), and overheating [2] [3], to name a few. However, maintaining accurate records of phase connections can be made difficult due to events such as maintenance, system reconfigurations, emergency power restorations (such as due to extreme weather events), and potential cyberattacks [4] [5].

As of 2018, 86.8 million out of a total of 154.1 million electric meters installed in the United States are Advanced Metering Infrastructure (AMI) meters, according to data from

the Energy Information Administration (EIA) [6] [7]. As AMI meter coverage increases, more streams of data will become available for use in data-driven methods. Unfortunately, meters are inherently erroneous and are available at different meter classes that ensure a measurement is at least within some percentage of the true underlying value [8]. As smart meters age, the resulting degradation will lead to an increase in meter error, which can require recalibration or replacement every few years [9]. This inherent error or noise can negatively affect the ability of data-driven methods to effectively perform phase identification.

Phase identification can involve either directly assigning phases to loads [10] or correcting any incorrect labels in the records [11] (Section II). We evaluate the effect of denoising with SVD on the problem of phase label correction using a synthetic dataset. The synthetic dataset (Section III) is composed of voltage magnitude measurements, sampled at 15-minute-intervals for an entire year, generated from the Electric Power Research Institute's (EPRI) ckt5 test feeder circuit [12]. We then present a feature engineering pipeline (Section IV) that incorporates Singular Value Decomposition (SVD) with the Singular Value Hard Threshold (SVHT) [13] to denoise voltage magnitude measurements for phase identification. To perform phase-label correction, we use the k-medoids clustering [14] algorithm with a majority-vote approach (Section V). Our simulation results (Section VI) show a reduction in Frobenius error and a corresponding increase in phase identification accuracy for different levels of injected Gaussian white noise. Though SVD has been used for various problems in power distribution systems in the past (Section VII), this is the first instance of SVD with the SVHT being used for directly denoising measurements in the phase identification literature.

II. PROBLEM FORMULATION

The goal of phase identification is to determine the phase(s) a load is connected to in a power distribution system. Phase identification can involve either assigning the labels directly (such as by using substation data [10]) or label correction (when the labels are available but some of them are incorrect [11]). In this work, we tackle the label correction problem. Label correction can be performed by first grouping loads that are similar to each other based on some pairwise distance. We can accomplish this grouping using a clustering algorithm. After the loads have been clustered, their labels can be corrected using a majority vote rule. For example, if most of the loads in a cluster are labeled 'Phase A', then all loads in the cluster will be given the label of 'Phase A'.

As long as most of the available labels within a cluster are correct, then any loads in the cluster with a wrong label will be corrected. However, such an approach is still speculative in nature and the actual phase connections would have to be physically inspected to verify the labeling [15].

III. SYNTHETIC DATASET

A. Electric Power Research Institute (EPRI) *ckt5* Test Circuit

The *ckt5* [12] test feeder circuit, made available by the Electric Power Research Institute (EPRI), is used to generate voltage magnitude measurements via their OpenDSS [16] software. The circuit is composed of a single feeder that feeds a radial network of single-phase loads, as can be seen in figure 1. The distribution of the different load classes and secondary distribution transformers is shown in Table I.

Table I: *ckt5* Load Distribution

Load Class	Load Count	Transformer Count
Commercial: Medium	3	2
Commercial: Small	72	34
Residential	1304	555
Total	1379	591

B. Synthetic Load Profiles

The *ckt5* circuit provides a single yearly real power load profile, sampled at 1-hour intervals (8760 timesteps) for each load class, as shown in figure 2. The period of the seasonality in each load profile is 1-day. Since each load in the circuit is defined with a constant real power value, the OpenDSS software will vary this value over time by the percentage indicated in the profiles. Since we want to work with 15-minute-samples, linear interpolation is used to convert the 1-hour-sampled profiles to 15-minute-sampled profiles (35040 timesteps). Ideally, we want each load to follow a unique load profile with its own variations. To synthesize a unique profile for each load, we simply inject Gaussian white noise to simulate variation in power consumption, as shown in equation 1.

$$p_t = d_t + N(0, 0.1) \quad (1)$$

d_t is the default load profile value at time t for each respective profile. The previous linear interpolation step should be performed before injecting variation. This approach allows us to maintain the yearly trend and daily seasonality of the original load profiles. The choice of $\sigma = 0.1$ is somewhat arbitrary and makes the phase-identification problem slightly more difficult since the variations are smaller. Other values, such as $\sigma = 0.3$ and $\sigma = 0.5$, have been tried and result in slightly higher phase-identification accuracies in some cases with the methods used in this work.

C. Voltage Magnitude Measurements

The OpenDSS software is used to run a Quasi-Static Time Series (QSTS) simulation of the *ckt5* circuit using the synthetic load profiles to generate voltage magnitude measurements for each load. This gives us a 35040×1379 matrix $\mathbf{X}$ of voltage

Figure 1: The *ckt5* test feeder circuit from the EPRI. Each of the small red diamonds represents a secondary distribution transformer. The large red triangle at the bottom right is the primary substation transformer supplying the feeder. The thickness of the blue lines indicate the amount of power flow.

magnitude measurements, where each of its 35040×1 column vectors $\tilde{x}_i$ is the time series of load i . To simulate measurement noise in the voltage magnitudes, we inject Gaussian white noise into each element x_{mn} of $\mathbf{X}$, as shown in equation 2.

$$x_{mn}^n = x_{mn} + N\left(0, \frac{\nu x_{mn}}{3}\right) \quad (2)$$

We denote the noisy measurements by the 35040×1379 matrix $\mathbf{X}^n$. The choice of $3\sigma = \nu x_{mn}$ ensures that there is a 99.7% probability that a noisy sample will be within ν percent of the original x_{mn} . A similar noise model has been used before [17] and allows us to model the noise as Gaussian.

