k-ShapeStream: Probabilistic Streaming Clustering for Electric Grid Events

Mohini Bariya, Alexandra von Meier
Dept. of Electrical Engineering
U.C. Berkeley
Berkeley, USA
{mohini, vonmeier}@berkeley.edu

John Paparrizos, Michael J. Franklin
Dept. of Computer Science
University of Chicago
Chicago, USA
{jopa, mjfranklin}@uchicago.edu

2021 IEEE Madrid PowerTech | 978-1-6654-3597-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/PowerTech46648.2021.9494830

***Abstract*—We present k-ShapeStream, a clustering method for streaming time-series data. In addition to the algorithmic novelty, the method represents a highly practical approach for electric grid data analytics, requiring no model assumptions or ground truth information, running sustainably on ever growing datasets, and providing intuitive and insightful results to grid operators. We demonstrate the effectiveness of k-ShapeStream using several months of real synchrophasor data from an operational distribution network in California. Through two case studies on (i) transformer tap changes; and (ii) voltage sags, we illustrate how k-ShapeStream assists in identifying and analyzing recurring grid events, a critical task for decision making in electric grids.**

***Index Terms*—clustering, streaming, PMU, time series, big data, situational awareness**

I. Introduction

The electric grid faces challenges from the proliferation of renewable generation technologies, novel loads, and networked grid equipment. Increasingly frequent extreme weather events damage grid infrastructure and may cause dangerous grid failures with catastrophic consequences. This is the case in California, where high winds have caused grid failures that sparked wildfires [1]. As networked control equipment spreads, cyber-attacks on the grid are another burgeoning concern; a successful attack on the Ukrainian grid in 2015 affected $225,000$ customers [2]. Integral to maintaining reliable and safe grid operations in the face of this multitude of challenges is better *situational awareness*: knowing what is happening in real time. This is especially necessary in distribution networks, traditionally the opaque and passive ends of the system. Today, distributed generation and novel loads are turning distribution systems into active and dynamic networks that must be monitored and controlled [3].

The need for better situational awareness has driven the development of advanced grid sensors, such as Phasor Measurement Units (PMUs). Already widely deployed in transmission networks and increasingly deployed in distribution, PMUs report accurately time-stamped phasor current and voltage measurements at up to 120 Hz [4]. Unfortunately, measurement data alone do not deliver situational awareness. If unmediated, high-resolution data streams can be overwhelming to already strained engineers and operators [5]. Algorithms are needed to transform measurements into insights. An important class of such methods focuses on event classification, which aims to categorize events in grid measurements. Events include any significant permanent or transient system state change. Events may be detected in any measurement stream (e.g., voltage, current, or frequency) and range from routine (e.g., capacitor switching and transformer tap changes) to unusual abnormalities (e.g., high-impedance faults).

Machine learning (ML) techniques are well-suited to event classification in the newly data-rich grid context, and the literature is replete with examples. Several works train classifiers on labelled grid events [6]–[14]. The details of the approaches differ: for example, the classifiers used include SVMs, decision trees, and neural networks; measurements may be real-world or simulated; some works transform raw time-series measurements into carefully engineered features; and the specificity of labels used varies. What they have in common, however, is a reliance on significant amounts of expert-labelled data. Labelled data is scarce, limited by time and privacy constraints. Even once trained, many ML models produce brittle and non-intuitive results. This is problematic for algorithms which need to generalize to different systems and require human feedback.

In light of these issues, clustering is a promising alternative. Clustering categorizes data points into groups based on some similarity measure in an *unsupervised* manner (i.e., requiring no labelled data). If performed on raw measurements or simple features, clustering produces highly intuitive results. In the grid context, clustering has been extensively applied to load data [15]–[17]. Closest to our work are [18]–[20], which apply off-the-shelf clustering approaches to PMU time-series measurements of events. We build on these work in two important respects. First, we develop a streaming clustering approach, namely, k-ShapeStream, that allows clusters to be updated with new event data without requiring access to historic event data. This is vital for algorithms that will run online as new

This work was supported in part by the National Science Foundation, Award 1840192-FW-HTF, "Collaborative Research: Augmenting and Advancing Cognitive Performance of Control Room Operators for Power Grid Resiliency" and gifts from NetApp, Cisco Systems, and Exelon Utilities.

measurements arrive. The approach in prior work necessitates re-clustering on the entire set of new and historic event data every time new events are added, which quickly becomes impractical. Second, we develop a probabilistic time-series distance measure for clustering with multiple benefits. Our distance measure improves algorithm performance, enables anomaly detection, and enhances human interpretation of the results. By returning a confidence measure rather than cut-and-dried answers, our algorithm provides more context to users and further engender trust in the analytic tool.

