21, rue d'Artois, F-75008 PARIS
http://www.cigre.org

CIGRE US National Committee
2019 Grid of the Future Symposium

Real-time PMU Data Recovery Application Based on Singular Value Decomposition

A. PINCETI
Arizona State University
USA

D. YANG, K. JONES, K. THOMAS
Dominion Energy, Inc.
USA

SUMMARY

Phasor measurement units (PMUs) allow for the enhancement of power system monitoring and control applications and they will prove even more crucial in the future, as the grid becomes more decentralized and subject to higher uncertainty. Tools that improve PMU data quality and facilitate data analytics workflows are thus needed. In this work, we leverage a previously described algorithm to develop a python application for PMU data recovery. Because of its intrinsic nature, PMU data can be dimensionally reduced using singular value decomposition (SVD). Moreover, the high spatio-temporal correlation can be leveraged to estimate the value of measurements that are missing due to drop-outs. These observations are at the base of the data recovery application described in this work. Extensive testing is performed to study the performance under different data drop-out scenarios, and the results show very high recovery accuracy. Additionally, the application is designed to take advantage of a high performance PMU data platform called PredictiveGrid™, developed by PingThings.

KEYWORDS

Synchrophasor, PMU, data quality, data drop-out, data recovery, singular value decomposition, low rank, matrix completion.

andrea.pinceti@asu.edu

I. INTRODUCTION

Phasor measurement units (PMUs) are considered to be a key technology in the development of the grid in the next few decades. The enhanced monitoring and control capabilities made possible by the increasing number of installed PMUs are crucial to be able to respond to the paradigm shift represented by the move towards renewable resources and higher penetration of distributed generation. Yet, one of the main challenges in the use of PMU data for real world applications is data quality.

In a 2017 NASPI report, the PMU Applications Requirements Task Force (PARTF) defines a framework for the study of the impact of data quality on applications that make use of synchrophasors [1]. Among the many aspects they investigated, data drop-outs are identified as a major issue that plagues PMU data and limits the development of PMU applications. For example, a statistical study on the PMU data collected by Dominion Energy, Inc. shows that one can expect about 1% of data not being reported at any given time. Moreover, the average number of consecutive drop-outs for any given data stream is around 3.5, but in some cases up to 120 consecutive samples might be lost "while the data stream is still considered live and operational" [1]. While advanced applications are being designed to withstand some degree of data drop-out, an accurate and efficient way to recover missing values is crucial for the future of PMU-based monitoring and control applications.

In this framework, successful data recovery can be achieved by leveraging the high correlation between PMU measurements which results in the low rank property of PMU data. Recently, many techniques for low rank matrix completion have been proposed, such as singular value thresholding [2], atomic decomposition [3], and singular value projection [4]. Each of these methods rely on different assumptions and thus they are only suited for specific use cases. In [5], some of these techniques are applied to the problem of PMU data recovery. In addition to comparing the performance of different matrix completion schemes, the authors demonstrate that information cascading matrix completion (ICMC) [6] can be applied to PMU data recovery even when the drop-outs are statistically correlated and not independent of each other. Moreover, an online algorithm for real-time PMU data recovery is proposed. While limited in scope, the numerical results they present are promising and suggest that such an algorithm can be used for real world applications. The authors in [7] present a different algorithm to leverage the low rank property of PMU data for missing value recovery. Their approach is based on an alternating direction method for multipliers which is used to solve the matrix completion problem; while the algorithm is computationally highly efficient, the data recovery results are not as good as those in [5]. Another example of PMU data recovery can be found in [8], where cubic spline interpolation is used to estimate the missing values.

In this paper, we borrow the basic algorithm proposed in [5] to build an online application for PMU data recovery that can be used by utilities on large scale systems. Moreover, we perform thorough performance testing to verify the accuracy of the recovery and an in-depth analysis to assign statistical guarantees on the quality of the recovered data in real time. The application developed and here described can be used for real time data conditioning or offline as necessary for specific studies. It represents an effective and simple tool which greatly improves the usefulness of PMU data in the context of a data analysis workflow. Such a tool, for example, would be crucial for the use of temporal predictive filters for event detection and cybersecurity issues [9]. Previous efforts on PMU data analytics undertaken at Dominion Energy focused on the use of OpenPDC for applications such as voltage control [10][11]. However, the application described in this work is developed in Python and it is tested in a high-performance state-of-the-art cloud-based environment called PredictiveGrid™ by PingThings [12]. The remainder of the paper is organized as follows: in Section II, the mathematical background and problem formulation are presented; in Section III, the practical implementation of the algorithm and the design of the application are described; finally, in Section IV, extensive tests are performed to quantify and validate the data recovery performance.

