2015 IEEE 14th International Conference on Machine Learning and Applications

Detecting Erosion Events in Earth Dam and Levee Passive Seismic Data with Clustering

Wendy Belcher, Tracy Camp
Dept. of Electrical Engineering and Computer Science
Colorado School of Mines
Golden, CO USA
wbelcher@mines.edu, tcamp@mines.edu

Valeria V. Krzhizhanovskaya
University of Amsterdam, The Netherlands
St. Petersburg State Polytechnic University Russia
ITMO University, Russia
V.Krzhizhanovskaya@uva.nl

***Abstract*—Geophysical sensor technologies can be used to understand the structural integrity of Earth Dams and Levees (EDLs). We are part of an interdisciplinary team researching techniques for the advancement of EDL health monitoring and the automatic detection of internal erosion events. We present results from our performance study that uses signal processing, feature extraction, and unsupervised learning on passive seismic data from an experimental laboratory earth embankment. We used popular unsupervised clustering algorithms to gain insights to this real-world problem, and evaluated our results using internal and external validation techniques. In four of the clustering algorithms applied, results consistently show a clear separation of events from non-events. We provide proof of concept and an initial pattern recognition process that could be used as a tool for nonintrusive and long-term EDL monitoring.**

Keywords-earth levee, passive seismic, time series, pattern recognition, unsupervised clustering, machine learning, signal processing, geophysical

I. Introduction and Motivation

Earth Dams and Levees (EDLs) are built primarily for flood control, water storage, or irrigation and are constructed with earthen materials such as rock, sand, and clay [1]. Since many earth dams in the U.S. are over 60 years old [2], it is important to efficiently monitor the stability of these structures. The use of geophysical sensor technologies can be used to monitor and understand the structural integrity of EDLs.

Fig. 1: Earth dam failure by internal erosion (piping) [3]

Failures of EDLs are typically due to slope instability, piping, overtopping, or foundation issues [4]. Our study focuses on detecting the different stages of internal erosion. Erosion within the earthen material primarily begins with cracks, which allow seepage of water through the embankment. As more water flows or rushes through the earthen structure, the possibility for piping presents. Piping occurs when embankment particles accompany water flow and create pipes or voids under the earthen material. Occasionally self-healing occurs when the earthen material fills the void and blocks the water flow. When significant piping occurs, collapse (or catastrophic failure) of the earthen material follows (see Figure 1).

Current failure detection methods include visual inspection of the dam by a trained expert, which does not guarantee a levee failure is detected early enough to prevent its collapse. Through the application of machine learning techniques, we are working with an interdisciplinary research team to develop a process for the automatic detection of internal erosion events in EDL passive seismic data. Solving a real-world pattern recognition problem is difficult due to limited ground-truth information and a large amount of background or spurious noise events in the data.

Researchers have been able to detect anomalies in EDLs from sensors installed inside the dams (e.g., temperature, pore pressure, relative inclination) using a one-sided classification approach [5]. In other words, researchers detect deviations from what is considered a normal state of the dam. To our knowledge, there is no previous work in the field on using machine learning for the automatic detection of internal erosion events in passive seismic data. Our novel approach investigates detecting events that lead to failure by using geophysical data collected from sensors located on the surface, thereby retaining the integrity of the structure.

Due to limited availability of ground-truth information in EDL data, our study investigates the use of unsupervised learning, specifically clustering, which eliminates the need for levee specific labeled data. We present findings of our performance study that explores signal processing, feature extraction, and unsupervised clustering. We use a single long-length series of passive seismic data from an experimental laboratory earth embankment.

978-1-5090-0287-0/15 $31.00 © 2015 IEEE
DOI 10.1109/ICMLA.2015.9

II. Background and Related Work

With the amount and size of collected data increasing, efficient human or manual analysis becomes unmanageable. There is a need for automated data mining techniques to understand, i.e., find meaning and patterns in, "big data". Data mining combines techniques from several fields including machine learning, pattern recognition, and statistics. Wu et al. provides descriptions, extensions, issues, limitations, and impacts on some of the most influential data mining algorithms found in the literature [6]. The algorithms discussed fall into five broad categories: classification, clustering, statistical learning, association analysis, and link mining. Berkhin narrows down the topic of data mining to provide a detailed survey of different clustering techniques [7]. Xu et al. present a comprehensive survey of clustering algorithms and validation techniques for data sets appearing in statistics, computer science, and machine learning [8]. Our study focuses on unsupervised clustering, which is one of the most popular types of data mining used for finding hidden patterns in data.

Clustering algorithms group similar multi-dimensional data instances and predominantly fall into categories of partitioning and hierarchical methods [7]. These categories can then be further subdivided, e.g., density-based, grid-based and model-based [6] [8]. We define the two broad categories herein.