We start by briefly describing k-Shape (Section II), a time-series clustering algorithm [21], [22]. Then, we show how k-ShapeStream enables k-Shape to (i) operate over streaming data; (ii) produce probabilistic interpretable results; and (iii) separate outliers from data (Section III). Finally, we demonstrate the effectiveness of k-ShapeStream on events detected in voltage magnitude measurements from an operational distribution network in California (Sections IV-V).

II. Background: k-Shape Time-Series Clustering

k-Shape is a time-series clustering algorithm that has achieved state-of-the-art performance across a multitude of domains [22], including the energy sector [23]–[25]. Similarly to k-means [26], k-Shape segregates data points into k clusters—with k, specified by the user—by iteratively maximizing intra-cluster similarity and ultimately returns cluster members and representative *centroids*. k-means uses the Euclidean distance to compare data points and computes the centroid of a cluster as the arithmetic mean of all its members' points. The distinctive differences of k-shape are its use of a normalized version of cross-correlation as the similarity measure between time series, termed *Shape-based Distance (SBD)*, and an eigen-decomposition-based method for centroid computation. These modifications are especially suited to time-series measurements. SBD is intuitive, robust to time-series scaling and misalignment, and can be efficiently computed via Fast Fourier Transform (FFT) [27]. While the arithmetic mean for centroid computation tends to have a low pass effect, eigen-decomposition preserves sharp time-series signatures better and, therefore, produces more representative centroids. Together, these features make k-Shape an attractive algorithm for clustering time-series grid events.

III. Clustering Streams of Time Series

k-Shape requires access to the entire set of time series and becomes prohibitively expensive to operate in streaming settings due to the need to re-cluster new and historic data. To alleviate that critical issue, we develop k-ShapeStream. k-ShapeStream clustering begins by initializing k clusters. These clusters are updated with each new event data set (hence this is *streaming* clustering). Specifically, in round r of clustering, n_r time series of length t, contained in the n_r-by-t data matrix X_r, are added to the existing clusters. Associated with each cluster are six parameters: a scalar cumulative member count, a t-by-1 centroid, a t-by-t shape matrix, as well as three parameters to parametrize the distribution of distances between the time-series cluster members and the cluster centroid. These are a scalar mean, a scalar standard deviation, and a scalar squared mean. For cluster j at the *end* of round r, the six parameters are denoted as $m_r(j)$, $u_r(j)$, $S_r(j)$, $\mu_r(j)$, $\sigma_r(j)$, $\delta_r(i)$, respectively. These parameters are efficiently updated in each round and are the *only* data carried forward between rounds. A length n_r list of indices indicating the cluster assignment of time series in X_r—denoted IDX_r—is also returned after each round. Fundamental to the streaming approach is that shape matrices—from which cluster centroids are extracted via eigen-decomposition—can be linearly updated with each new round of data, allowing centroids to reflect the entire set of cluster member without accessing data from prior rounds.

k-ShapeStream assigns each time series in data matrix X_r either to one of the k clusters, or to an *outlier* set, based on the normalized cross-correlation distance between the time series and the centroid of each cluster. The assignment depends on the distribution of distances between the existing centroid and members of the cluster. The distribution is assumed to be Gaussian, and, for cluster j, is fully parameterized by mean $\mu_{r-1}(j)$ and standard deviation $\sigma_{r-1}(j)$. Therefore, time series i is assigned to a cluster or to the outlier set as follows:

$$dist_i(j) \triangleq \frac{SBD(X_r(i), u_r(j)) - \mu_{r-1}(j)}{\sigma_{r-1}(j)}$$

$$IDX_r(i) = \begin{cases} arg\min_j dist_i(j) & \text{if } \min_j dist_i(j) < \tau \\ outlier & \text{otherwise} \end{cases}$$

τ is a user-set threshold of the number of σ's of permissible deviation. A typical choice—used in this work—is $\tau = 2$. By labeling outliers, k-ShapeStream allows unusual or unfamiliar events to be flagged for analysis and also avoids cluster contamination by outliers. Within round r, cluster memberships are iteratively refined, either until they have stabilized or until the maximum number of iterations has been reached. At the end of the round, cluster parameters are updated based on the final assignments of time series in X_r. Now, X_r can be completely discarded; all pertinent information for the next round is captured in the cluster parameters. This makes the method sustainable for streams of indefinite duration.