II. PROBLEM FORMULATION

a. Problem setup

The measurements computed by the PMUs in a system represent time-series data streams which share the same sampling times; for this reason, it is useful to organize such data in matrix form. Assume a total of P individual PMU measurements are collected over T time samples. We define the measurement matrix $\boldsymbol{M}$, where each column indicates one data stream and each row indicates a time sample; thus, $\boldsymbol{M}$ has dimension $T \times P$, as shown by equation $\boldsymbol{M}= \begin{bmatrix} m_{1,1} & \cdots & m_{1,P} \\ \vdots & \ddots & \vdots \\ m_{T,1} & \cdots & m_{T,P} \end{bmatrix} \in \mathbb{R}^{T\times P}$ or $\in \mathbb{C}^{T\times P}$ **Error! Reference source not found.**. It has to be noted that the measurement matrix can store either real values (i.e. magnitudes and angles separately) or complex values in rectangular form (i.e. complex voltages or currents); the data recovery technique described in this work can be applied to both cases.

$$\boldsymbol{M} = \begin{bmatrix} m_{1,1} & \cdots & m_{1,P} \\ \vdots & \ddots & \vdots \\ m_{T,1} & \cdots & m_{T,P} \end{bmatrix} \in \mathbb{R}^{T\times P} \text{ or } \in \mathbb{C}^{T\times P} \tag{1}$$

Given the measurement matrix $\boldsymbol{M}$, the goal of the data recovery algorithm is to estimate any missing measurement(s) from vector $\boldsymbol{m}$, which is the vector of measurements collected at time $T+1$ as illustrated in equation $\boldsymbol{m} = [m_{T+1,1} \quad \cdots \quad m_{T+1,P}] \in \mathbb{R}^{1\times P}$ or $\in \mathbb{C}^{1\times P}$ (2).

$$\boldsymbol{m} = [m_{T+1,1} \quad \cdots \quad m_{T+1,P}] \in \mathbb{R}^{1\times P} \text{ or } \in \mathbb{C}^{1\times P} \tag{2}$$

b. Singular value decomposition

As mentioned in the introduction, it has been shown before that because PMU data streams are time-synchronized and because the physical quantities they measure are closely related to each other (by the power system topology and electrical physical laws), the measurements are highly correlated both in space and time. As a consequence of this observation, a high-dimensional PMU measurement matrix $\boldsymbol{M}$ will be approximately low-rank; that is, the measurements can be approximated in a lower dimensional space with negligible data loss. As shown in the following sections, this fact can be leveraged to efficiently recover missing data from measurement vectors $\boldsymbol{m}$ which are being acquired in real time.

The first step in leveraging the low-rank nature of the measurement matrix $\boldsymbol{M}$ consists in decomposing
i
t

$$\boldsymbol{M}^T = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{3}$$

a
s
where the three new matrices have the following dimensions: $\boldsymbol{U} \in \mathbb{R}^{P\times P}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{P\times T}$, and $\boldsymbol{V}^T \in \mathbb{R}^{T\times T}$. The entries of the diagonal matrix $\boldsymbol{\Sigma}$ resulting from this factorization are called singular values and they can be seen as scaling factors of the orthonormal basis represented by the $\boldsymbol{U}$ and $\boldsymbol{V}$ matrices. The relative size of these singular values is directly related to the rank of the initial matrix. To illustrate this, we collected 98 PMU current magnitudes over 300 samples (10 seconds at a sampling rate of 30 Hz) and performed SVD on the resulting matrix $\boldsymbol{M} \in \mathbb{R}^{300\times 98}$. Figure 1

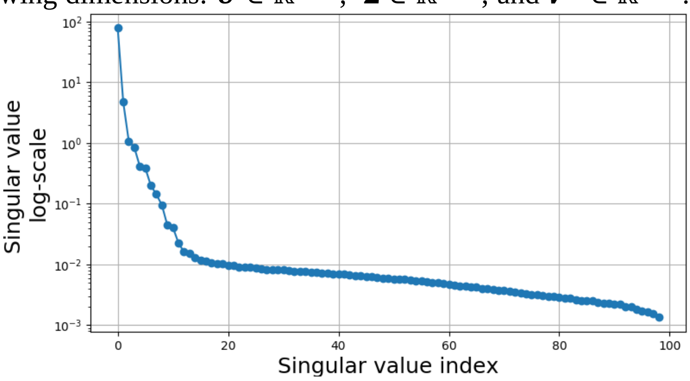


Figure 1: Singular values of a 300×98 matrix.