Partitioning is where the algorithm learns clusters by moving data points from one cluster to another until a stop condition is met. The stop condition is based on the optimization of an objective function (e.g., sum of the squared error). The iterative nature of partitioning algorithms guarantees their convergence; however, they may converge to a local instead of a global minimum. Partitioning works well with small to medium data sets and produces spherical (or other convex shaped) clusters. The main disadvantages of using the partitioning methods are the requirement to pre-select the number of clusters and they can be computationally inefficient with large data sets.

Hierarchical methods establish clusters gradually through either an agglomerative or divisive technique. Agglomerative starts with all points in separate singleton clusters and combines data points until the number of clusters is obtained or until all data points belong to one group. Divisive is the opposite concept, it starts with all data points combined in one cluster and divides until requested number of clusters are achieved. The development of clusters (or dividing) is based on a calculated similarity measure, commonly termed single, complete, or average linkage. An advantage of the hierarchical method is the results can be displayed in a dendrogram tree diagram and cut at the desired number of clusters. Disadvantages include slow run-time and the inability to backtrack (i.e., re-assign data points to different clusters).

Unsupervised learning is used in gaining insights to many real-world problems in fields such as bioinformatics, telecommunications, networking, computer vision, and seismology. We highlight related surveys and applications in these fields.

Chen et al. present analysis of using clustering algorithms in bioinformatics to analyze microarray gene expression data and explore methods for the extraction of meaningful biological information [9]. The overall best performing algorithms were k-means and partitioning around medoids (both of which are used in our study), and the worst was consistently the average linkage hierarchical. The limited or inconsistent results may stem from using a complicated biological data set or the quality measures selected for comparison. The authors state selection of the clustering method can be different for each data set and finding meaningful results is dependent upon the information desired.

Zhang et al. examine the use of unsupervised machine learning techniques in bioinformatics [10]. The authors aim to create an automatic detection and tracking system for the investigation of zebrafish larvae behavior to explore changes caused by chemicals, toxins, or genetic modification. The algorithms in the study include partitioning and agglomerative hierarchical methods (both of which are used in our study). Internal and stability validation performance metrics were used to determine the best performing algorithm. The hierarchical method was the best in grouping similar activities.

Hilas and Mastorocostas investigate the use of machine learning techniques for the identification of telecommunications fraud cases [11]. The goal was to provide an analysis of users' activity (or extreme changes) to find distinctions between legitimate and fraudulent behavior. Supervised binary classification using a neural network worked well; however, the technique did not reveal the distinct characteristics that separated the classes. The authors then examined the use of agglomerative hierarchical clustering (a technique used in our study) for a comparison. Dendrograms were used to display the results and possible clusters, yet there was no clear distinction between legitimate and fraudulent activity. The optimal result was found in the range of three to five clusters instead of the two manually identified clusters. The paper reveals some insights and limitations of using unsupervised learning for the complex problem of fraud detection.

Nguyen and Armitage describe their survey of using unsupervised, supervised, and hybrid approaches to machine learning for internet traffic classification [12]. Internet service providers (ISPs) are concerned with providing quality of service and automatically detecting fraud or abuse (similar to phone companies). Clustering results are provided for internet flow traffic using expectation maximization (which is used in our study), an unsupervised Bayesian classifier, and k-means (also used in our study). Overall, the methods are able to separate traffic based on type. The authors recommended using clustering as a first step to give insights into the data groupings, then add labels for classification.

Köhler et al. study the application of Self-Organizing Maps (SOMs) for data-driven feature selection, visualization and clustering using synthetic waveforms and real-world earthquake [13] and volcanic [14] data. The authors aim to provide an unsupervised learning approach to discover patterns and find a subset of features to represent the seismic data. The use of a SOM also allows for visualization of the data in a lower dimension (2D space) and provides correlations between features. The study finds a SOM can be an effective tool for initial learning steps and data inspection of seismic wavefields that could be used for supervised machine learning.

III. Passive Seismic Time Series Data

An experimental laboratory earth embankment, or "crack box" testbed (Figure 2), was equipped with geophysical instrumentation, and brought to failure by an interdisciplinary team to study internal erosion and cracking of embankment dams [15]. The structure was 7 cubic meters with a 2000-liter reservoir (Figure 2(c)) and a 2.7-meter long channel (Figure 2(b)) to move water from the reservoir to the embankment. The testbed was constructed over a hinged joint located in the bottom centerline that was used to induce the cracking. Figure 2(e) shows the result of inducing a 2.5 cm crack in the embankment.