Pseudocode for k-ShapeStream is provided in A1-3. The SBD function in the pseudocode returns the shape based distance and aligned time series. SBD is fully described in [21] and has achieved state-of-the-art accuracy and runtime performance [28]. The time series must initially be z-normalized, as described in [29]. The updates of the intra-cluster distance statistics are described in A3. Notice that to update the standard deviation of intra-cluster distances, we must keep track of the mean of intra-cluster distances and the mean of squared intra-cluster distances. Motivated by an maximum likelihood approach, we choose to use a smoothing factor when updating the standard deviation to capture increasing certainty in the distribution parameters with increasing number of cluster members [30]. Importantly, the cluster parameters are updated using only the earlier parameters but none of the

A 1: $[IDX_r, C_r] = k$-ShapeStream$(X_r, C_{r-1}(j))$

Input : X_r is an n_r-by-t matrix containing n_r z-normalized time series of length t.
C_{r-1} contains cluster parameters from the prior round.

Output : IDX_r is an n_r-by-1 vector containing the assignment of n_r time series to k clusters.
C_r contains cluster parameters at the end of this round.

$u_r \leftarrow C_{r-1}.u$ `// prior centroids`
$\mu_r \leftarrow C_{r-1}.\mu$, $\sigma_r \leftarrow C_{r-1}.\sigma$ `// prior params`
$iter \leftarrow 0$, $IDX'_r \leftarrow []$, $S_r \leftarrow []$
$mindist \leftarrow \mathbf{0}$ `//` n_r`-by-1 zeros vector`
while $IDX'_r \neq IDX_r$ & $iter < 100$ **do**
  $IDX'_r \leftarrow IDX_r$
  `// Refinement`
  for $j \leftarrow 1$ **to** k **do**
    $X' \leftarrow []$
    for $i \leftarrow 1$ **to** n_r **do**
      if $IDX_r(i) = j$ **then**
        $X' \leftarrow [X', X_r(i)]$
      end
    end
    $[C_r.u_r(j), C_r.S_r(j)] \leftarrow$ ShapeExtraction(X', $C_{r-1}(j)$)
  end
  `// Assignment`
  for $i \leftarrow 1$ **to** n_r **do**
    $mindist(i) \leftarrow \infty$
    for $j \leftarrow 1$ **to** k **do**
      $[d, x'] \leftarrow SBD(u_r(j),\ X_r(i))$
      $dist \leftarrow \frac{|d-\mu_{r-1}(j)|}{\sigma_{r-1}(j)}$
      if $dist < mindist(i)$ **then**
        $mindist(i) \leftarrow dist$, $IDX_r(i) \leftarrow j$
      end
    end
    if $mindist(i) > \tau$ **then**
      $IDX_r(i) \leftarrow k+1$
    end
  end
  $iter \leftarrow iter + 1$
  $[C_r.\mu_r, C_r.\sigma_r, C_r.\delta_r]$ = UpdateStats(X_r, IDX_r, $mindist$, C_{r-1})
end
`// Update counts`
$C_r.m_r \leftarrow C_{r-1}.m$
for $i \leftarrow 1$ **to** n_r **do**
  $m_r(X_r(i)) = m_r(X_r(i)) + 1$
end

A 2: $[u, S'] =$ ShapeExtraction(X, C)

Input : X is an n-by-t matrix of z-normalized time series
C cluster parameters of interest.

Output : u' is new t-by-1 centroid.
S' is new t-by-t shape matrix.

$X' \leftarrow []$
for $i \leftarrow 1$ **to** n **do**
  [dist, x'] $\leftarrow$ SBD(C.u, $X(i)$)
  $X' \leftarrow [X', x']$
end
$S' \leftarrow X'^T \cdot X' + C.S$ `// incrementally updated`
$Q \leftarrow I - \frac{1}{t} \cdot O$ `//` I, O `are identity & ones matrices respectively`
$M \leftarrow Q^T \cdot S' \cdot Q$
$u' \leftarrow eig(M, 1)$

A 3: $[\mu_r, \sigma_r, \delta_r] =$ NewStats$(IDX_r, mindist, C_{r-1})$

Input : X_r, IDX_r, $mind$, C_{r-1} as defined in Alg. 1.