E
F
_
R
e
f
1

shows the resulting value of the singular values on a log-scale, ranked by size. We can see that the first few singular values are a few orders of magnitude larger than the remaining ones, confirming that there are just a few important principal components: that is, the measurement matrix $\boldsymbol{M}$ can be efficiently reduced to a lower dimensional space with negligible information loss.

Having demonstrated the low-rank nature of a PMU measurement matrix, it is possible to approximate $\boldsymbol{M}$ as $\boldsymbol{M}_r = (\boldsymbol{U}_r \boldsymbol{\Sigma}_\mathrm{r} \boldsymbol{V}_r^T)^T$, where r is an integer in the range $1 < r < P$ and the subscript indicates that the matrices are truncated as follows:

$$\begin{array}{ccc} \boldsymbol{U} \in \mathbb{R}^{P\times P} & & \boldsymbol{U}_r \in \mathbb{R}^{P\times r} \\ \boldsymbol{\Sigma} \in \mathbb{R}^{P\times T} & \Longrightarrow & \boldsymbol{\Sigma}_\mathrm{r} \in \mathbb{R}^{r\times r} \\ \boldsymbol{V}^T \in \mathbb{R}^{T\times T} & & \boldsymbol{V}_r^T \in \mathbb{R}^{r\times T} \end{array} \tag{4}$$

In order to achieve the most accurate data recovery, avoiding overfitting, we need to determine the smallest value of r that allows for the approximation of $\boldsymbol{M}$ within a given error threshold ϵ. The approximation error e is defined as shown in equation $e = \frac{\|\boldsymbol{M} - \boldsymbol{U}_r \boldsymbol{\Sigma}_\mathrm{r} \boldsymbol{V}_r^T\|}{\|\boldsymbol{M}\|} < \epsilon$ (5) and it is a function of the l_2-*norm* between the original matrix and its approximation using r components. The determination of the optimal value for the threshold ϵ is discussed in the next chapter.

$$e = \frac{\|\boldsymbol{M} - \boldsymbol{U}_r \boldsymbol{\Sigma}_\mathrm{r} \boldsymbol{V}_r^T\|}{\|\boldsymbol{M}\|} < \epsilon \tag{5}$$

c. Data recovery

Because of the low-rank property of PMU data demonstrated in the previous section, the measurement vector $\boldsymbol{m}$ at time $T+1$ can be approximated with low error as the product of the reduced matrix $\boldsymbol{U}_r$ and vector $\boldsymbol{x}_r \in \mathbb{R}^{r\times 1}$ which corresponds to the representation of the measurements in the lower dimensional space. This relationship is demonstrated by equation $\boldsymbol{m}T = \boldsymbol{U}\,\boldsymbol{x} \cong \boldsymbol{U}_r\,\boldsymbol{x}_r$ (6) below.

$$\boldsymbol{m}^T = \boldsymbol{U}\,\boldsymbol{x} \cong \boldsymbol{U}_r\,\boldsymbol{x}_r \tag{6}$$

The data recovery process is based on computing $\boldsymbol{x}_r$ from the known measurements in $\boldsymbol{m}$ and then using it to estimate the missing values. Specifically, let's assume that at time $T+1$ the measurement vector $\boldsymbol{m}$ is missing one or more values in position(s) p; the goal is to estimate $\boldsymbol{m}_p$ (the p^{th} sample(s) of $\boldsymbol{m}$). In this case, equation $\boldsymbol{m}^T = \boldsymbol{U}\,\boldsymbol{x} \cong \boldsymbol{U}_r\,\boldsymbol{x}_r$ (6) can be re-written as in $(\boldsymbol{m}_{p'})^T \cong \boldsymbol{U}_r^{p'}\,\boldsymbol{x}_r$ (7) to only consider the known values of $\boldsymbol{m}$ and excluding the missing samples:

$$(\boldsymbol{m}_{p'})^T \cong \boldsymbol{U}_r^{p'}\,\boldsymbol{x}_r \tag{7}$$

where $\boldsymbol{m}_{p'}$ indicates the available measurements and $\boldsymbol{U}_r^{p'}$ indicates the rows of $\boldsymbol{U}_r$ corresponding to the available measurements. Given this setup, $\boldsymbol{x}_r$ can be estimated by solving the minimum least square problem shown in equation (8).