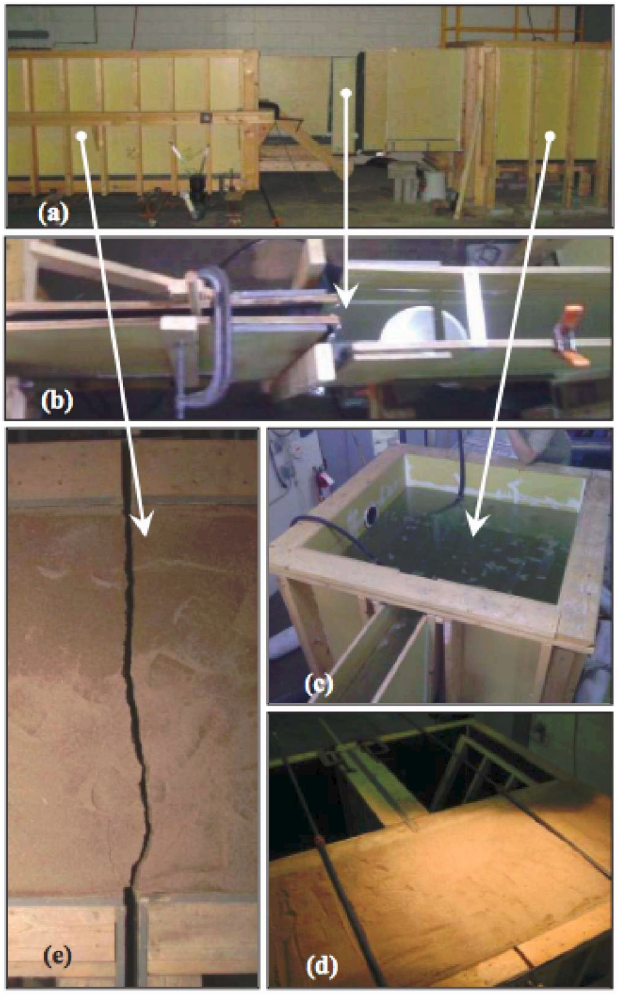


Fig. 2: Layout of the crack box testbed: (a) assembled model, (b) upstream channel, (c) constant head reservoir, (d) un-cracked filter, and (e) cracked filter (size=2.5 cm).

Amongst other sensing equipment, a vertical array of 12 geophones collected passive seismic data at 500Hz over several days and provided 4,140 seconds of data before, during, and after crack initiation. In addition to normal (baseline) activity, events such as cracking, pumping, flow, and spurious noise events were captured and used in our experiments. Detailed laboratory notes, photos, and videos were also provided and used in our research for ground truth documentation of the various events. Our study uses the data captured from a single sensor (sensor 6) located in the center of the vertical geophone array. Figure 3 displays the provided time series data for sensor 6 with pre-crack and highlighted crack events that were confirmed by the geotechnical engineers.

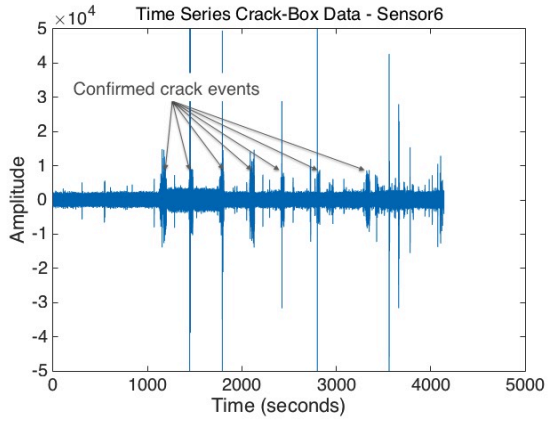


Fig. 3: The time series data for a single sensor with pre-crack and several confirmed crack events highlighted.

A. Frequency Analysis

Time-frequency analysis using a non-overlapping Fast Fourier Transform (FFT) shows events are separable and provide unique signatures within the spectra. The spectrogram in Figure 4 plots the data in the frequency domain to show the spectrum of the frequencies in the signal and how they change with time. Seismic data can be quite noisy; specifically, there is background and spurious noise caused by adjustment of equipment, people walking on the box, and nearby machinery.

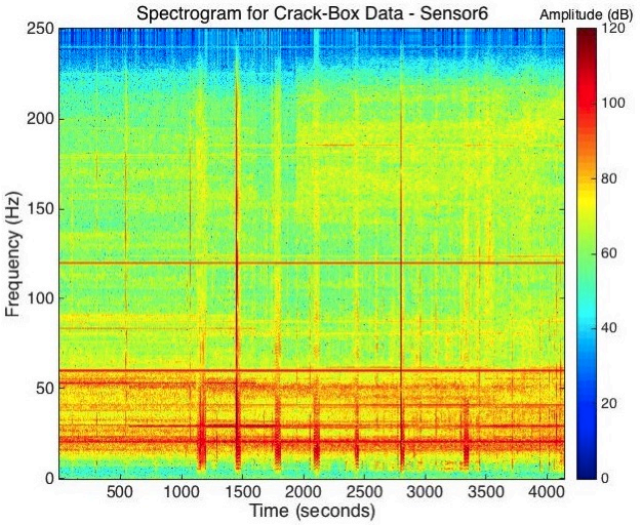


Fig. 4: Spectrogram shows seismic signals generated by various noises in the environment and concentrated activity for crack events.