Output : $\mu_r, \sigma_r, \delta_r$ are k-by-1 vectors of new scalar cluster means, std. devs, and squared means respectively

for $j \leftarrow 1$ **to** k **do**
  $m_{r-1} \leftarrow C_{r-1}.m(j)$
  $count \leftarrow 0$, $s \leftarrow 0$, $ss \leftarrow 0$
  for $i \leftarrow 1$ **to** n_r **do**
    if $IDX_r(i) = j$ **then**
      $count = count + 1$
      $s = s + mindist(i)$
      $ss = ss + mindist(i)^2$
    end
  end
  $\mu_r(j) \leftarrow \frac{m_{r-1} \cdot C_{r-1}.\mu(j) + s}{m_{r-1} + count}$
  $\delta_r(j) \leftarrow \frac{m_{r-1} \cdot C_{r-1}.\delta(j) + ss}{m_{r-1} + count}$
  $\alpha \leftarrow \frac{m_{r-1} + count}{1 + m_{r-1} + count}$ `// Std. Deviation smoothing`
  $\sigma_r(j) \leftarrow \alpha\sqrt{\delta_r(j) - \mu_r(j)^2} + (1 - \alpha)$
end

prior cluster members. This is the fundamental benefit of the streaming approach.

IV. Demonstration

The algorithm is demonstrated on open-source voltage magnitude measurements from a single PMU on an operational distribution feeder in California, accessed through the NI4AI project platform[1] [31]. Similar to [18], [32], we define event points as sharp, significant changes in voltage magnitudes and extract a window of 2 seconds (240 samples) around each event point from the measurement stream. These time series are the inputs to k-ShapeStream. Note that k-ShapeStream is for post-detection event analysis: any event detection method can be used to find the events that are passed to k-ShapeStream. We use a simple approach, but there are a multitude of other options [6], [33]–[35].

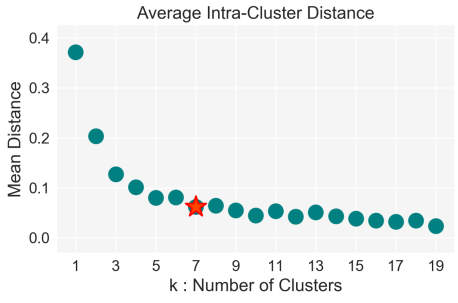

Fig. 1. Average intra-cluster distance for different choices of k. This analysis on the *first* batch of data is useful for choosing k: here we choose $k = 7$, just after the "knee" of the curve.