$$\widehat{\boldsymbol{x}}_r = \underset{\boldsymbol{x}_r}{\operatorname{argmin}} \left\| \boldsymbol{U}_r^{p'}\,\boldsymbol{x}_r - (\boldsymbol{m}_{p'})^T \right\| \tag{8}$$

where $\widehat{\boldsymbol{x}}_r$ indicates the estimated value of $\boldsymbol{x}_r$. At this point, the estimates $\widetilde{\boldsymbol{m}}_p$ of the missing values are computed by multiplying the rows of $\boldsymbol{U}_r$ corresponding to the missing values by the vector $\widehat{\boldsymbol{x}}_r$ as shown in equation $\widetilde{\boldsymbol{m}}_p = \boldsymbol{U}_r^p\,\widehat{\boldsymbol{x}}_r$ (9),

$$\widetilde{\boldsymbol{m}}_p = \boldsymbol{U}_r^p \ \widehat{\boldsymbol{x}}_r \tag{9}$$

where $\boldsymbol{U}_r^p$ is the matrix containing only the row(s) corresponding to the missing measurement(s).

III. IMPLEMENTATION

a. The PredictiveGrid™ platform

The PredictiveGrid™ platform is an Advanced Sensor Analytics Platform (ASAP) developed and commercialized by PingThings. Built specifically to ingest and store high resolution time series data, it is an extremely fast and powerful platform for handling large amounts of PMU data. The data recovery application described in this work is designed to work in conjunction with the PredictiveGrid™ to efficiently process at real time speeds the large amount of PMU data which is being collected by Dominion Energy.

At the heart of PredictiveGrid™ is a database specially designed for the storage of dense time-series data called Berkley Tree Database [13]-[14]. The data structure adopted for this database consists in storing the raw data points as well as aggregate statistics (maximum, minimum, mean, and point count) at different aggregation levels, corresponding to different time resolutions. This structure allows for extremely fast querying times as well as enabling efficient data visualization and event identification. As an example, a four-node cluster was shown to support the real-time ingestion of data coming from over 100,000 PMUs, each reporting at least 20 data streams at 60 Hz.

b. Data recovery application

One of the aims of this work is to design and build and effective and easy-to-use tool to improve PMU data quality which can be used in the context of offline data analytics. To this end, the algorithm presented in Section II is implemented in a standalone Python application.

Figure 2 shows a flowchart illustrating the main blocks and the logic of the data recovery application. The inputs to the function are the measurement matrix ($\boldsymbol{M}$) and the successive set of measurements ($\boldsymbol{m}$), where one or more value are missing. After performing SVD, the algorithm determines the value r which represents the number of principal components (and singular values) to be used. This is achieved by iteratively computing the approximation error e with increasing values of r, until the error is smaller than the predefined threshold ϵ. Preliminary testing has shown that the best results are obtained with a threshold value set as $\epsilon \sim 10^{-3}$ or 10^{-4}. After the optimal value of r is computed, the estimates of the missing values are calculated by solving equations (8) and $\widetilde{\boldsymbol{m}}_p = \boldsymbol{U}_r^p\ \widehat{\boldsymbol{x}}_r$
(9).

