Outlier Detection for ARM Data

Yuping Lu[1], Jitendra Kumar[2], Nathan Collier[2] and Michael A. Langston[1]

***Abstract*— The Atmospheric Radiation Measurement (ARM) Data Center collects data from either permanent or mobile facilities around the globe. These data are then ingested and used to create high level scientific products which requires great accuracy. Multiple methods are available to detect these outliers from ARM time series data. As outliers are common in the collected data which could be either an instrument failure or extreme weather event, Pearson Correlation Coefficient was first examined to measure the pairwise correlations between variables. New version of Singular Spectrum Analysis (SSA) was also introduced to detect outliers. K-means was applied in a different manner to filter out the abnormal records as well. Pearson Correlation Coefficient, SSA and K-means methods were later combined together as a whole framework to track down these outliers. Compared to the current data quality reports stored in the DQR database, our results showed this framework is promising.**

I. Introduction

The Atmospheric Radiation Measurement (ARM) user facility was founded by the U.S. Department of Energy (DOE) in 1989 [1]. Since then, its aim is to be the platforms for the observation and study of Earth's climate. Huge ARM datasets are collected from instruments deployed in different ground stations across the globe [2]. ARM Data Center is responsible for ingesting these collected data and creates high level scientific data products for distribution and the improvement of global climate models (GCMs) [3]. These high level data products, also called "Value Added Products" (VAPs) are highly dependent on the correctness of the raw data. Thus it is crucial to detect those outliers in the raw data and correct them.

As outliers are pretty common in these datasets. Currently, these datasets are checked manually and outliers are stored in Data Quality Report (DQR) database to be fixed.

We will use this section to introduce the background of outlier detection for time series data. [4]

II. Datasets

ARM data are stored in Network Common Data Form (NetCDF) format which is self-describing and machine-independent [5], [6]. NetCDF format also has good performance and data compression. It is commonly used to handle scientific data, especially those from the climatology, meteorology, oceanography and GIS projects. ARM data is publicly available and can be downloaded from ARM Data Archive (http://www.archive.arm.gov). Kinds of raw data are stored in ARM Data Center. It ranges from *Atmospheric Profiling* to *Satellite Observations*. All these data are measured at different locations using different instruments. Each instrument may only work on a specified time range. For the raw NetCDF dataset collected from each instrument, it contains multiple variables.

TABLE I
SGPMET DATASETS TESTED

Instrument	E1	E3	E4	E5	E6	E7
Begin Year	1996	1997	1996	1997	1997	1996
End Year	2008	2008	2010	2008	2010	2011
Instrument	E8	E9	E11	E13	E15	E20
Begin Year	1994	1994	1996	1994	1994	1994
End Year	2008	2017	2017	2017	2017	2010
Instrument	E21	E24	E25	E27	E31	E32
Begin Year	2000	1996	1997	2004	2012	2012
End Year	2017	2008	2001	2009	2017	2017
Instrument	E33	E34	E35	E36	E37	E38
Begin Year	2012	2012	2012	2012	2012	2012
End Year	2017	2017	2017	2017	2017	2017

In this paper, we only tested Surface Meteorology Systems (MET) data collected from the Southern Great Plains (SGP). There were total 24 instruments in SGP area and we chose 5 typical variables which are *temp_mean*, *vapor_pressure_mean*, *atmos_pressure*, *rh_mean* and *wspd_arith_mean* from multiple variables. Table 1 contains the detail of these datasets.

III. Methodology

We have introduced many kinds of outlier detection method in the first section. We carefully picked three algorithms and tested them on ARM data. We also did necessary preprocessing before running these algorithms. The first level raw data is stored in minute level. It is normalized for pairwise comparison algorithm. Some algorithms may not need so much detail information to extract outliers. Thus we created a second level data by averaging the 1440 minute data points into one day point from the raw data. The second level data can save a lot of running time and is easier for Plotly [7] and Matplotlib [8] to visualize. The third level data was especially created for multivariate method by standardization all the 5 variables into the same scale based on the second level data. Below we will talk about each algorithm in detail.