Spectral frame decomposition divides the time series data into smaller blocks (or frames) for feature extraction. We explored 1, 2, 3, 5, and 10-second frame sizes; we discuss the results from this exploration in Section V. Figure 5 depicts a zoomed in section of the original time series data for the single sensor studied with a 10-second frame decomposition. The signal processing was performed using MATLAB and the open-source MIRToolbox [16].

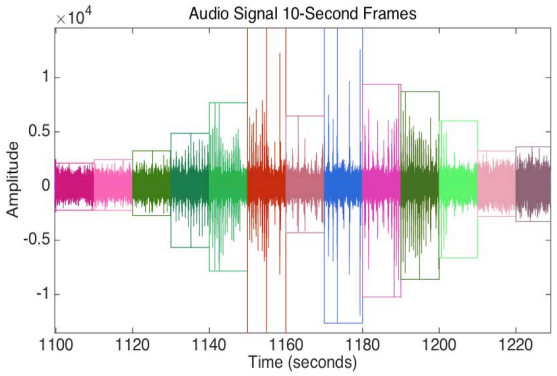


Fig. 5: Spectral frame decomposition divides the time series data into 10-second frames to use for the extraction of features.

TABLE I: Features extracted from each 10-second frame

Feature	Description
Zerocross	A temporal feature and an indicator of the noisiness of the signal or a count of the number of times the spectral signal crosses the zero axis (changes sign) [16].
Centroid	One of the statistical descriptors of spectral distribution and is the mean or geometric center of the spectrum. The centroid is usually associated with the brightness of a sound signal and is a measure of central tendency for the random variable [16].
Spread	A spectral feature that is a measure of the bandwidth of the spectrum. It can be used to describe the asymmetry and peakedness. The feature value is the standard deviation of the data distribution [16].
RMS	A temporal feature related to the loudness of the signal. The global energy of the signal x can be computed simply by taking the root of the average of the square of the amplitude [16].
85% Rolloff	Used to estimate the amount of high frequency in the signal by finding the frequency that a certain fraction (85%) of the total energy is below [16].

B. Feature Selection

We leverage previous work with passive seismic data to reduce the choices for selecting spectral features that best represent the data [17]. Features experimented with include several commonly used in audio signal processing. The five features identified and selected to use for our work are briefly described in Table I. Once a subset of the features were selected, we then reduced the dimensions by using principal component analysis (PCA) [18]. PCA converts a set of observations into principal components of a lower dimension to represent the variance within the data set. By applying PCA, we are able to reduce the data for analysis and visualization in lower dimensions; we discuss these results in Section V.

C. Data Normalization

The attributes were normalized to scale the feature values to [0,1]. Since clusters can be strongly influenced by the magnitudes of the variables, data normalization removes the bias. Figure 6 shows histograms for the normalized values of our five features with a 10 second frame size, providing 414 observations from our 4,140 seconds of passive seismic data.

The authors of [15] found that seismic events tend to have significant low frequency energy. Abrupt high energetic or sudden increase or decrease in seismic energy represents cracking, erosion, or concentrated flow events. Since our dataset included a mix of normal/baseline, crack, pump, and concentrated flow events, we see the values are generally distributed normally in zerocross, centroid and rolloff. We hypothesize, spread has a bimodal distribution (peaks twice) due to our experiment using the entire data set. The normal or baseline activity is represented in the first peak and higher energy events are represented in the second peak, with very few observations in the middle. When we consider just the first half of the data (more normal/baseline events) or the second half of the data (more cracking, pumping, and flow events), we observe each half has a unimodal distribution with the mean matching that of its respective peak in the entire set. Examination of the data distribution of the RMS feature reveals mostly low (near-zero) values accompanied by a few higher values; the results for RMS are what we expect since large sudden energetic bursts do not dominate the signal.

IV. Clustering with EDL Data

We investigate the use of unsupervised machine learning techniques to discover patterns or groupings of internal erosion events. We experimented with ten common unsupervised clustering algorithms including farthest first [19] (modeled after k-means with max distances for determination of assignments), simple expectation maximization, and variations of hierarchical clustering with average, complete, and single linkage. We have selected five of the best performing algorithms to apply to our EDL dataset. We chose algorithms that were efficient, worked well with large datasets, and had different techniques for the selection of cluster centers and boundary conditions.