IV. FEATURE ENGINEERING

A. Stationarity: Second Order Time Difference

Arya et al. [11] have shown that the Pearson correlation coefficient is a good measure for capturing the linear relationship between the voltage magnitude measurements of loads connected to the same phase. Before estimating the Pearson correlation coefficient, we need to ensure that each time series is stationary. When a time series is non-stationary with serial correlation, the estimation of classical summary

Figure 2: The three default load profiles packaged with the ckt5 test feeder circuit. Note that they are in the range $[0, 1]$, which indicates the percent by which to vary a load.

statistics such as the sample mean, variance, and covariance between two time series become meaningless [18]. We use the notion of weak stationarity, which only requires that the mean of the time series is constant and that its autocovariance function only depends on the distance between time lags rather than on time itself (constant mean, constant variance, and no seasonality) [18] [19]. It should be clear from the default load profiles in figure 2 that the resulting voltage measurements obtained from simulation will not be stationary (there is obvious trend and seasonality). Below are some methods that have been used in the phase identification literature to remove trend and seasonality:

- Mitra et al. [20] have used the first-order time difference of each time series.
- Xu et al. [21] apply the Discrete Fourier Transform (DFT) to each time series. They zero out the low-frequency components and inverse DFT the series back to the time domain.
- Hosseini et al. [3] use a Finite Impulse Response (FIR) high-pass filter to filter out the low frequencies in each time series.

In this work, we choose to use the time difference transformation, as it is a non-parametric technique that is computationally fast (runs in linear time) and acts as a high-pass filter [18].

Before looking at the time difference transformation, we need to define the backshift and difference operators [18].

Definition 1: The k th order **backshift operator** B^k can be defined as

$$B^k x_t = x_{t-k} \quad (3)$$

where k is the number of timesteps to lag the input x_t .

Definition 2: The d th order **difference operator** ∇^d can be defined as

$$\nabla^d = (1 - B)^d \quad (4)$$

With the synthetic dataset used in this work, we only consider the first (equation 5) and second (equation 6) order time-difference transformations.

$$\nabla x_t = (1 - B)x_t = x_t - x_{t-1} \quad (5)$$

$$\nabla^2 x_t = (1 - B)^2 x_t = x_t - 2x_{t-1} + x_{t-2} \quad (6)$$

We use $\nabla \vec{x}_i$ and $\nabla^2 \vec{x}_i$ to indicate the first-order and second-order time difference of the voltage time series of load i . Each order of time difference will reduce the number of timesteps

in the original time series by 1. $\nabla \mathbf{X}$ and $\nabla^2 \mathbf{X}$ indicate the respective time differences applied to all column vectors of $\mathbf{X}$.

To observe the effects of the time difference orders, we look at a 7-day window of data starting at the beginning of the year, which gives us a 672×1379 matrix $\mathbf{W}$. Figure 3 shows the plot of a single load $\vec{w}_{1337}$, its autocorrelation function (ACF) with 96-time-lags (the period of seasonality is one day or 96 timesteps for the 15-minute-samples), and a heatmap of the correlation matrix of $\mathbf{W}$ for different time-difference orders using the Pearson correlation coefficient. Since our goal isn't to fit a time-series model to $\vec{w}_{1337}$, we don't show the partial autocorrelation function (PACF) plot. In the first column of figure 3, the gradual shrink/growth of $\text{ACF}(\vec{w}_{1337})$ is characteristic of an underlying trend and the periodic movement indicates the presence of seasonality (this is evident in figure 3a). Due to this non-stationarity, it can be seen that no distinctive structure can be observed by the Pearson correlation coefficient, as shown in figure 3g. In the second column of figure 3, the first-order difference causes the underlying correlation structure to begin to appear. However, $\text{ACF}(\nabla \vec{w}_{1337})$ shows significant serial correlation is still present and periodic in nature. The plot of $\nabla \vec{w}_{1337}$ in figure 3b shows that a seasonal component still exists. In the third column of figure 3, the correlation structure is much more distinct with the second-order difference. It can be seen that $\nabla^2 \vec{w}_{1337}$ looks closely like white noise and that most of the serial correlation has been removed. However, there is still a significant negative peak in $\text{ACF}(\nabla^2 \vec{w}_{1337})$ at the first-time lag, with a sudden drop afterwards. This is indicative of a possible underlying Moving Average MA(1) process, which is itself stationary. A second-order time difference indicates that the underlying trend of $\vec{w}_{1337}$ (figure 3a) within the 7-day window was quadratic rather than linear (in the case of a first-order difference). Going forward, we will utilize the second-order time difference.

B. Singular Value Decomposition (SVD)

The Singular Value Decomposition (SVD) [22] of an $M \times N$ matrix $\mathbf{A}$ is shown in equation 7.

$$\mathbf{A} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T = \sum_{i=1}^N \sigma_i \vec{u}_i \vec{v}_i^T = \sum_{i=1}^N \sigma_i \mathbf{P}_i \quad (7)$$

Figure 3: Each column contains a time series plot of $\vec{w}_{1337}$, its ACF for 96-timesteps (the period of seasonality is one day or 96-timesteps for the 15-minute-samples), and a heatmap of $\text{corr}(\mathbf{W})$ for each order of time difference. In the heatmaps, the loads are sorted and grouped by their true phase, with the red dashed lines separating the groups. A 7-day window starting at the beginning of the dataset is used.

$\mathbf{U}$ is the $M \times M$ matrix of left-singular vectors. $\mathbf{\Sigma}$ is the $M \times N$ matrix of zeros on the off-diagonals and r singular values along the main diagonal where $r = \text{rank}(\mathbf{A})$. $\mathbf{V}$ is the $N \times N$ matrix of right-singular vectors. The SVD of $\mathbf{A}$ can be broken down into a linear combination of the singular values σ_i times the outer product of the respective left-singular-column vector $\vec{u}_i$ and right-singular-column vector $\vec{v}_i$. This in turn gives us a linear combination of $M \times N$ rank-one matrices $\mathbf{P}_i$ weighted by σ_i . For practical reasons, we don't calculate the entirety of the SVD, but rather we use the economy SVD [22], as shown in equation 8.

$$\mathbf{A} = \hat{\mathbf{U}} \hat{\mathbf{\Sigma}} \mathbf{V}^T = \sum_{i=1}^r \sigma_i \vec{u}_i \vec{v}_i^T = \sum_{i=1}^r \sigma_i \mathbf{P}_i \quad (8)$$

$\hat{\mathbf{U}}$ is the $M \times r$ matrix of left-singular vectors. $\hat{\mathbf{\Sigma}}$ is the first diagonal $r \times r$ block of $\mathbf{\Sigma}$. Since there can only ever be r singular values, there is no reason to burn CPU cycles and waste memory on calculating the other $M - r$ left singular vectors.