[1]https://ni4ai.org/

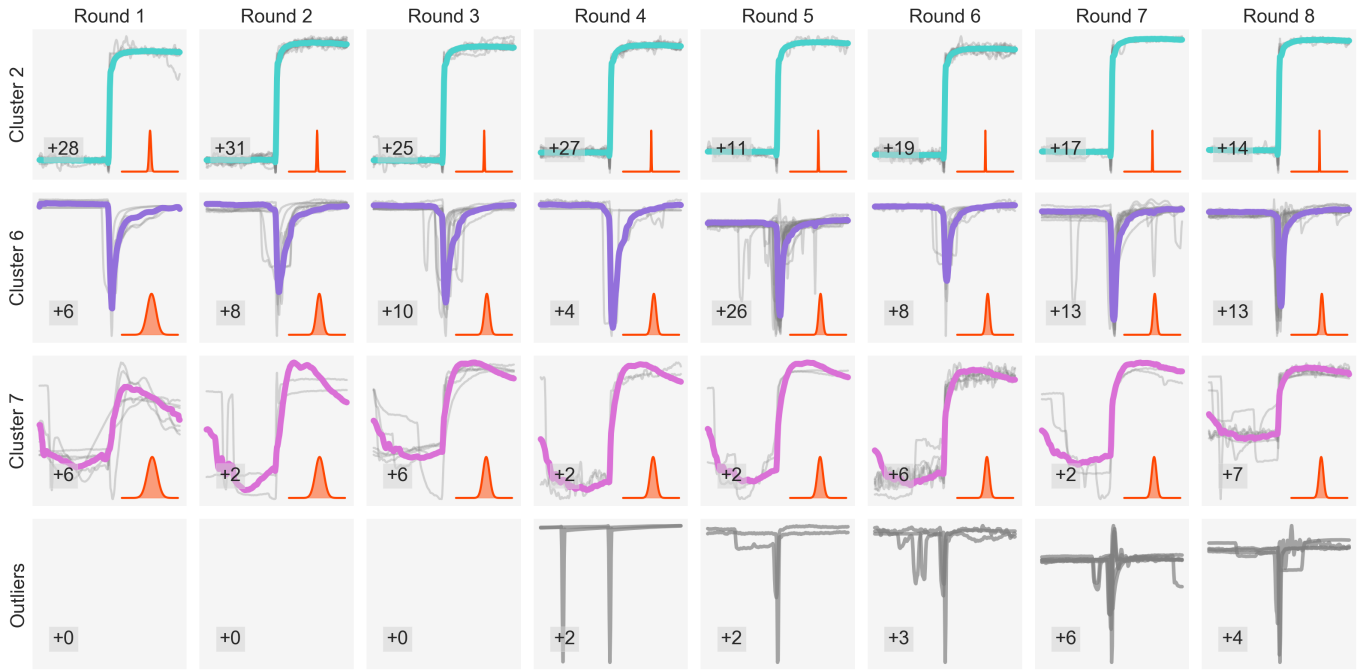

Fig. 2. Three clusters out of the seven and the set of outliers visualized over eight rounds of clustering. Gray lines indicate individual events. Colored lines show the cluster centroids. The inset number shows the number of events added to the cluster in each round. The inset distribution visualizes how intra-cluster distance distributions evolve: notice the narrowing distribution representing increasing certainty in the event's signature.

To emulate a streaming situation, we cluster events in batches of 30 across a total of over 700 events from four months of data. For realism, we choose k based only on the first batch of data by clustering it with several values of k and considering the average intra-cluster distance—the average of the SBD between each time series and its centroid—for each choice of k. The results are shown in Fig. 1, based on which we choose $k = 7$, as it lies just after the "knee" of the curve. Fig. 2 shows three of the resulting clusters, along with the outlier set, over eight rounds of clustering.

V. Example Use Case

To showcase the utility of k-ShapeStream, we perform analyses on some recognizable clusters from the full set of seven. These examples are not meant to present technically novel methods for system monitoring. Instead, we hope to illustrate how k-ShapeStream enables identification and analysis of recurring grid events and can be easily integrated into an analysis workflow to support a human analyst.

A. Transformer Tap Events

Load tap changing transformers (LTCs), common at distribution substations between medium and low voltage, mechanically adjust the effective turns ratio between their primary and secondary coils. They periodically "tap" the voltage up or down to compensate for changing voltage drop due to load variation, thus maintaining customer voltages within permissible limits. LTC failures can be costly and highly disruptive, motivating transformer monitoring [37]. Analysis of LTC operation based on PMU data has been *manually* demonstrated in the past [38]. k-ShapeStream can be used to automatically identify LTC tap events. Two clusters found in the data showing sharp step changes in voltage clearly correspond to LTCs operating to step voltage up and down (Fig. 3(a)-(b)). Notice the narrowness of the intra-cluster distance distributions indicating the high regularity of the LTC signatures. Once the signatures are isolated, different features can be analyzed. We consider the pre-event voltage (Fig. 3(c)), voltage change (Fig. 3(d)), and time of occurrence (Fig. 3(e)). For this set of LTC operations, all these features seem normal: magnitudes are generally lower preceding a tap up operation than a tap down, the size of the voltage step is highly regular, and tap ups tend to occur later in the day while tap downs occur earlier (as we would expect under a typical residential feeder load profile). Such an analysis could reveal irregular transformer behavior. For example, the intra-cluster distance distribution found by k-ShapeStream could be used to reveal an anomalous LTC signature that might indicate incipient failure. Note that no prior knowledge whatsoever about LTCs was required for the algorithm to suggest the relevant clusters.

B. Voltage Sag Events

Voltage sags are large transient dips in a network voltage magnitude that can last from less than a cycle to several seconds. They may be caused by motor starts, equipment misoperation, or faults [39], including dangerous high-impedance faults that fail to trip overcurrent protection. Recurrent sags could be caused by repeated vegetation contact and indicate a fire hazard. Large, long, or frequent voltage sags are also