Selection of initial cluster centers is typically accomplished using either a seed value for generating a random number, the first K (number of clusters) values in the data table, or by using a random selection of K values from the data. We used random selection and the k-means++ algorithm in our clustering experiments to compare different techniques. We note, however, that we found similar results over the 100 runs if all initial selections were random. Multiple runs of the clustering algorithm minimizes the effect of the random initial centroids, especially if using a small number of clusters $(K < 10)$, or by using a heuristic similar to the k-means++ algorithm [20].

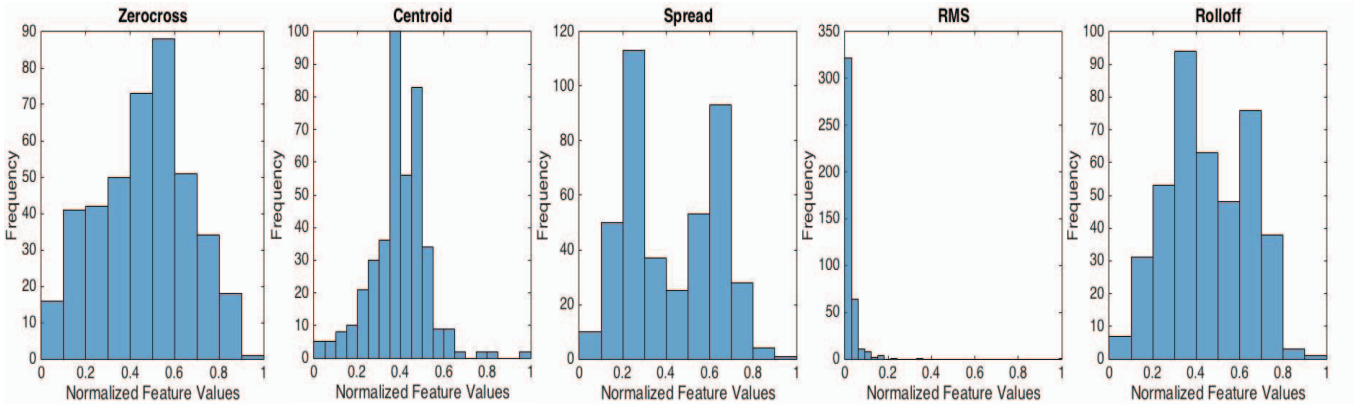


Fig. 6: Histograms depict normalized values for the five features (zerocross, centroid, spread, RMS, and 85% rolloff) with 10-second frames.

The best value of K is specific for each data set and is often ambiguous when using manual visualization. Comparing results from several repeated runs of the algorithm using different values of K is therefore needed. After repeated runs of each clustering algorithm using a range of the number of clusters between 1 and 10, we observed the ideal candidate for the number of clusters, whether 3, 5, or 10-second frames, was three ($K = 3$). We discuss the results from this analysis in Section V.

The machine learning algorithms were implemented using MATLAB Version R2015a and the Statistics, Machine Learning, and Fuzzy Logic toolboxes. We selected the preprocessed feature vectors with 3, 5, and 10-second frame decomposition as the input to each of the clustering algorithms and used cluster size $K = 3$. The five selected clustering algorithms are listed in Table II and briefly described next. Again, we experimented with ten common algorithms and found these five suited our goals.

TABLE II: Clustering algorithms used in our study

Algorithm	Category
K-Means Clustering (KM)	Partitioning
Hierarchical Clustering (HC)	Hierarchical
Gaussian Mixture Model (GMM)	Model based
Partitioning Around Medoids (PAM)	Partitioning
Fuzzy C-Means Clustering (FCM)	Partitioning

K-Means Clustering (KM) is one of the most simple unsupervised machine learning algorithms and is commonly used for the clustering of data. The k-means clustering method is based on the work by Stuart Lloyd [21]. The idea is to partition M points in D dimensions into K clusters while minimizing an objective function. We used the squared Euclidean distance measure and the k-means++ algorithm [20] to determine initial centroid selections for our experiments with the k-means algorithm.

Hierarchical Clustering (HC) works by building a hierarchy of clusters and is broadly defined in Section II. Our study applied the agglomerative hierarchical clustering technique using the Euclidean distance measure. The development of clusters was based on the Ward similarity measure [22], which uses an objective function to determine which pairs of data points to merge.

Gaussian Mixture Model (GMM) is a model-based clustering algorithm that, when used in conjunction with the expectation maximization technique (an iterative method for finding the maximum likelihood), often produces results comparable to the k-means algorithm. The cluster assignments in a Gaussian mixture model are based on membership scores to represent the probability that samples belong to a certain cluster. Our experiments with the Gaussian mixture model and the expectation maximization algorithm used a random initial selection of cluster centers.