A. Pearson Correlation Coefficient

Pearson Correlation Coefficient was first introduced by Karl Pearson [9]. It is used to measure the linear correlation between two variables. Pearson correlation coefficient is calculated from the covariance of two variables divided by the multiplication of the standard deviation of those two variables. Thus the value falls in [-1, 1]. If the value is close to -1, it means those two variables are highly negatively related. On the other hand, then the two variables are strongly

[1]University of Tennessee, Knoxville, TN, USA
[2]Oak Ridge National Laboratory, Oak Ridge, TN, USA

positively related. If the value is near 0, it means those two variables don't have linear relation.

We performed pairwise comparison of the 5 variables using Pearson Correlation on all the instruments in a seasonal level. The result in figure 1 makes sense and all the correlations in this violin plot are normally distributed. For example, x5 the Pearson Correlation between *temp_mean* and *vapor_pressure_mean* is positively correlated with correlation mean close to 0.75. x1 is negatively correlated with correlation mean close to -0.60. We used this correlation as base knowledge. If a pairwise pearson correlation of two variables from a specific season of a instrument falls out of that range, we treated that seasonal data as outliers. x1 is negatively correlated with correlation mean close to -0.60. We used this correlation as base knowledge. If a pairwise pearson correlation of two variables from a specific season of a instrument falls out of that range, we treated that seasonal data as outliers.

B. Singular Spectrum Analysis

Singular Spectrum Analysis (SSA) is a popular method for time series data analysis [10], [11]. The general idea is to use a subset of the decomposition of trajectory matrix to approximate it. Many applications can be found in [10]. For example, SSA can be applied to monitor volcanic activity [12]. It can also be used to extract trend [13]. Different from the classic SSA method, we defined our own version of SSA to best work on ARM data. Figure 2 is a demonstration of the workflow of SSA. Below is the formal description of the algorithm.

Assume we have an ARM time series data Y of length T.

$$Y = (y_1,\ \ldots,\ y_T)$$

Here $T > 2$ and y_i is not empty. Let L $(1 < L \leq T/2)$ be the window size and $K = T-L+1$. In general, the algorithm contains two main parts: decomposition and reconstruction.

1) The first step is to form trajectory matrix $\mathbf{X}$ from vector Y by embedding subsets of Y. These subsets of Y X_i are lagged vectors of length L.

$$X_i = (y_i,\ \ldots,\ y_{L+i-1})^T \quad (1 \leq i \leq K)$$
$$\mathbf{X} = [X_i,\ \ldots,\ X_K]$$

Thus the trajectory matrix is

$$\mathbf{X} = (x_{ij})_{i,j=1}^{L,K} = \begin{pmatrix} y_1 & y_2 & y_3 & \cdots & y_K \\ y_2 & y_3 & y_4 & \cdots & y_{K+1} \\ y_3 & y_4 & y_5 & \cdots & y_{K+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_L & y_{L+1} & y_{L+2} & \cdots & y_T \end{pmatrix} \quad (1)$$

where $x_{ij} = y_{i+j-1}$. We can see from equation 1 that matrix $\mathbf{X}$ has equal elements on anti-diagonals and therefore it is Hankel matrix.

2) Assume matrix $\mathbf{S} = \mathbf{X}\mathbf{X}^T$, thus we perform the singular value decomposition (SVD) on $\mathbf{S}$. The eigenvalues of S are denoted by $\lambda_1, \ldots, \lambda_L$ in the decreasing order of magnitude $(\lambda_1 \geq \ldots \geq \lambda_L \geq 0)$ and corresponding eigenvectors are denoted by $P_1, \ldots, P_L$. Let $d = rank\ \mathbf{X}$ and $V_i = \mathbf{X}^T P_i / \sqrt{\lambda_i} (i = 1, \ldots, d)$. Thus, the trajectory matrix X can also be written as

$$\mathbf{X} = \mathbf{X_1} + \ldots + \mathbf{X_d} \quad (2)$$

where $\mathbf{X_i} = \sqrt{\lambda_i} P_i V_i^T$.

3) In this step, we use Fast Fourier transform (FFT) to find the dominant frequency of each eigenvector [14] . Algorithm 1 shows the whole process.