The singular values in $\hat{\mathbf{\Sigma}}$ will be in descending order, where the value of σ_i indicates the importance / amount of information contained within $\vec{u}_i$ and $\vec{v}_i$. The idea behind

denoising using SVD is to determine the number of singular values to keep to only retain the underlying information while discarding the noise. We use the optimal Singular Value Hard Threshold (SVHT) with unknown noise level, first proposed by Gavish et al. [13], to determine the number of singular values to keep, as shown in equation 9.

$$\tau = \omega(\beta) * \sigma_{\text{median}} \quad (9)$$

σ_{median} is the median singular value. $\omega(\beta)$ is the optimal hard threshold coefficient which can be approximated using equation 10.

$$\omega(\beta) \approx 0.56\beta^3 - 0.95\beta^2 + 1.82\beta + 1.43 \quad (10)$$

β is simply the ratio of the number of rows to the number of columns $\beta = M/N$. We only keep the first k singular values where $\sigma_i \geq \tau$ and discard all the others. The denoised matrix $\mathbf{A}^d$ can then be reconstructed using the first k singular values, as shown in equation 11.

$$\mathbf{A}^d = \sum_{i=1}^k \sigma_i \vec{u}_i \vec{v}_i^T = \sum_{i=1}^k \sigma_i \mathbf{P}_i \quad (11)$$

To evaluate how well the denoised matrix approximates the original noiseless data, we use the Frobenius error [22] between two matrices, $\mathbf{A}$ and $\mathbf{B}$, as shown in equation 12.

$$E_F(\mathbf{A}, \mathbf{B}) = \|\mathbf{A} - \mathbf{B}\|_F \quad (12)$$

$\|\cdot\|_F$ is the Frobenius norm of a matrix.

In figure 4, we examine the effectiveness of the denoising process by looking at a 7-day window starting from the beginning of the year with a second-order time difference transformation for different noise levels using the noise model in equation 2. The noise levels correspond to the ANSI C12.20-2015 [8] 0.1, 0.2, and 0.5 -meter classes, as well as a class-1 meter. In the first row of figure 4, the correlation structure begins to fade away as the noise level increases. Any phase-identification algorithm that relies on exploiting the correlations between the voltage magnitude measurements of loads would essentially be neutralized, starting at $\nu = 0.5\%$, for a 7-day window. This fading structure behavior has also been observed by Modarresi et al. [23]. In the second row of figure 4, we can see that the SVD and SVHT can recover a good approximation of the original correlation structure for $\nu = 0.1\%$, $\nu = 0.2\%$, and $\nu = 0.5\%$. However, we can observe a limitation of the SVHT with the approximation of $\omega(\beta)$ for $\nu = 1\%$. The SVHT determined that only the first three singular values should be kept, but, as can be seen in figure 4i, the correlation structure isn't recovered in any meaningful manner, despite a significant decrease in frobenius error (783.77, down to 179.88). In figure 4j, we can recover a moderate amount of the structure by manually selecting $k = 5$ which is just past the elbow in figure 4o.

C. Pairwise Distances

The zero-time lagged sample cross-correlation of two jointly stationary time series simplifies to the sample Pearson correlation coefficient, as shown in equation 13 [18].

$$r_{xy} = \frac{\sum_{t=1}^n (x_t - \bar{x})(y_t - \bar{y})}{\sqrt{\sum_{t=1}^n (x_t - \bar{x})^2} \sqrt{\sum_{t=1}^n (y_t - \bar{y})^2}} \quad (13)$$

$\bar{x}$ and $\bar{y}$ are the sample means of the two stationary time series $\vec{x}$ and $\vec{y}$, respectively. n is the number of timesteps in the window used to estimate the correlation. Once the correlation is calculated, it is converted into a measure of distance by using the *Rooted Normalized One-Minus-Correlation* [24], as shown in equation 14.

$$d_{xy}^c = \sqrt{\frac{1 - r_{xy}}{2}} \quad (14)$$

This gives us a distance in the range $[0, 1]$. We can then calculate a 1379×1379 symmetric distance matrix $\mathbf{D}^c$, where each element of the matrix is calculated from d_{xy}^c .

D. Pipeline

Figure 5 shows the feature engineering pipeline that describes how the data should be preprocessed. The SVD of a window of noisy data $\mathbf{W}^n$ will be computed and the denoised matrix $\mathbf{W}^d$ will be reconstructed (equation 11) using the number of singular values determined by the SVHT. Once $\mathbf{W}^d$

is built, the second-order time difference is applied to each time series (column vector) in $\mathbf{W}^d$ to make them stationary.

An important thing to note in equation 9 is that the SVHT is a data-driven threshold, since it depends on the median of the singular values. This means that the SVHT can be used as part of an automated process to automatically select the number of singular values for a given context, without requiring human intervention. The number of singular values kept will be specifically tailored for each window $\mathbf{W}^n$ that is fed into the pipeline.

V. CLUSTERING

Since we have some domain knowledge regarding the dataset (3-phase-system with single-phase loads and 519 transformers), we know that we should at least be looking for three clusters for phase identification [11] in the dataset. For phase identification, we use k-medoids clustering with the d_{xy}^c pairwise distance to find three clusters.