Partitioning Around Medoids (PAM) is an iterative method commonly used for solving the k-medoids problem. The medoid is a member of the data set that is used instead of the mean or centroid and is the element with the least average dissimilarity to the other elements in the cluster (i.e., closest to the center). Our experiments used the city-block (Manhattan) distance measure; the k-means++ algorithm was used for initial selection of medoid positions.

Fuzzy C-Means Clustering (FCM) is a soft clustering algorithm similar in concept to the traditional hard clustering k-means algorithm, except a data item can belong to more than one cluster. Soft clustering is useful when the boundaries between the clusters are not well separated or unclear. In our experiments with the fuzzy c-means algorithm, the Euclidean distance measure was used and the initial cluster centers were selected randomly.

V. Comparison and Validation Metrics

We used provided ground truth data to compare cluster results to the original time series data. We validate events are separable with a minor overlap between event types. The output of the k-means clustering algorithm is plotted in Figures 7, 8 and 9. We applied PCA to reduce the dimensionality of the feature set to visualize the results of our clustering without compromising the meaning of the data. Figure 7 shows the plot of the first two principal components.

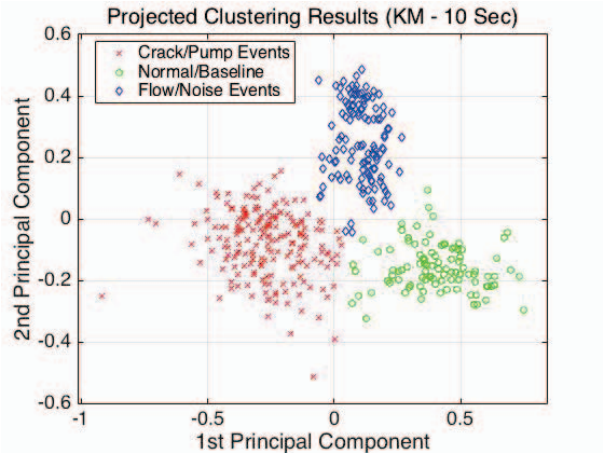
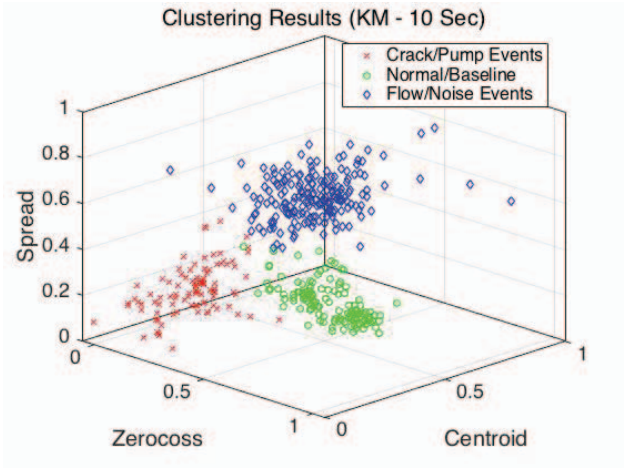


Fig. 7: Plot shows the projections for the first two principal components of the data with respect to the three separate event types found during k-means clustering ($K = 3$ and 10-second frames).

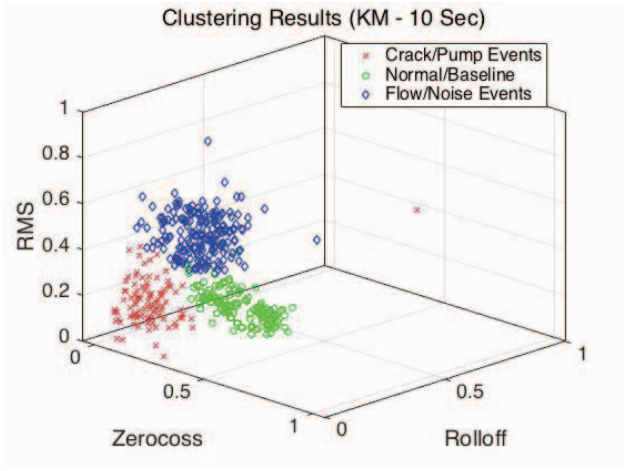
Figure 8 represents further visualization of the clustering results. We plotted three of the various raw feature value combinations in 3-dimensions to show the distinct clusters and intriguing artifacts in the data set. Looking at zerocross, centroid, and spread (Figure 8a), we see well-formed clusters that are separated in the feature space. The combination of zerocross, rolloff, and RMS (Figure 8b) show tight, well-separated clusters and expose interesting outliers in the data. In our future work (Section VI), we plan to explore outlier detection and removal techniques; however, even with the current outliers remaining, it is still clear events are separable. The plot for zerocross, spread, and rolloff (Figure 8c) allows us to see that rolloff does not have the same impact as the other features on the cluster formation. We chose the features to be used in our study based on prior work; we plan to investigate how to automatically select robust features in future work (Section VI).