Algorithm 1: Dominant Frequency Finder

Input : λ of $\mathbf{S}$ and corresponding eigenvectors $\mathbf{P}$
Output: Dominant frequency of each eigenvector

```
fftfreq ← Discrete Fourier Transform sample frequencies
fft ← Discrete Fourier Transform
len ← size of λ
frequencies ← zero vector of size len
fs ← fftfreq(λ)
ix ← indices that sort fs
fs ← fs[ix]
for i in range(len) do
    p1 ← abs(fft(P[:,i]))
    ps ← p1**2
    ps ← ps[ix]
    frequencies[i] ← fs[index of the maximum value in ps]
end
return abs(frequencies)
```

4) Let G be the vector of user specified periodicity. We then convert G into a vector of targeted frequency TF for the next step. Here 0 is also added to TF. We use M to denote the length of TF.

5) As mentioned in step 2, there are d $\mathbf{X_i}$. The goal is to build an approximation of X by taking a subset of the decomposition $\mathbf{X_i}$. This approximation is formed by taking eigenvectors whose dominant frequency is close to the targeted frequency. Thus we have an approximation matrix $\mathbf{Xt}$ of size $M \times L \times K$.

6) Now we reconstruct the signal $\hat{Y}$ by taking a mean of all the approximations. The generated matrix $\mathbf{Yt}$ with size $M \times T$ can be used to approximate Y.

$$\hat{Y} = \sum_{i=1}^{M} \mathbf{Yt}[i] \quad (3)$$

In this paper, we chose the temp_mean data from instrument E33 as Y to illustrate SSA. Because SSA requires the time series data to be continuous, we replaced the empty points with the average temp_mean value for that day in a year.

We set L = 400 and picked year and month as the periodicity groups G = [365, 30]. Thus TF = [0, 0.00273973, 0.03333333]. The generated matrix $\mathbf{Yt}$ has 3 rows after performing SSA. And $\hat{Y}$ = $\mathbf{Yt}[0]$ + $\mathbf{Yt}[1]$ + $\mathbf{Yt}[2]$. The residual is then extracted from the raw data R = Y - $\hat{Y}$. As $\hat{Y}$ is the