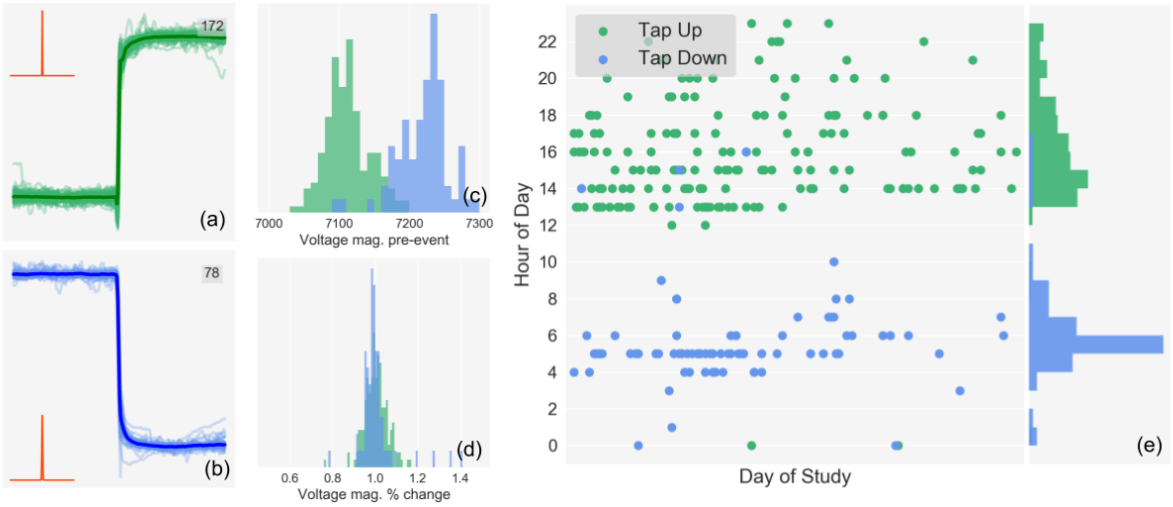

Fig. 3. Analyzing clusters containing LTC operation events. Tap up (a) and down (b) signatures clustered together across several months of data and multiple rounds of clustering. Distribution insets are very narrow indicating highly regular event signatures. (c) Voltage magnitude preceding tap event, showing lower magnitudes for tap up events and higher magnitudes for tap down events. (d) Histogram of percent change in voltage during event showing highly regular step size. (e) Occurrence of tap up and down events over study period, with histograms showing hourly distribution. Tap up events tend to occur later in the day while tap down events tend to occur earlier, as is expected under a typical residential load profile.

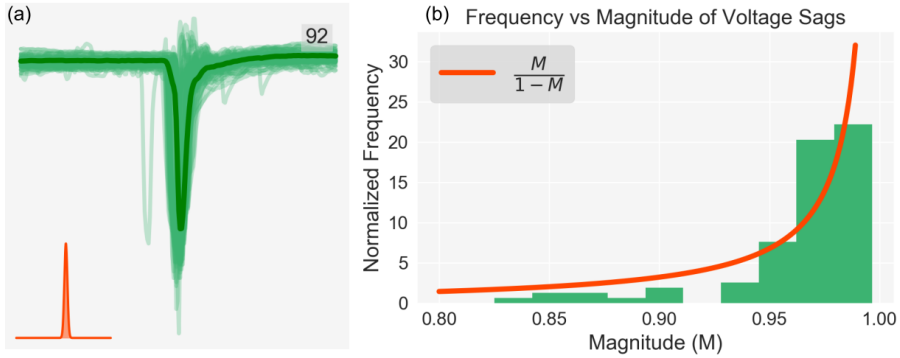

Fig. 4. Analyzing cluster containing voltage sags. (a) A cluster containing 92 voltage sag events found over several months of data and multiple rounds of clustering. The inset distribution of intra-cluster distances is wider than for the tap events in Fig. 3(a),(b) indicating less consistent event signatures. (b) Comparison of the empirical distribution of sag magnitudes in the cluster to a theoretical model suggested in [36].

problematic in that they may cause sensitive loads and solar PV inverters to trip offline. Monitoring voltage sags is therefore important for maintaining safety and reliability. We find one cluster containing a sharp, transient voltage drop that corresponds to a recurring voltage sag signature (4(a)). Notice from the intra-cluster distance distribution that while this event signature is quite regular, it is less so than the LTC tap signatures, as expected when comparing a random event to equipment operation. A voltage sag feature with implications for reliability is sag magnitude: the minimum voltage magnitude attained during the event. A result in the literature based on a simplified, general model of fault-induced sags posits that the normalized frequency of sags with magnitude M will be proportional to $\frac{M}{1-M}$ [36]. Fig. 4(b) compares this model to the empirical distribution of sag magnitudes in the cluster found by k-ShapeStream. The model appears to describe the empirical distribution quite well, indicating its efficacy for prediction and monitoring on this feeder. Again, k-ShapeStream produces a characterization of voltage sag type and frequency, and thereby generates possible insights into physical occurrences, in an entirely unsupervised learning process.