Figure 9 is provided as a comparison between the original time-series data (see Figure 3) and the cluster assignments produced by the k-means clustering algorithm. The result of visualizing the color overlay reinforces the identification of event types from our experiments. Plots for the other clustering algorithms in our study produced similar results, but are omitted for brevity.

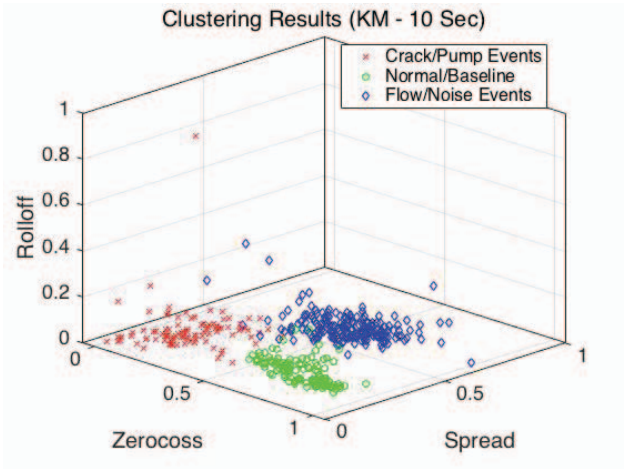
In order to further validate our results, and determine the effectiveness of the cluster assignments, the performance of the five selected machine learning algorithms was evaluated using common internal and external validation measures: purity [23] and silhouette width [24]. We discuss these two common validation measures in the rest of this section. Purity is an external validation metric used to determine cluster quality against known ground truth labels. The purity measure is not typically practical for a large unlabeled dataset; however, since we were able to hand-label the 4,140 seconds of our experimental data using provided lab notes, it was possible to calculate purity for our experiment. The resultant values represent the percentage of correctly clustered items and fall in the range [0, 1]; higher values are associated with better clusterings.



(a)



(b)



(c)

Fig. 8: Plots depict results for three sample combinations of three of the five features with groupings of events from k-means clustering ($K = 3$ and 10-second frames): (a) zerocross, centroid, and spread, (b) zerocross, rolloff, and RMS, and (c) zerocross, spread, and rolloff. Plots including additional feature value combinations also result in a clear separation of event types.

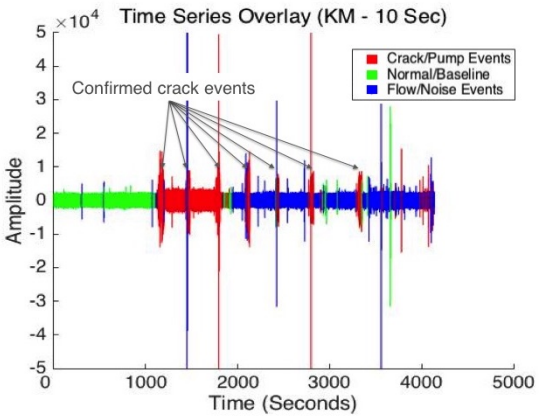


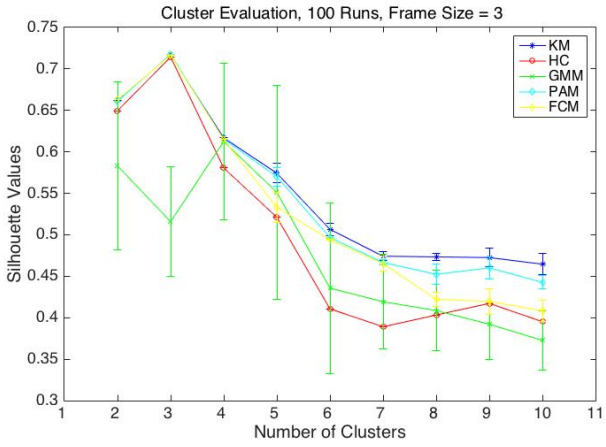
Fig. 9: Time series data from a single sensor with cluster color overlay highlights normal/baseline (green), flow/noise (blue), and confirmed crack/pump events (red).

Table III details the results averaged over 100 repeated runs for each of the five clustering algorithms in our study using 3, 5, and 10-second frame decomposition. The algorithms consistently perform well, i.e., over 0.823, purity values, with the exception of the Gaussian mixture model. The slightly lower values of 0.715, 0.728, and 0.765 purity for the Gaussian mixture model matches the lower average performance we found with silhouette width values.