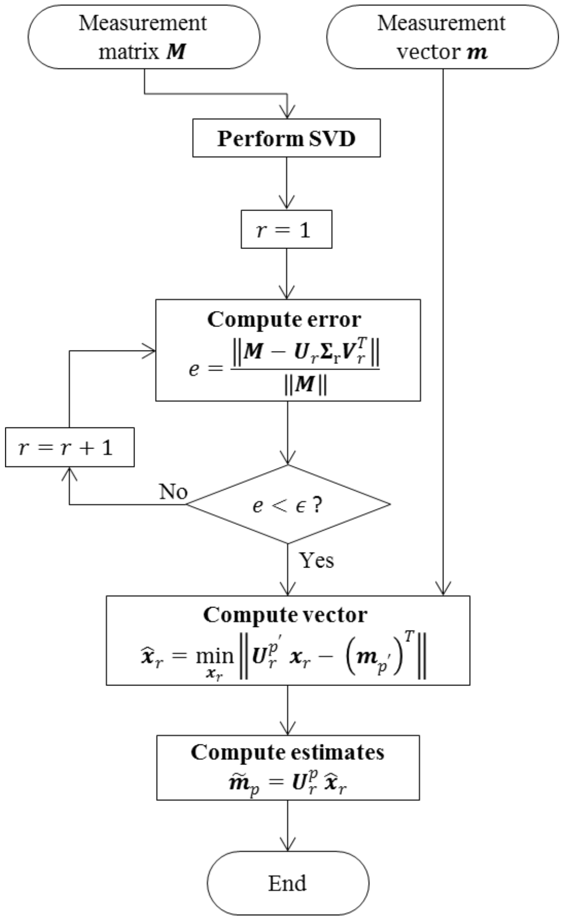


Figure 2: Data recovery algorithm.

While this algorithm is computationally highly efficient, it is to be noted that in the case of some offline studies which require large amounts of PMU data over long periods of time (hours and days) performing this type of data recovery might require considerable amounts of time and processing power. This process can be made substantially faster by not performing SVD and the calculation of r for each measurement vector; instead, these operations can be done once and the resulting $\boldsymbol{U}$ matrix and r used to compute missing values over multiple consecutive time samples. The longer the operating conditions are constant over time, the higher the number of consecutive measurement vectors that can be processed with a common matrix factorization. While not explored in this work, we intend to study the trade-off between computational complexity and modelling accuracy as system conditions change (especially during events) to determine the optimal sample interval at which to perform SVD and re-computing r.

IV. RESULTS

This section presents the results of different tests performed to verify the performance of the data recovery application developed in this work. The metric chosen to quantify the ability of the algorithm to correctly estimate the missing measurements is the normalized percentage difference as shown in equation $\delta_p = \frac{|\widetilde{\boldsymbol{m}}_p - \boldsymbol{m}_p|}{\boldsymbol{m}_p} \times 100$ (10) below. For each missing measurement p we compute the estimation error δ_p as:

$$\delta_p = \frac{|\widetilde{\boldsymbol{m}}_p - \boldsymbol{m}_p|}{\boldsymbol{m}_p} \times 100 \qquad (10)$$

where $\widetilde{\boldsymbol{m}}_p$ is the value of the p^{th} missing value recovered by the algorithm and $\boldsymbol{m}_p$ is the corresponding actual value.

The algorithm is tested on a total of 83 PMU streams, representing all current magnitude measurements available in the Dominion Energy 500 kV level transmission system; for the following tests, one month worth of PMU data is used. Different values for the number of time samples T on which the singular value decomposition is performed (see equation $\boldsymbol{M} = \begin{bmatrix} m_{1,1} & \cdots & m_{1,P} \\ \vdots & \ddots & \vdots \\ m_{T,1} & \cdots & m_{T,P} \end{bmatrix} \in \mathbb{R}^{T\times P}$ or $\in \mathbb{C}^{T\times P}$ **Error! Reference source not found.**) are tested, ranging from 5 seconds (150 samples, at 30Hz) to 20 seconds (600 samples). Since the data recovery performance was almost identical in every case, for the following tests $T = 300$ is used, which means that the measurement matrix $\boldsymbol{M}$ has dimensions 300×83.

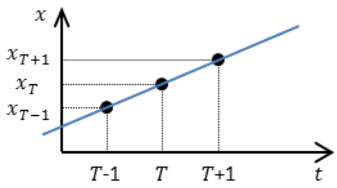


Figure 3: Linear estimator.

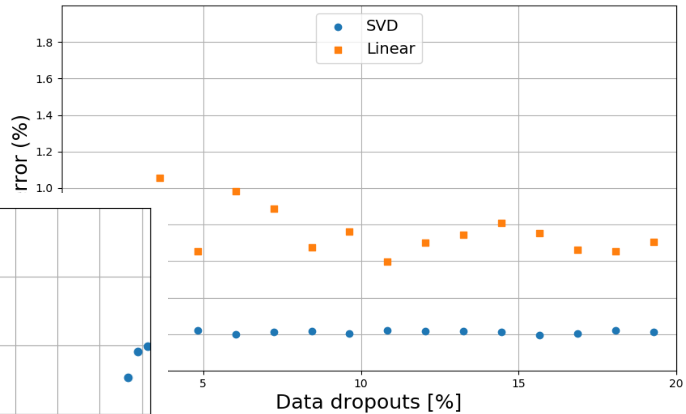
Additionally, to further justify the use of SVD and demonstrate its high effectiveness on PMU data, the recovery performance of the SVD-based algorithm is compared to a linear estimator. The linear approach estimates a missing value at time $t = T + 1$ by performing a linear fit of the two previous (known) samples as shown in Figure 3. The missing value x_{T+1} is thus computed as:

$$x_{T+1} = 2x_T - x_{T-1}\,. \tag{11}$$

a. Recovery error with multiple simultaneous drop-outs

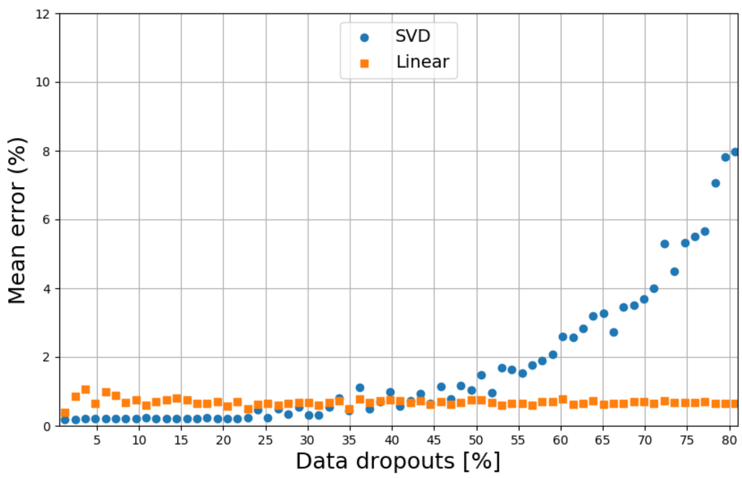
In this section, the robustness of the data recovery application against multiple simultaneous drop-outs is verified. That is, the average estimation error is calculated for different scenarios in which an increasing number of missing measurements need to be estimated for the same time sample. The test consists in randomly dropping p measurements from vector $\boldsymbol{m}$ and estimate them based on $\boldsymbol{M}$ with $T = 300$ samples; for each value p from 1 to 83, 100 trials are run (e.g. 100 different $\boldsymbol{m}$). The percentage error is then measured between each $\tilde{\boldsymbol{m}}_p$ and $\boldsymbol{m}_p$ and the average for each value of p is calculated.

Figure 4 shows the average estimation error as a function of the percentage of simultaneous data drop-outs in a measurement vector (calculated as $\frac{p}{83} \times 100$). As we can see, on average the SVD-based algorithm can recover a missing value with a very high accuracy of 0.2%. Moreover, even when 20% of the measurements at a given time are dropped, they can be recovered with the same accuracy. As expected, this approach performs considerably better than the simple linear estimator which shows an error 3 to 5 times higher.



Average estimation error as a function of
centage of simultaneous missing values

In order to understand the limits of the data recovery application, it is interesting to see how the performance varies as even more measurements are dropped. Figure 5 shows the mean estimation error for drop-out rates of up to 80%. While the error is mostly constant up to around 25%, as the percentage of drop-outs increases beyond 30/35% the recovery error



quickly rises reaching 8%. Overall, the performance of the SVD-based algorithm is extremely satisfactory considering that, as explained in the introduction, normally around only 1% of data is subject to drop-out at any given time.

Finally, while the linear estimator has higher error rates for low drop-out percentages, its behavior is constant and does not depend on the number of drop-outs in a measurement vector. This can be explained by the fact that, unlike the SVD-based algorithm, the linear estimator is used on each stream independently and does not rely on information from other streams.

Figure 5: Average estimation error as a function of the percentage of simultaneous missing values for high drop-out rate

b. Recovery performance of each data stream

In the previous section, for each measurement vector $\boldsymbol{m}$**,** values were dropped by randomly selecting p measurements in each trial; thus, the resulting estimation errors only describe the average performance of the algorithm. Since the data recovery process is based on the fact that PMU measurements are highly correlated, we can assume that the recovery performance will vary between each data stream depending on how many similar measurements there are and on how strong the correlation is.

In this test, each measurement stream is studied independently: given a vector $\boldsymbol{m}$, each measurement p is dropped, one at a time, and the error e_p computed. This process is repeated for 50 different $\boldsymbol{m}$ vectors and the average error for each measurement p is calculated.