A. K-Medoids Clustering

K-medoids clustering [14] is a partitioning algorithm that uses an actual observation in the data as the center or medoid, of each cluster. This is opposed to k-means clustering [25], which uses the mean of observations in a cluster as the center or centroid. Because of this, k-medoids clustering is more robust to outliers than k-means clustering. K-medoids clustering can also be used with any precomputed arbitrary pairwise distance. We use the 'alternate' k-medoids algorithm by Park et al [26] that is implemented in the software package [27] used. The steps for k-medoids are listed below:

- 1) Step 1: Initialization (via k-means++ [28])
 - a) Randomly (and uniformly) select an observation to be the medoid of the initial cluster.
 - b) Calculate the distance d_{xc} to the cluster c to the closest medoid for each observation x .
 - c) Randomly sample the probability mass function $p_X(x) = \frac{d_{xc}^2}{\sum_x d_{xc}^2}$ to obtain the observation that will be the medoid of the next cluster. The observation furthest from all current cluster medoids will have the highest probability of being selected as the next cluster medoid.
 - d) Repeat from step 1b until k initial clusters have been selected.
- 2) Step 2: Update (via 'alternate' method [26])
 - a) The observation within a cluster that minimizes the distance to all other observations in the cluster is chosen as the new medoid for that cluster.
 - b) Repeat step 2a for all clusters.
- 3) Step 3: Assignment
 - a) Assign each observation to the cluster with the closest medoid.
 - b) If there is no change in the sum of distances of observations to medoids in each cluster, then stop else, go back to step 2a.

Figure 4: From top to bottom, in each column for each noise level contains $\text{corr}(\nabla^2 \mathbf{W}^n)$, $\text{corr}(\nabla^2 \mathbf{W}^d)$, and a log plot of the singular values. Under each heatmap is the respective Frobenius error. The red dashed line in the singular value plots indicate the number of singular values used to reconstruct $\mathbf{W}^d$ as determined by the SVHT (except for the last column where k is selected manually). The same 7-day window as in figure 3 is used.

Figure 5: Feature Engineering Pipeline

VI. SIMULATION RESULTS

A. Setup

All voltage magnitude measurements are time-synchronized and no topology change events are injected into the yearly simulation. Sliding window widths from one to seven days are used. Since smart meters will send measurement data at least daily back to the utility company [7], we use a window stride of one day. This gives $365 - w + 1$ windows of width w in days for the year. We inject noise into the voltage magnitude measurements using $\nu = 0.1\%$, $\nu = 0.2\%$, and $\nu = 0.5\%$, corresponding to the ANSI C12.20-2015 [8] meter accuracy classes of 0.1%, 0.2%, and 0.5%, respectively. We also consider 1% for a class-1 meter.

B. Discussion

Figure 6 shows a strip plot of the accuracies and Frobenius errors for the $365 - w + 1$ windows superimposed on a violin plot. Table II shows the arithmetic mean, median, standard deviation, minimum value, maximum value, skewness, and fisher's kurtosis of the accuracies. Table III shows the same summary statistics for the Frobenius errors.

In the left column of figure 6, we can see that the accuracies for $\nu = 0.1\%$ and $\nu = 0.2\%$ are generally separated

into two distributions for the noisy data. However, once the data is denoised, most of the accuracies fall within a single distribution centered above 95%, with some outliers below 75%. These outliers are primarily due the SVHT selecting too many singular values for reconstruction. This was observed in figure 4k and figure 4l for the 7-day snapshot. For $\nu = 0.5\%$, most accuracies are below 50% for the noisy data. At a window width of four days and above, we see that the distribution of accuracies is centered above 95%, with a single outlier for the 4-day window. Upon examination of the outlier, it is simply due to the randomness when selecting initial medoids. The $\nu = 1\%$ case shows only a minor improvement in accuracy after denoising. However, it is still a dramatic increase compared to the $\nu = 0.5\%$ case. This makes sense, as the number of singular values selected by the SVHT is unable to effectively reconstruct the correlation structure, as shown in figure 4i.

In the right column of figure 6, we can see that the Frobenius errors are quite stable over the year. The center of the error distributions increase as the window width increases. This is because the Frobenius norm sums over all elements of a matrix. A larger matrix will have a larger number of noisy elements to sum. However, despite this increase in error, we still see a relative reduction in error after denoising. An interesting result occurs for $\nu = 0.1\%$, where the denoised error is higher than the noisy error. However, despite this increase in error, we can see in figure 6a that there is not only an increase in average accuracy but also more stable