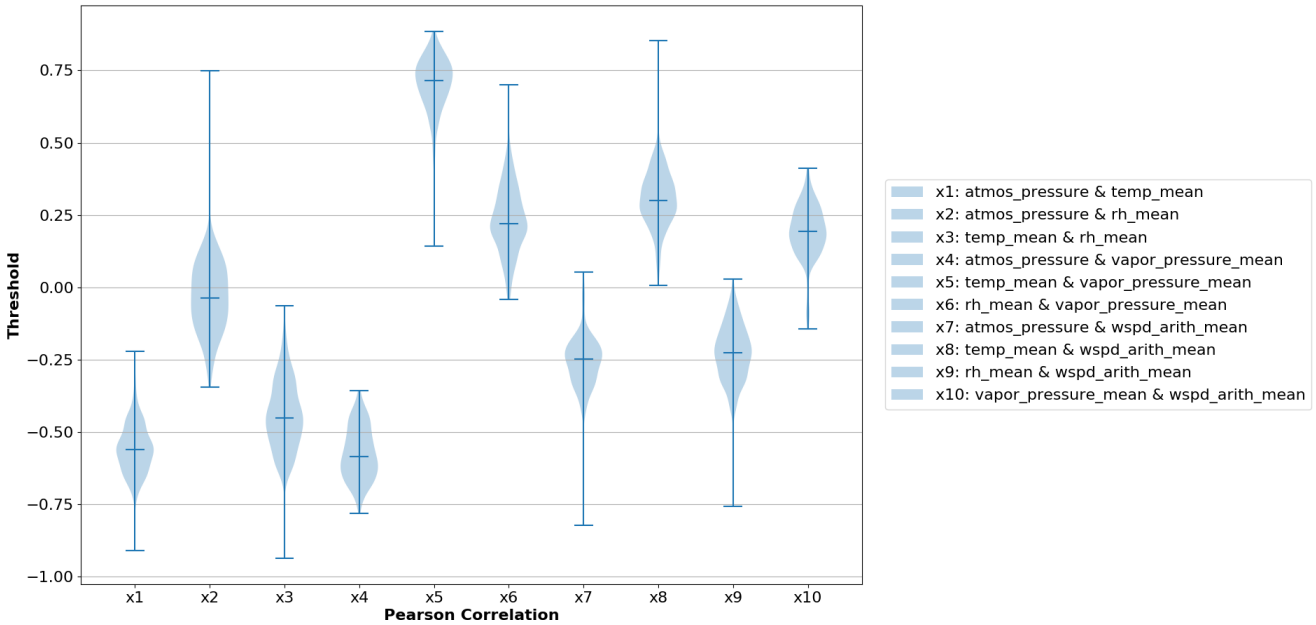

Fig. 1. Violin plot: Spring 5 variables from SGPMET

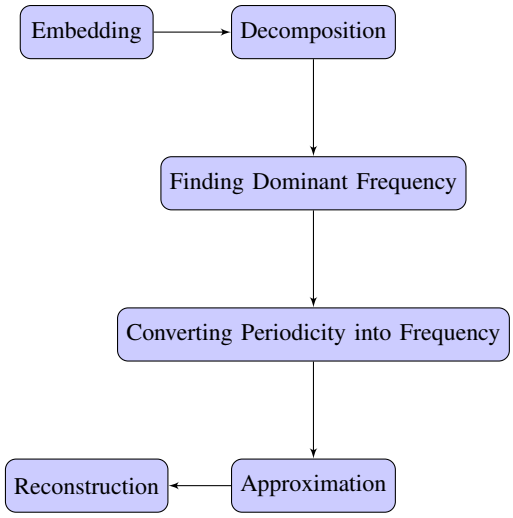

Fig. 2. Flowchart of SSA

approximation which is a ”perfect” representation of Y. We then extracted the extreme values from the residual. Those extracted outliers are outliers. Figure 3 is a visualization of the result. The first row is the raw data Y. The orange line $\mathbf{Yt}[0]$ is the trend. As we can see, the trend is pretty flat from 2012 to 2017. The second row and third row are $\mathbf{Yt}[1]$, $\mathbf{Yt}[1]$ respectively. The Year data matches the pattern of the raw data. The last row is the residual. Those peak values outside the blue shaded area are outliers.

C. K-means

K-means is a partitioning clustering algorithm [15], [16]. It starts with the k centroids user specified, and assigns the points to the nearest centroid. Then it computes new k centroids and assign the rest points to these centroids again. The process repeats until it converges.

Algorithm 2: K-means Outlier Detection

Input : ARM time series data
Output: Outliers

```
outliers ← ∅
df ← ARM time series data
data ← df['atmos_pressure','temp_mean',
  'rh_mean','vapor_pressure_mean','wspd_arith_mean']
number_of_clusters ← 4
clusters ← K-means(data, number_of_clusters)
distances ← Distance between each point and its
  centroid
mean ← arithmetic mean of distances
sigma ← standard deviation of distances
threshold ← mean + 3 * sigma
for i in range(size of distances) do
    if distances[i] > threshold then
        outliers ← outliers ∪ distances[i]
    end
end
return outliers
```