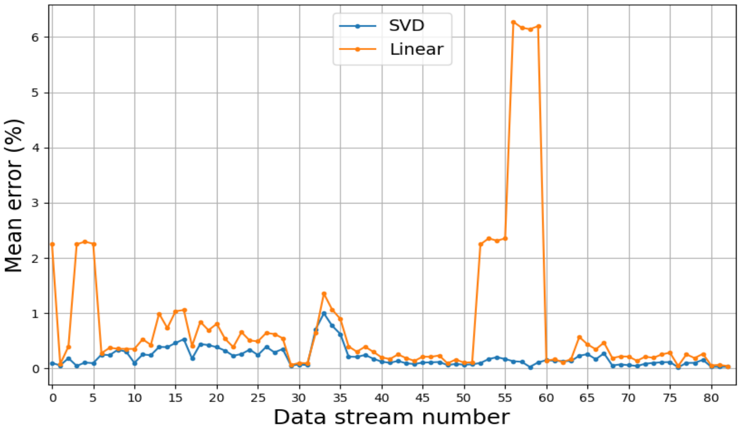


Figure 6: Average estimation error for each individual measurement

Figure 6 shows that the average recovery error is different for every measurement. In the case of the SVD-based algorithm, most of the measurements can be recovered with an error of 0.2% or lower. However, some streams show a much higher error reaching 1%. When performing data recovery on a new dataset, the results of this test can be used as a way to determine and assign a confidence score to the estimated values depending on the measurement stream that was missing.
Finally, it can be seen that the linear estimator consistently performs worse than the SVD-based one and in some cases it shows extremely high error peaks which are not present in the results of the more advanced algorithm.

c. Performance in case of consecutive data drop-outs on one stream

In this section, the performance of the data recovery algorithm is studied in the case of consecutive drop-outs of the same measurement. For this test, a stream p is selected and the measurement at time $t = T + 1$ is estimated; the estimation error $\delta_{p,T+1}$ between the real value $\boldsymbol{m}_{p,T+1}$ and the estimated value $\widetilde{\boldsymbol{m}}_{p,T+1}$ is computed. At this point, the measurement matrix $\boldsymbol{M}$ is updated by substituting the real value of $\boldsymbol{m}_{p,T+1}$ with the one just estimated. After that, the successive value $\widetilde{\boldsymbol{m}}_{p,T+2}$ is estimated and the error calculated. This process is repeated up to time $t = T + 100$, and it simulates what would happen if 100 consecutive samples (over 3 seconds) of stream p were lost and had to be estimated. The procedure just described is used for 75 different values of p, and the average error at each time step t is calculated.

Figure 7 shows the average error over the 75 trials as a function of the number of consecutive drop-outs. As we can see, even if a data stream is lost for many consecutive samples, the SVD-based algorithm maintains good recovery accuracy. It is interesting to notice how the average error slowly increases from around 0.2% to about 0.4%. It is also important to reiterate that the average length of drop-outs observed in real data is only 3.5.

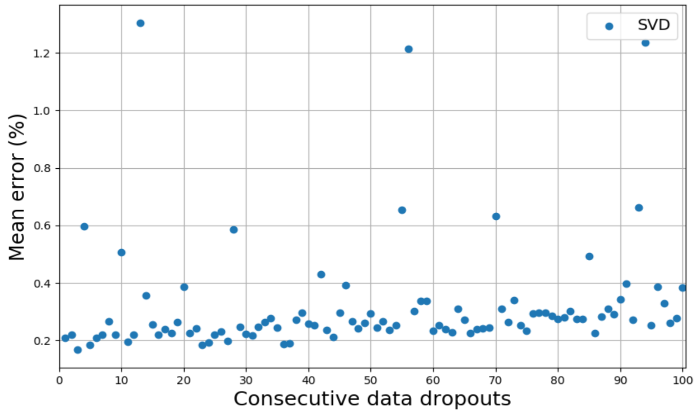


Figure 7: Average error as a function of the number of consecutive drop-outs on one data stream

Since the SVD-based algorithm estimates missing values based both on the currently observed measurements as well as the past values up to T samples before ($T = 300$ in these tests), it is very robust against consecutive drop-outs. This is not the case for the linear estimator as shown in Figure 8. Since each estimate is only based on the previous two samples, the error is compounded and it quickly diverges to very high values. To further illustrate this, when testing the linear estimator, from time sample $t = 15$ to $t = 18$ the estimated values are substituted with the true values. As expected, when using true values, the error immediately goes back to lower levels but it starts diverging again as soon as the estimated value are again used for the estimation of successive samples.