Table II: Summary Statistics of Accuracies

Window	Mean	Median	Std	Min	Max	Skewness	Kurtosis
$\nu = 0.1\%$ (Noisy / Denoised)							
1 Day	0.7495 / 0.9791	0.7875 / 0.9826	0.1221 / 0.0296	0.3967 / 0.6577	0.9326 / 0.9971	-0.5055 / -9.8132	-0.9594 / 102.0317
2 Day	0.8656 / 0.9917	0.922 / 0.9964	0.1199 / 0.0362	0.4162 / 0.6744	0.9768 / 1.0	-1.3808 / -8.3531	0.4382 / 68.8804
3 Day	0.9162 / 0.9804	0.9608 / 0.9985	0.1082 / 0.0724	0.5997 / 0.6679	0.9898 / 1.0	-1.8919 / -3.9196	1.7492 / 13.4764
4 Day	0.9364 / 0.986	0.9775 / 0.9993	0.1051 / 0.0625	0.6178 / 0.6425	0.9964 / 1.0	-2.1064 / -4.8168	2.5471 / 21.4016
5 Day	0.9612 / 0.9877	0.9869 / 0.9993	0.0845 / 0.059	0.6352 / 0.6432	0.9985 / 1.0	-3.0423 / -5.042	7.4317 / 23.7187
6 Day	0.9671 / 0.9895	0.992 / 1.0	0.0836 / 0.0551	0.6621 / 0.6759	0.9993 / 1.0	-3.1229 / -5.4779	7.9031 / 28.1862
7 Day	0.9648 / 0.9862	0.9949 / 1.0	0.0928 / 0.0637	0.6418 / 0.6476	1.0 / 1.0	-2.7979 / -4.6166	5.9169 / 19.4832
$\nu = 0.2\%$ (Noisy / Denoised)							
1 Day	0.5314 / 0.9512	0.5199 / 0.9529	0.0844 / 0.0138	0.3648 / 0.8883	0.8006 / 0.9797	0.7075 / -0.8618	0.2099 / 1.4835
2 Day	0.6109 / 0.99	0.5961 / 0.9927	0.1037 / 0.0241	0.4017 / 0.6628	0.9036 / 1.0	0.4079 / -12.9513	-0.5571 / 171.0098
3 Day	0.679 / 0.9924	0.6476 / 0.9978	0.1234 / 0.0416	0.4191 / 0.6555	0.9036 / 1.0	0.1258 / -7.6043	-1.2403 / 56.2674
4 Day	0.7334 / 0.9917	0.7603 / 0.9993	0.1319 / 0.0471	0.4387 / 0.6744	0.9282 / 1.0	-0.2056 / -6.5217	-1.4001 / 40.7954
5 Day	0.7909 / 0.9893	0.8615 / 0.9993	0.1302 / 0.0556	0.4271 / 0.6432	0.9536 / 1.0	-0.6576 / -5.4964	-1.1293 / 28.4362
6 Day	0.8201 / 0.9912	0.8956 / 0.9993	0.1292 / 0.0503	0.5149 / 0.6744	0.9616 / 1.0	-0.7353 / -6.1082	-1.2599 / 35.5352
7 Day	0.8439 / 0.9893	0.9159 / 1.0	0.1283 / 0.056	0.5453 / 0.641	0.9674 / 1.0	-0.9446 / -5.4787	-0.9374 / 28.2379
$\nu = 0.5\%$ (Noisy / Denoised)							
1 Day	0.4004 / 0.5074	0.3981 / 0.4982	0.0237 / 0.0416	0.3568 / 0.4242	0.4982 / 0.6998	0.621 / 1.3516	0.4192 / 2.4463
2 Day	0.4087 / 0.7448	0.4061 / 0.7476	0.0256 / 0.0894	0.3582 / 0.4866	0.504 / 0.9028	0.5238 / -0.3475	0.1544 / -0.4352
3 Day	0.422 / 0.9138	0.4199 / 0.9217	0.0321 / 0.0307	0.3568 / 0.7701	0.5352 / 0.9529	0.4473 / -2.5301	0.0598 / 6.9874
4 Day	0.4285 / 0.9541	0.4275 / 0.955	0.0322 / 0.0191	0.3611 / 0.6265	0.5308 / 0.9753	0.3242 / -13.9831	0.018 / 239.3731
5 Day	0.4387 / 0.9727	0.4358 / 0.9732	0.0372 / 0.0062	0.3568 / 0.955	0.5497 / 0.9891	0.4232 / -0.1899	0.0535 / -0.0244
6 Day	0.4473 / 0.983	0.4431 / 0.9833	0.0407 / 0.0043	0.3691 / 0.9674	0.5729 / 0.9935	0.47 / -0.3011	-0.2342 / -0.09
7 Day	0.4547 / 0.9892	0.4511 / 0.9898	0.0408 / 0.0034	0.3568 / 0.9797	0.5874 / 0.9956	0.2622 / -0.3986	-0.2127 / -0.3952
$\nu = 1.0\%$ (Noisy / Denoised)							
1 Day	0.3716 / 0.4569	0.3684 / 0.467	0.0121 / 0.0348	0.3568 / 0.3568	0.4206 / 0.5076	1.0304 / -2.0016	1.0715 / 3.2524
2 Day	0.3722 / 0.4783	0.3698 / 0.4764	0.0118 / 0.0163	0.3568 / 0.4387	0.4141 / 0.5308	0.8556 / 0.5696	0.4167 / 0.3135
3 Day	0.375 / 0.488	0.3727 / 0.483	0.0138 / 0.0228	0.3568 / 0.4525	0.4336 / 0.5547	0.9405 / 0.7924	0.9484 / -0.0618
4 Day	0.3748 / 0.4977	0.3735 / 0.4913	0.0125 / 0.0273	0.3568 / 0.4554	0.4177 / 0.5925	0.6754 / 0.7197	0.1422 / -0.1003
5 Day	0.3778 / 0.5067	0.3764 / 0.5025	0.0142 / 0.0334	0.3568 / 0.4511	0.4235 / 0.612	0.5348 / 0.5482	-0.3511 / -0.4688
6 Day	0.3784 / 0.5151	0.3756 / 0.5098	0.015 / 0.0377	0.3568 / 0.454	0.4307 / 0.6273	0.692 / 0.4162	-0.0082 / -0.8015
7 Day	0.3796 / 0.5313	0.3771 / 0.533	0.0161 / 0.0403	0.3568 / 0.4576	0.4336 / 0.6265	0.7569 / 0.102	0.2525 / -0.9271

performance over the year.

VII. RELATED WORKS

SVD is a classical linear algebra technique that has been used extensively in the power systems literature. We summarize a few recent works that utilize SVD. Wei et al. [29] use SVD to denoise Transient Zero Sequence Currents (TZSC) measurements for detecting fault lines using the euclidean distance between ideal clustering curves. Choqueuse et al. [30] used SVD to determine if a system is unbalanced and to estimate angular frequencies from 3-phase-signals. Zhao et al. [31] used SVD to compress the phasor data from Phasor Measurement Units (PMUs). Gupta et al. [32] have used SVD to show that the ratio correction factor (RCF) for capacitive voltage transformers (CVTs) can be inferred from the first-rank-one approximation of voltage synchrophasor data from PMUs.

Surprisingly, there have not been many attempts in the phase identification literature to directly remove noise from sensor measurements as a preprocessing step. Modarresi et al. [23] model the IEEE 13-bus test feeder using a neural network and use the voltage magnitude measurements that are generated by the network as denoised measurements. The network is composed of a 9-node dense input layer, 10-node dense hidden layer, and an 18-node dense output layer (they don't indicate the activation functions used). They modify the test feeder by adding 18 loads to the system, with six loads on each phase. The low/high voltage and current measurements for all three phases at the substation transformer (nine total measurements)

are used as input to the network. The voltage magnitude measurements from the 18 loads are used as the targets for regression, with noise injected into three of the loads. Their approach requires pre-training the network on historical data and requires retraining in the event of a topology change (however, one could store a bank of pre-trained networks for each topology configuration). They use k-means clustering on the covariance matrix estimated from the voltages generated by the network to perform phase identification. In our approach, we use the voltage magnitude measurements of loads and inject noise into all loads. Our approach also doesn't require pre-training on historical data. Our approach also has no hyperparameters to tune since the selection of singular values is automated by the SVHT.