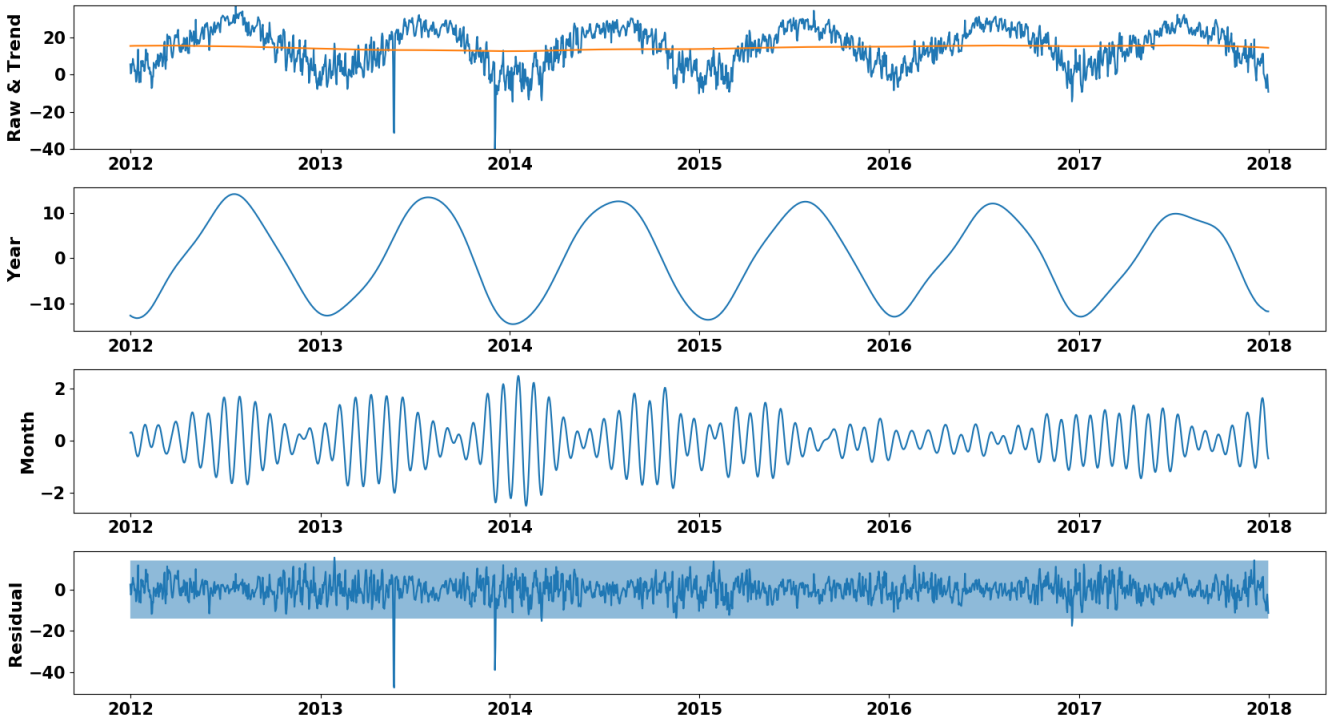

Fig. 3. Example of SSA application on ARM data. The full decomposition of temp_mean data from instrument E33.

In this paper, we didn't stop after clustering ARM data with K-means. We transformed the generated clusters into a vector of distance between each point and its corresponding centroid. Algorithm 2 describes the whole process. Unlike SSA, we used all the 5 variables mentioned in Datasets section together to extract outliers.

Again, we used data from instrument E33 as an example for K-means. Here we set k to 4 as each year has 4 seasons. Figure 4 visualized the outliers detected from E33. Y axes in this figure is the distance metric. The pattern of these points is close to the raw data.

IV. Results and Discussion

SSA is an univariate method. K-means is a multivariate method. Results and pics go here. Comparison metric: DQR database [17].

Add the drawbacks of each algorithm and their strong parts.

How do you pick extreme values as outliers? Some methods do not work. Here we use the three sigma rule to extract outliers [18].

Precision and recall was first defined in [19]. It is commonly used to measure the quality of classification tasks [20]. Precision is calculated from True Positives divided by the sum of True Positives and False Positives. On the other hand, recall is measured from True Positives divided by the sum of True Positives and False Negatives. In this paper, detected outliers in the DQR database are the ground truth. So we treated these as True Positives. Thus detected outliers not in the DQR database are False Positives. Undetected values which in the DQR database are False Negatives, and which not in the DQR database are True Negatives. Analysis of table 2 and 3 goes here.

TABLE II
Precision and Recall of SSA and K-means

Method	Variable	Precision	Recall
SSA	temp_mean	16.00%	1.20%
SSA	vapor_pressure_mean	20.70%	1.40%
SSA	atmos_pressure	0.00%	0.00%
SSA	rh_mean	14.80%	0.50%
SSA	wspd_arith_mean	0.60%	1.50%
Kmeans	5 together	12.90%	1.90%
Combined	5 together	11.10%	4.10%

TABLE III
Comparison of SSA and K-means Outlier Set Size

	Outlier Set Size
SSA	922
K-means	508
Intersection	378
Symmetric Difference	674

More figures and results can be found on github.

V. Conclusions

After we compute those extreme values, use other methods to extract part of them as outliers? We presented a combined model to detect outliers for ARM data. Future work: ML and tried methods working on multiple instruments multiple sites [21].