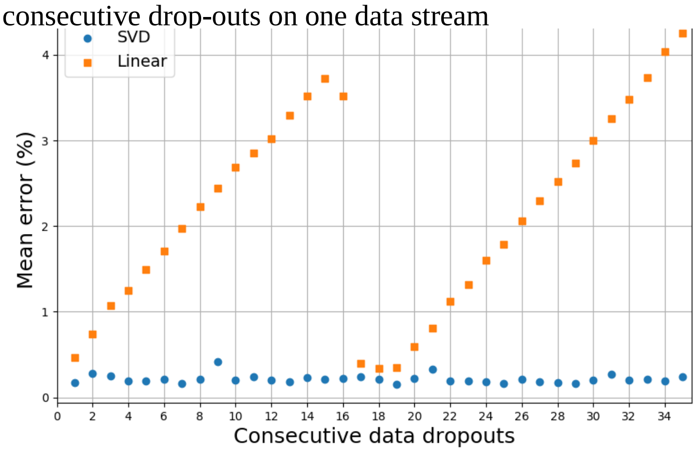


Figure 8: Comparison between SVD-based and linear algorithms

V. CONCLUSIONS

In this paper, we have described the design of a python application to perform data recovery of missing PMU values. The algorithm uses SVD to leverage the low-rank nature of PMU data which originates from the high spatio-temporal correlation existing between neighbouring measurements. We have shown that this approach guarantees excellent recovery performance under different scenarios, both with simultaneous drop-outs and consecutive drop-outs. In fact, all the cases tested represented extreme scenarios of bad data quality; in reality, the drop-outs we observe are much less severe. This application represents a valid tool to improve the workflow of PMU data analytics, assisting in the development of new applications.

BIBLIOGRAPHY

[1] PMU Applications Requirements Task Force (PARTF), "PMU Data Quality," NASPI, PNNL 26313 / NASPI-2017-TR-002, March 2017

[2] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[3] K. Lee and Y. Bresler, “ADMiRA: Atomic decomposition for minimum rank approximation,” *IEEE Trans. Inf. Theory*, vol. 56, no. 9, pp. 4402–4416, 2010.
[4] P. Jain, R. Meka, and I. S. Dhillon, “Guaranteed rank minimization via singular value projection,” *Advances in Neural Information Processing Systems*, 2010, pp. 937–945.
[5] P. Gao, M. Wang, S. G. Ghiocel, J. H. Chow, B. Fardanesh and G. Stefopoulos, “Missing Data Recovery by Exploiting Low-Dimensionality in Power System Synchrophasor Measurements,” in *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 1006-1013, March 2016.
[6] R. Meka, P. Jain, and I. S. Dhillon, “Matrix completion from powerlaw distributed samples,” *Advances in neural Information Processing Systems*, 2009, pp. 1258-1266.
[7] M. Liao, D. Shi, Z. Yu, W. Zhu, Z. Wang and Y. Xiang, “Estimate the Lost Phasor Measurement Unit Data Using Alternating Direction Multipliers Method,” *2018 IEEE/PES Transmission and Distribution Conference and Exposition (T&D)*, Denver, CO, 2018, pp. 1-9.
[8] Z. Yang, H. Liu, T. Bi, Q.Yang, and A. Xue, “A PMU data recovering method based on preferred selection strategy,” *Global Energy Interconnection*, vol. 1, no. 1, pp. 63-69, January 2018.
[9] Z. Chu, A. Pinceti, R. Sen Biswas, O. Kosut, A. Pal, and L. Sankar, “Can predictive filters detect gradually ramping false data injection attacks against PMUs?,” Accepted *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*
[10] D. Yang, “A data analytics framework for regional voltage control,” Ph.D dissertation, Virginia Tech, 2017.
[11] D. Yang, Z. Nie, K. Jones and V. Centeno, “Adaptive decision-trees-based regional voltage control,” *2017 North American Power Symposium (NAPS)*, Morgantown, WV, 2017, pp. 1-6.
[12] PingThings, [Online]. Available: https://www.pingthings.io/index.html
[13] M. P. Andersen, S. Kumar, C. Brooks, A. von Meier and D. E. Culler, “DISTIL: Design and implementation of a scalable synchrophasor data processing system,” *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Miami, FL, 2015, pp. 271-277.
[14] M. P. Andersen and D. E. Culler, “BTrDB: Optimizing Storage System Design for Timeseries Processing,” 14th USENIX Conference on File and Storage Technologies (FAST 16), Santa Clara, CA, 2016, pp. 39-52.