Many different clustering algorithms have been used in the phase-identification literature, such as k-means clustering [11], k-medoids clustering [20], Gaussian Mixture Model (GMM) [33], Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [15], and spectral clustering [34]. In this work, we chose to use k-medoids clustering for its use with any precomputed arbitrary pairwise distance, robustness to outliers, and simplicity.

VIII. CONCLUSION

We showed that using Singular Value Decomposition (SVD) with the optimal Singular Value Hard Threshold (SVHT) as a component of a feature engineering pipeline is effective for denoising a matrix of voltage magnitude measurements. An average reduction in Frobenius error of over two times

Table III: Summary Statistics of Frobenius Errors

Window	Mean	Median	Std	Min	Max	Skewness	Kurtosis
$\nu = 0.1\%$ (Noisy / Denoised)							
1 Day	29.72 / 55.85	29.75 / 55.88	0.1733 / 0.442	29.21 / 54.71	30.03 / 57.07	-0.5264 / -0.1347	-0.4736 / -0.2931
2 Day	42.03 / 75.35	42.07 / 75.38	0.2346 / 0.493	41.4 / 73.84	42.41 / 76.58	-0.5798 / -0.2511	-0.5258 / -0.1609
3 Day	51.47 / 87.96	51.53 / 87.96	0.2805 / 0.4564	50.76 / 86.46	51.9 / 89.37	-0.611 / -0.0701	-0.5095 / 0.5568
4 Day	59.44 / 97.15	59.51 / 97.15	0.3182 / 0.4783	58.63 / 95.8	59.92 / 98.57	-0.6344 / -0.0996	-0.4825 / -0.0043
5 Day	66.45 / 104.18	66.54 / 104.21	0.3506 / 0.4533	65.55 / 102.84	66.94 / 105.28	-0.6558 / -0.2466	-0.4403 / -0.2614
6 Day	72.8 / 109.52	72.88 / 109.52	0.3794 / 0.4645	71.82 / 107.89	73.32 / 110.88	-0.6747 / -0.0764	-0.3881 / 0.2145
7 Day	78.63 / 113.67	78.7 / 113.73	0.4057 / 0.4806	77.59 / 112.33	79.17 / 114.69	-0.6934 / -0.5134	-0.3398 / -0.0978
$\nu = 0.2\%$ (Noisy / Denoised)							
1 Day	59.44 / 58.12	59.51 / 58.11	0.3466 / 0.3465	58.42 / 57.08	60.07 / 59.23	-0.5264 / 0.0627	-0.4736 / 0.615
2 Day	84.06 / 81.3	84.15 / 81.34	0.4692 / 0.4165	82.8 / 79.88	84.82 / 82.45	-0.5798 / -0.1919	-0.5258 / -0.1741
3 Day	102.95 / 98.14	103.07 / 98.17	0.5611 / 0.4167	101.52 / 96.91	103.8 / 99.16	-0.611 / -0.2075	-0.5095 / -0.3776
4 Day	118.88 / 111.86	119.03 / 111.88	0.6364 / 0.4252	117.26 / 110.49	119.83 / 112.98	-0.6344 / -0.2127	-0.4825 / 0.1474
5 Day	132.91 / 123.46	133.08 / 123.47	0.7012 / 0.4344	131.09 / 122.31	133.88 / 124.44	-0.6558 / -0.2644	-0.4403 / -0.3662
6 Day	145.6 / 133.58	145.76 / 133.61	0.7588 / 0.4649	143.63 / 132.17	146.64 / 134.64	-0.6747 / -0.4321	-0.3881 / -0.0418
7 Day	157.26 / 142.6	157.4 / 142.64	0.8115 / 0.4638	155.18 / 141.17	158.35 / 143.54	-0.6934 / -0.5341	-0.3398 / 0.1353
$\nu = 0.5\%$ (Noisy / Denoised)							
1 Day	148.59 / 66.15	148.77 / 66.17	0.8666 / 0.5636	146.05 / 64.43	150.17 / 67.79	-0.5264 / -0.1673	-0.4736 / 0.1821
2 Day	210.14 / 89.63	210.37 / 89.62	1.1729 / 0.5267	206.99 / 88.14	212.05 / 91.09	-0.5798 / -0.0619	-0.5258 / -0.3092
3 Day	257.37 / 107.16	257.67 / 107.22	1.4026 / 0.5522	253.8 / 105.64	259.5 / 108.4	-0.611 / -0.337	-0.5095 / -0.5282
4 Day	297.19 / 122.19	297.57 / 122.24	1.591 / 0.5332	293.14 / 120.75	299.59 / 123.49	-0.6344 / -0.3575	-0.4825 / -0.2507
5 Day	332.27 / 135.61	332.69 / 135.67	1.7531 / 0.5363	327.73 / 134.2	334.69 / 136.65	-0.6558 / -0.4654	-0.4403 / -0.3124
6 Day	363.99 / 147.82	364.4 / 147.87	1.8969 / 0.5517	359.09 / 146.27	366.6 / 148.86	-0.6747 / -0.5121	-0.3881 / -0.2102
7 Day	393.16 / 159.09	393.5 / 159.14	2.0287 / 0.5645	387.95 / 157.44	395.87 / 160.19	-0.6934 / -0.5596	-0.3398 / -0.0641
$\nu = 1.0\%$ (Noisy / Denoised)							
1 Day	297.18 / 78.28	297.54 / 78.11	1.7332 / 1.7593	292.1 / 74.79	300.34 / 83.16	-0.5264 / 0.4823	-0.4736 / -0.3977
2 Day	420.28 / 102.11	420.75 / 102.17	2.3458 / 1.5504	413.99 / 98.32	424.1 / 105.55	-0.5798 / -0.1089	-0.5258 / -0.8108
3 Day	514.74 / 120.88	515.34 / 120.94	2.8053 / 1.1906	507.61 / 117.77	519.0 / 123.44	-0.611 / -0.2569	-0.5095 / -0.4905
4 Day	594.38 / 136.88	595.14 / 136.92	3.182 / 0.9458	586.28 / 134.24	599.17 / 139.31	-0.6344 / -0.0351	-0.4825 / -0.4195
5 Day	664.55 / 151.03	665.39 / 151.01	3.5062 / 0.8616	655.47 / 148.89	669.39 / 153.22	-0.6558 / 0.1494	-0.4403 / -0.6259
6 Day	727.99 / 163.82	728.81 / 163.81	3.7938 / 0.8001	718.17 / 162.05	733.19 / 166.16	-0.6747 / -0.0355	-0.3881 / -0.5567
7 Day	786.32 / 175.56	787.0 / 175.62	4.0575 / 0.7552	775.89 / 173.67	791.73 / 177.42	-0.6934 / -0.212	-0.3398 / -0.5294