VI. Conclusion

The analyses of Sections V-A and V-B highlight the efficacy of k-ShapeStream for identifying recurring and unusual ("outlier") event signatures in grid data. Once identified by k-ShapeStream, these signatures can then be analyzed further to identify issues, understand system behaviour, and improve overall situational awareness. Without such a streaming clustering approach, event signatures would have to be identified manually, which is always time consuming and sometimes

impossible. Furthermore, k-ShapeStream generates highly intuitive results including a distribution that reflects the degree of confidence in a given cluster. These features make the algorithm particularly suitable for assisting and collaborating with a human user, which we believe is essential in the electric grid context.

REFERENCES

[1] Joseph W Mitchell. Power lines and catastrophic wildland fire in southern california. In *Proceedings of the 11th International Conference on Fire and Materials*, pages 225–238. Citeseer, 2009.

[2] Defense Use Case. Analysis of the cyber attack on the ukrainian power grid. *Electricity Information Sharing and Analysis Center (E-ISAC)*, 388, 2016.

[3] Reza Arghandeh, Merwin Brown, Alberto Del Rosso, Girish Ghatikar, Emma Stewart, Ali Vojdani, and Alexandra von Meier. The local team: Leveraging distributed resources to improve resilience. *IEEE Power and Energy Magazine*, 12(5):76–83, 2014.

[4] Alison Silverstein. Naspi and synchrophasor technology progress. In *NERC OC-PC Meetings*, 2013.

[5] Trip Doggett. Overcoming barriers to smart grids & new energy services. In *Proceedings of UT Smart Grid Conference, The University of Texas, Austin, ERCOT*, volume 7, 2011.

[6] Yuxun Zhou, Reza Arghandeh, Ioannis Konstantakopoulos, Shayaan Abdullah, Alexandra von Meier, and Costas J Spanos. Abnormal event detection with high resolution micro-pmu data. In *2016 Power Systems Computation Conference (PSCC)*, pages 1–7. IEEE, 2016.

[7] Milan Biswal, Sukumar M Brahma, and Huiping Cao. Supervisory protection and automated event diagnosis using pmu data. *IEEE Transactions on power delivery*, 31(4):1855–1863, 2016.

[8] Iman Niazazari and Hanif Livani. A pmu-data-driven disruptive event classification in distribution systems. *Electric Power Systems Research*, 157:251–260, 2018.

[9] Sai Akhil R Konakalla and Raymond A de Callafon. Feature based grid event classification from synchrophasor data. *Procedia Computer Science*, 108:1582–1591, 2017.

[10] Wenting Li, Meng Wang, and Joe H Chow. Fast event identification through subspace characterization of pmu data in power systems. In *2017 IEEE Power & Energy Society General Meeting*, pages 1–5. IEEE, 2017.

[11] Yixin Cai and Mo-Yuen Chow. Exploratory analysis of massive data for distribution fault diagnosis in smart grids. In *2009 IEEE Power & Energy Society General Meeting*, pages 1–6. IEEE, 2009.

[12] Alireza Shahsavari, Mohammad Farajollahi, Emma M Stewart, Ed Cortez, and Hamed Mohsenian-Rad. Situational awareness in distribution grid using micro-pmu data: A machine learning approach. *IEEE Transactions on Smart Grid*, 10(6):6167–6177, 2019.

[13] Manohar Mishra and Pravat Kumar Rout. Detection and classification of micro-grid faults based on hht and machine learning techniques. *IET Generation, Transmission & Distribution*, 12(2):388–397, 2017.

[14] Ravi Yadav, Shristi Raj, and Ashok Kumar Pradhan. Real-time event classification in power system with renewables using kernel density estimation and deep neural network. *IEEE Transactions on Smart Grid*, 10(6):6849–6859, 2019.

[15] Shan-lin Yang, Chao Shen, et al. A review of electric load classification in smart grid environment. *Renewable and Sustainable Energy Reviews*, 24:103–110, 2013.

[16] Akanksha Maurya, Alper Sinan Akyurek, Baris Aksanli, and Tajana Simunic Rosing. Time-series clustering for data analysis in smart grid. In *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 606–611. IEEE, 2016.

[17] Guoying Fan, Kunpeng Shi, Taiyi Zheng, Limin Feng, and Zhenyuan Li. Cluster analysis of grid-connected large scale wind farms. *Power System Technology*, 11:62–66, 2011.

[18] Daniel B Arnold, Ciaran Roberts, Omid Ardakanian, and Emma M Stewart. Synchrophasor data analytics in distribution grids. In *2017 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5. IEEE, 2017.