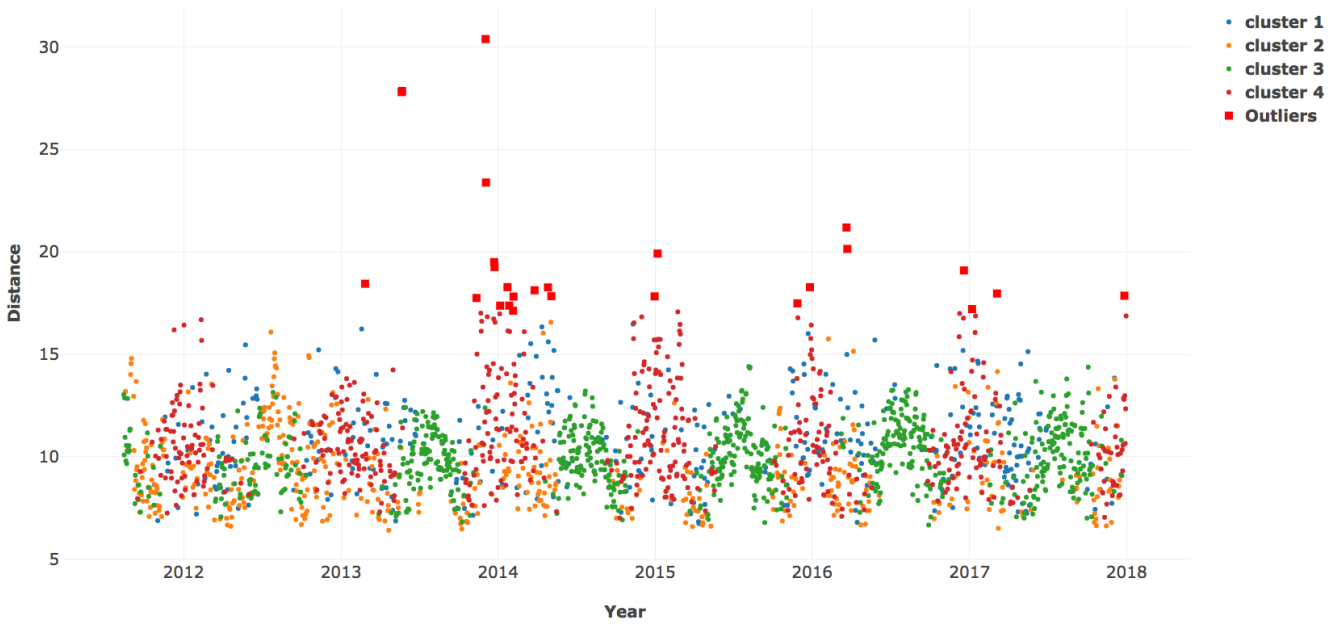

Fig. 4. Outliers detected using K-means for E33

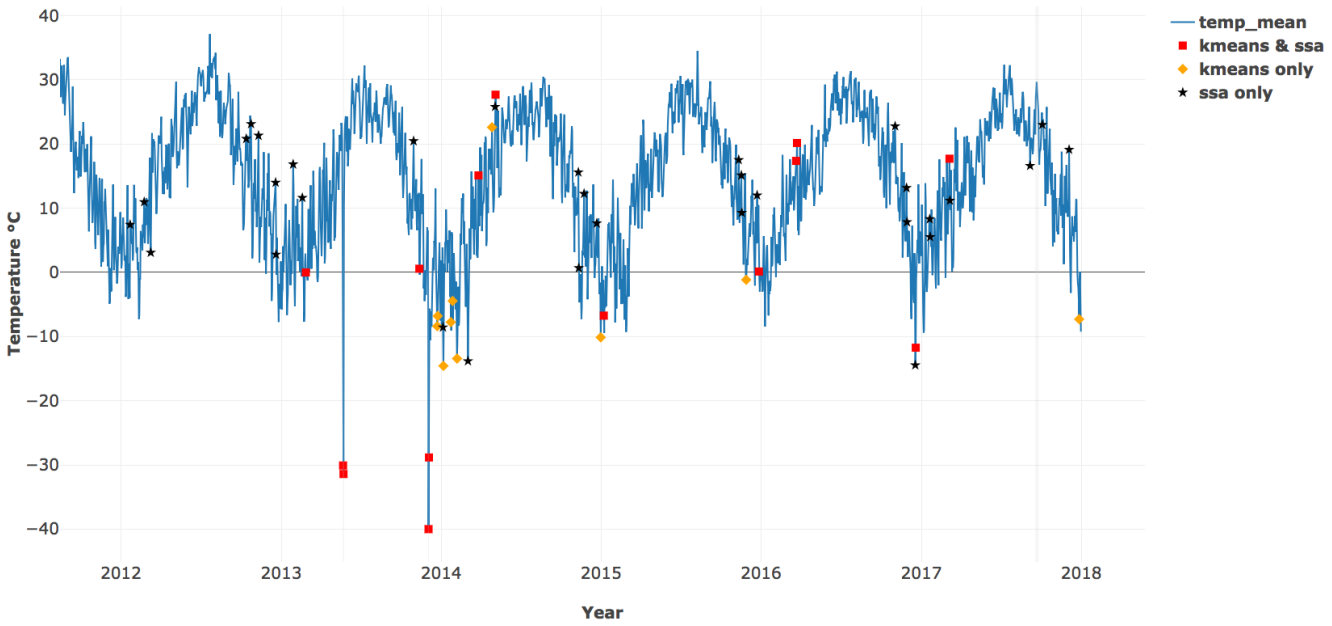

Fig. 5. E33 temp_mean combined

ACKNOWLEDGMENT

This research was supported by the Atmospheric Radiation Measurement (ARM) user facility, a U.S. Department of Energy (DOE) Office of Science user facility managed by the Office of Biological and Environmental Research.

REFERENCES

[1] "Arm research facility," https://www.arm.gov/, accessed: 2018-06-22.
[2] G. M. Stokes and S. E. Schwartz, "The atmospheric radiation measurement (arm) program: Programmatic background and design of the cloud and radiation test bed," *Bulletin of the American Meteorological Society*, vol. 75, no. 7, pp. 1201–1222, 1994.
[3] K. Gaustad, T. Shippert, B. Ermold, S. Beus, J. Daily, A. Borsholm, and K. Fox, "A scientific data processing framework for time series

netcdf data," *Environmental modelling & software*, vol. 60, pp. 241–249, 2014.

[4] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2250–2267, 2014.

[5] R. Rew and G. Davis, "Netcdf: an interface for scientific data access," *IEEE computer graphics and applications*, vol. 10, no. 4, pp. 76–82, 1990.

[6] Unidata. (2014) Network common data form (netcdf) version 4.1.1. Boulder, CO: UCAR/Unidata. [Online]. Available: https://doi.org/10.5065/D6H70CW6

[7] P. T. Inc. (2015) Collaborative data science. Montreal, QC. [Online]. Available: https://plot.ly

[8] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[9] K. Pearson, "Note on regression and inheritance in the case of two parents," *Proceedings of the Royal Society of London*, vol. 58, pp. 240–242, 1895.