was achieved for window widths ranging from one to seven days for a white Gaussian noise level of $\nu = 0.5\%$. An average phase identification accuracy of 45.47% for noisy data was improved to 98.92% after denoising for a 7-day sliding window over a year of data. K-medoids clustering with $k = 3$ was used to perform phase identification using the *Rooted Normalized One-Minus-Correlation* as a pairwise distance between loads. However, the feature engineering pipeline presented in this work could be adapted to be used with any phase identification algorithm or in some other application, such as topology identification. Some potential future work for the community can be to utilize different approaches for singular value selection, non-linear denoising methods, or applying the presented feature engineering pipeline to different problems in power distribution systems. The synthetic load profiles, dataset, python code, and results to recreate this research is available on GitHub [35].

IX. ACKNOWLEDGEMENT

We gratefully acknowledge the support of the US Department of Education program on the Graduate Assistance in Areas of National Need (GAANN) grant to Texas Tech University. The authors acknowledge the High Performance Computing Center (HPCC) at Texas Tech University for providing computational resources that have contributed to the research results reported within this paper. URL: <http://www.hpcc.ttu.edu> We would also like to acknowledge the following python libraries for making this research possible: numpy [36], pandas [37] [38], pyarrow [39], matplotlib [40], scipy [41],

seaborn [42], scikit-learn [43], scikit-learn-extra [27], and statsmodels [44].

REFERENCES

- [1] S. Liu, X. Cui, Z. Lin, Z. Lian, Z. Lin, F. Wen, Y. Ding, Q. Wang, L. Yang, R. Jin, and H. Qiu, "Practical Method for Mitigating Three-Phase Unbalance Based on Data-Driven User Phase Identification," *IEEE Transactions on Power Systems*, vol. 35, pp. 1653–1656, Mar. 2020. Number: 2 Conference Name: IEEE Transactions on Power Systems.
- [2] C. S. Chen, T. T. Ku, and C. H. Lin, "Design of phase identification system to support three-phase loading balance of distribution feeders," in *2011 IEEE Industrial and Commercial Power Systems Technical Conference*, pp. 1–8, May 2011. ISSN: 2158-4907.
- [3] Z. S. Hosseini, A. Khodaei, and A. Paaso, "Machine Learning-Enabled Distribution Network Phase Identification," *IEEE Transactions on Power Systems*, pp. 1–1, 2020. Conference Name: IEEE Transactions on Power Systems.
- [4] M. H. Wen, R. Arghandeh, A. von Meier, K. Poola, and V. O. Li, "Phase identification in distribution networks with micro-synchrophasors," in *2015 IEEE Power Energy Society General Meeting*, pp. 1–5, July 2015. ISSN: 1932-5517.
- [5] M. Bariya, D. Deka, and A. v. Meier, "Guaranteed Phase Topology Identification in Three Phase Distribution Grids," *IEEE Transactions on Smart Grid*, pp. 1–1, 2021. Conference Name: IEEE Transactions on Smart Grid.
- [6] Energy Information Administration, "Annual electricity power industry report." <https://www.eia.gov/electricity/data/eia861>.
- [7] J. Danly, N. Chatterjee, R. Glick, and A. Clements, "2020 Assessment of Demand Response and Advanced Metering," Dec. 2020.
- [8] L. A. Irwin, "A high accuracy standard for electricity meters," in *IEEE PES T D 2010*, pp. 1–3, Apr. 2010. ISSN: 2160-8563.
- [9] X. Kong, X. Zhang, N. Lu, Y. Ma, and Y. Li, "Online Smart Meter Measurement Error Estimation Based on EKF and LMRLS Method," *IEEE Transactions on Smart Grid*, pp. 1–1, 2021. Conference Name: IEEE Transactions on Smart Grid.