[19] Armin Aligholian, Alireza Shahsavari, Emma Stewart, Ed Cortez, and Hamed Mohsenian-Rad. Unsupervised event detection, clustering, and use case exposition in micro-pmu measurements. *arXiv preprint arXiv:2007.15237*, 2020.

[20] Eric Klinginsmith, Richard Barella, Xinghui Zhao, and Scott Wallace. Unsupervised clustering on pmu data for event characterization on smart grid. In *2016 5th International Conference on Smart Cities and Green ICT Systems (SMARTGREENS)*, pages 1–8. IEEE, 2016.

[21] John Paparrizos and Luis Gravano. k-shape: Efficient and accurate clustering of time series. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1855–1870, 2015.

[22] John Paparrizos and Luis Gravano. Fast and accurate time-series clustering. *ACM Transactions on Database Systems (TODS)*, 42(2):1–49, 2017.

[23] Fateme Fahiman, Sarah M Erfani, Sutharshan Rajasegarar, Marimuthu Palaniswami, and Christopher Leckie. Improving load forecasting based on deep learning and k-shape clustering. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 4134–4141. IEEE, 2017.

[24] Junjing Yang, Chao Ning, Chirag Deb, Fan Zhang, David Cheong, Siew Eang Lee, Chandra Sekhar, and Kwok Wai Tham. k-shape clustering algorithm for building energy usage patterns analysis and forecasting model accuracy improvement. *Energy and Buildings*, 146:27–37, 2017.

[25] Seyedeh Narjes Fallah, Ravinesh Chand Deo, Mohammad Shojafar, Mauro Conti, and Shahaboddin Shamshirband. Computational intelligence approaches for energy load forecasting in smart energy management grids: state of the art, future challenges, and research directions. *Energies*, 11(3):596, 2018.

[26] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[27] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

[28] John Paparrizos, Chunwei Liu, Aaron J Elmore, and Michael J Franklin. Debunking four long-standing misconceptions of time-series distance measures. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 1887–1905, 2020.

[29] Dina Q Goldin and Paris C Kanellakis. On similarity queries for time-series data: constraint specification and implementation. In *International Conference on Principles and Practice of Constraint Programming*, pages 137–153. Springer, 1995.

[30] Kevin P Murphy. Conjugate bayesian analysis of the gaussian distribution. *def*, $1(2\sigma 2)$:16, 2007.

[31] Sean Murphy, Kevin Jones, Theo Laughner, Mohini Bariya, and Alexandra Von Meier. Accelerating artificial intelligence on the grid. In *2020 Clemson University Power Systems Conference (PSC)*, pages 1–7. IEEE, 2020.

[32] Mohini Bariya, Sean Murphy, Kevin D. Jones, Theo Laughner, and Michael Andersen. Analytics at warp speed - from prototypes to production. In *CIGRE Grid of the Future*. CIGRE, August 2018.

[33] Do-In Kim, Tae Yoon Chun, Sung-Hwa Yoon, Gyul Lee, and Yong-June Shin. Wavelet-based event detection method using pmu data. *IEEE Transactions on Smart grid*, 8(3):1154–1162, 2015.

[34] Le Xie, Yang Chen, and PR Kumar. Dimensionality reduction of synchrophasor data for early event detection: Linearized analysis. *IEEE Transactions on Power Systems*, 29(6):2784–2794, 2014.

[35] Ti Xu and Thomas Overbye. Real-time event detection and feature extraction using pmu measurement data. In *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 265–270. IEEE, 2015.

[36] Math HJ Bollen. Voltage sags: effects, mitigation and prediction. *Power Engineering Journal*, 10(3):129–135, 1996.

[37] Rogier Jongen, Peter Morshuis, Johan Smit, Anton Janssen, and Edward Gulski. A statistical approach to processing power transformer failure data. In *19th International Conference on Electricity Distribution*, page 4, 2007.

[38] Ciaran Roberts, Anna Scaglione, Mahdi Jamei, Reinhard Gentz, Sean Peisert, Emma M Stewart, Chuck McParland, Alex McEachern, and Daniel Arnold. Learning behavior of distribution system discrete control devices for cyber-physical security. *IEEE Transactions on Smart Grid*, 11(1):749–761, 2019.

[39] Jovica V Milanovic, Myo Thu Aung, and CP Gupta. The influence of fault distribution on stochastic prediction of voltage sags. *IEEE Transactions on Power Delivery*, 20(1):278–285, 2005.