[10] N. Golyandina and A. Zhigljavsky, *Singular Spectrum Analysis for time series*. Springer Science & Business Media, 2013.

[11] N. Golyandina and A. Korobeynikov, "Basic singular spectrum analysis and forecasting with r," *Computational Statistics & Data Analysis*, vol. 71, pp. 934–954, 2014.

[12] E. Bozzo, R. Carniel, and D. Fasino, "Relationship between singular spectrum analysis and fourier analysis: Theory and application to the monitoring of volcanic activity," *Computers & Mathematics with Applications*, vol. 60, no. 3, pp. 812–820, 2010.

[13] T. Alexandrov, "A method of trend extraction using singular spectrum analysis," *arXiv preprint arXiv:0804.3367*, 2008.

[14] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of computation*, vol. 19, no. 90, pp. 297–301, 1965.

[15] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[16] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[17] R. McCord and J. Voyles, "The arm data system and archive," *Meteorological Monographs*, vol. 57, pp. 11–1, 2016.

[18] F. Pukelsheim, "The three sigma rule," *The American Statistician*, vol. 48, no. 2, pp. 88–91, 1994.

[19] J. W. Perry, A. Kent, and M. M. Berry, "Machine literature searching x. machine language; factors underlying its design and development," *Journal of the Association for Information Science and Technology*, vol. 6, no. 4, pp. 242–254, 1955.

[20] D. L. Olson and D. Delen, *Advanced data mining techniques*. Springer Science & Business Media, 2008.

[21] J. D. Phillips, W. Schwanghart, and T. Heckmann, "Graph theory in the geosciences," *Earth-Science Reviews*, vol. 143, pp. 147–160, 2015.