- [10] H. Pezeshki and P. Wolfs, "Correlation based method for phase identification in a three phase LV distribution network," in *2012 22nd Australasian Universities Power Engineering Conference (AUPEC)*, pp. 1–7, Sept. 2012.
- [11] V. Arya and R. Mitra, "Voltage-based clustering to identify connectivity relationships in distribution networks," in *2013 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 7–12, Oct. 2013.
- [12] EPRI, "Electric Power Research Institute (EPRI): Test circuit 'ckt5'." <http://svn.code.sf.net/p/electricdss/code/trunk/Distrib/EPRI/TestCircuits/ckt5/>.
- [13] M. Gavish and D. L. Donoho, "The Optimal Hard Threshold for Singular Values is $\sqrt{3}$," *IEEE Transactions on Information Theory*, vol. 60, pp. 5040–5053, Aug. 2014. Number: 8 Conference Name: IEEE Transactions on Information Theory.
- [14] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*. Wiley, 1990.
- [15] Z. L. Wenyu Wang, Nanpeng Yu, "Advanced metering infrastructure data driven phase identification in smart grid," *GREEN 2017: The Second International Conference on Green Communications, Computing and Technologies*, Sept. 2017.
- [16] EPRI, "OpenDSS." <https://sourceforge.net/projects/electricdss/>.
- [17] W. Wang and N. Yu, "Maximum Marginal Likelihood Estimation of Phase Connections in Power Distribution Systems," *IEEE Transactions on Power Systems*, vol. 35, pp. 3906–3917, Sept. 2020. Number: 5 Conference Name: IEEE Transactions on Power Systems.
- [18] R. Shumway and D. Stoffer, *Time Series Analysis and Its Applications: With R Examples*. Springer Texts in Statistics, Springer International Publishing, 2017.
- [19] R. Hyndman and G. Athanasopoulos, "Forecasting: principles and practice, 3rd edition." <https://otexts.com/fpp3>.
- [20] R. Mitra, R. Kota, S. Bandyopadhyay, V. Arya, B. Sullivan, R. Mueller, H. Storey, and G. Labut, "Voltage Correlations in Smart Meter Data," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15*, (New York, NY, USA), pp. 1999–2008, Association for Computing Machinery, Aug. 2015.
- [21] M. Xu, R. Li, and F. Li, "Phase Identification With Incomplete Data," *IEEE Transactions on Smart Grid*, vol. 9, pp. 2777–2785, July 2018. Number: 4 Conference Name: IEEE Transactions on Smart Grid.
- [22] S. Brunton and J. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019.
- [23] M. S. Modarresi, T. Huang, H. Ming, and L. Xie, "Robust phase detection in distribution systems," in *2017 IEEE Texas Power and Energy Conference (TPEC)*, pp. 1–5, Feb. 2017.
- [24] P. P. Rodrigues, J. Gama, and J. P. Pedroso, "ODAC: Hierarchical Clustering of Time Series Data Streams," in *Proceedings of the 2006 SIAM International Conference on Data Mining*, Proceedings, pp. 499–503, Society for Industrial and Applied Mathematics, Apr. 2006.
- [25] J. Macqueen, "Some methods for classification and analysis of multivariate observations," in *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, 1967.
- [26] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for K-medoids clustering," *Expert Systems with Applications*, vol. 36, pp. 3336–3341, Mar. 2009.
- [27] Scikit-Learn-Contrib, "scikit-learn-contrib/scikit-learn-extra." <https://github.com/scikit-learn-contrib/scikit-learn-extra>.
- [28] D. Arthur and S. Vassilvitskii, "K-means++: the advantages of careful seeding," in *In Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2007.
- [29] X. Wei, B. Wen, D. Yang, B. Wang, X. Wang, X. Yang, and J. Gao, "Fault line detection method based on the improved SVD de-noising and ideal clustering curve for distribution networks," *IET Science, Measurement & Technology*, vol. 12, pp. 262–270, Nov. 2017. Publisher: IET Digital Library.
- [30] V. Choqueuse, P. Granjon, A. Belouchrani, F. Auger, and M. Benbouzid, "Monitoring of Three-Phase Signals Based on Singular-Value Decomposition," *IEEE Transactions on Smart Grid*, vol. 10, pp. 6156–6166, Nov. 2019. Conference Name: IEEE Transactions on Smart Grid.
- [31] J. Zhao, Y. Ye, H. Yu, P. Li, P. Li, J. Wu, and C. Wang, "Improved SVD-based data compression method for synchronous phasor measurement in distribution networks," *International Journal of Electrical Power & Energy Systems*, vol. 129, p. 106877, July 2021.
- [32] P. Gupta and S. A. Soman, "Calibrating CVTs of a Substation From Local PMU Data: An SVD Approach," *IEEE Transactions on Power Systems*, vol. 36, pp. 3362–3371, July 2021. Conference Name: IEEE Transactions on Power Systems.
- [33] F. Ni, J. Q. Liu, F. Wei, C. D. Zhu, and S. X. Xie, "Phase identification in distribution systems by data mining methods," in *2017 IEEE Conference on Energy Internet and Energy System Integration (EI2)*, pp. 1–6, Nov. 2017.
- [34] Y. Ma, X. Fan, R. Tang, P. Duan, Y. Sun, J. Du, and Q. Duan, "Phase Identification of Smart Meters by Spectral Clustering," in *2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2)*, pp. 1–5, Oct. 2018.
- [35] msk-5s, "phase-svd." <https://github.com/msk-5s/phase-svd>.
- [36] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, Sept. 2020.
- [37] Wes McKinney, "Data Structures for Statistical Computing in Python," in *Proceedings of the 9th Python in Science Conference* (Stéfan van der Walt and Jarrod Millman, eds.), pp. 56 – 61, 2010.
- [38] T. pandas development team, "pandas-dev/pandas: Pandas," Feb. 2020.
- [39] "pyarrow." <https://pypi.org/project/pyarrow/>.
- [40] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.
- [41] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, Í. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.
- [42] M. L. Waskom, "seaborn: statistical data visualization," *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, 2021.
- [43] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [44] S. Seabold and J. Perktold, "statsmodels: Econometric and statistical modeling with python," in *9th Python in Science Conference*, 2010.

Nicholas Zaragoza earned a BS in Computer Engineering and MS in Electrical Engineering at Texas Tech University, Lubbock, TX. Currently, a PhD in Electrical Engineering at Texas Tech University is being pursued with research interests in applied Machine Learning.

Vittal Rao received the M.Tech. and Ph.D. degrees in electrical engineering from the Indian Institute of Technology Delhi. He is currently a Professor of electrical and computer engineering with Texas Tech University. He was a Rutledge-Emerson Distinguished Professor of electrical and computer engineering and the Director of Intelligent Systems Center with the University of Missouri-Rolla (UMR). He has also served as the Program Director with National Science Foundation. His research interests include cyber security of smart grid and industrial control systems, and distributed energy management systems. He was a recipient of the IEEE Centennial Medal, the Faculty Excellence Awards at UMR, and the NSF Director Award for Program Management Excellence.

(a) Accuracy $\nu = 0.1\%$ (b) Frobenius Error $\nu = 0.1\%$ (c) Accuracy $\nu = 0.2\%$ (d) Frobenius Error $\nu = 0.2\%$ (e) Accuracy $\nu = 0.5\%$ (f) Frobenius Error $\nu = 0.5\%$ (g) Accuracy $\nu = 1\%$ (h) Frobenius Error $\nu = 1\%$

Figure 6: A strip plot of the accuracies (left column) and errors (right column) of the $365 - w + 1$ windows throughout the year for different window widths w in days. The strip plot is overlaid on top of a violin plot, showing the density estimate of the accuracies and errors.