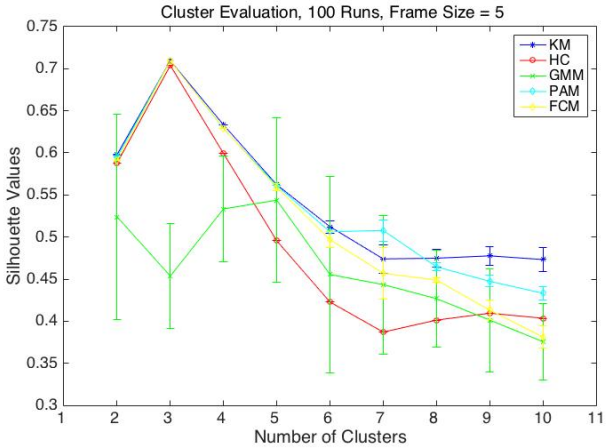
TABLE III: Purity Results

Frame Size	Obs.	KM	HC	GMM	PAM	FCM
3 Seconds	1380	0.830	0.827	0.715	0.825	0.826
5 Seconds	828	0.825	0.829	0.728	0.823	0.826
10 Seconds	414	0.831	0.838	0.765	0.829	0.831

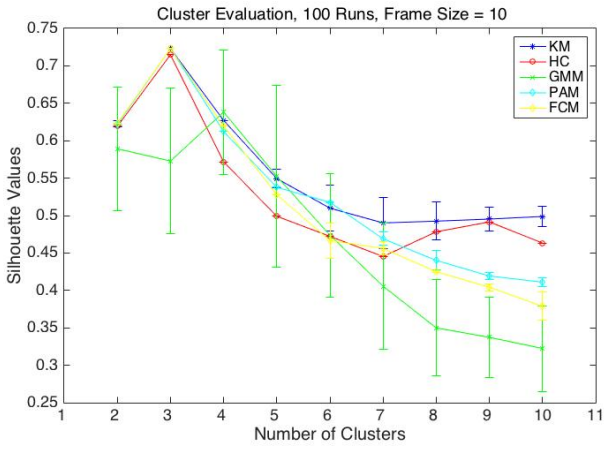
Silhouette width is an internal validation technique that measures the degree of confidence of cluster assignment (i.e., tightness of the cluster groupings or how well separated the clusters are). The silhouette values fall in the range [-1, 1]. The higher values (closer to 1) indicate a good assignment of each data point to its selected cluster and lower values (closer to -1) indicate the opposite. When these values are averaged over the entire dataset, the best performing number of clusters can be observed. Figure 10 shows our results for silhouette width averaged over 100 repeated runs (with 95% confidence intervals) of the five selected unsupervised learning algorithms in our study. The clustering algorithms examined generally produced the best silhouette width value at cluster size $K = 3$. Overall, the k-means most consistently had the highest silhouette width values, with similar performance from partitioning around medoids for $K < 6$. We note the hierarchical method produced low silhouette width values for $K > 6$ with the 3 and 5-second frame sizes (Figures 10a and 10b). Another observation is the Gaussian mixture model produced significantly lower silhouette width values than the other algorithms for $K = 3$ (Figures 10a, 10b, and 10c). We also note that the confidence intervals for the Gaussian mixture model are quite large, indicating the variability in the algorithm's results.



(a)



(b)



(c)

Fig. 10: Silhouette width results averaged from 100 repeated runs (with 95% confidence intervals) of the five clustering algorithms in our study for frame sizes: (a) 3, (b) 5, and (c) 10 seconds. A higher silhouette width value indicates better performance. The clustering algorithms generally suggest the best candidate for number of clusters to be three $(K = 3)$.

VI. Conclusions and Future Work

We present promising results of applying unsupervised clustering algorithms to gain insights to a real-world problem, i.e., Earth Dam and Levee (EDL) health monitoring. We provide proof of concept and an initial pattern recognition process that could be used as a tool for nonintrusive long-term earth dam and levee monitoring. In four of the five unsupervised clustering algorithms applied, results show a clear separation of events from non-events.

We plan to continue to apply various machine-learning techniques to EDL geophysical data to determine how we can best detect internal erosion events, such as anomaly detection. Our future goals also include experimenting with: de-noising of the data and outlier detection/removal, adaptive windowing for frame decomposition, and automatic analysis and selection of relevant features.

To fully test our process, we must also produce results on many different types of earth dam and levee data. Thus, we plan to apply our process to additional full-scale test and real-world passive seismic levee data that has been collected by a team of geoscientists and provided for our experimentation. For example, we have data from the IJkdijk full-scale test embankment located in Booneschans, Netherlands [25] that was constructed and equipped to study seepage and internal erosion in the fall of 2013 [26] [27]. We also have seepage and erosion data from the real-world Colijnsplaat levee in the Netherlands [28] from the fall of 2014. In addition, we participated and collected passive seismic data from an interdisciplinary piping experiment on a laboratory earth embankment at the United States Bureau of Reclamation in the summer of 2015.

Acknowledgment

This work is supported in part by National Science Foundation Grant OISE-1243539. We would also like to thank Dr. Marc J. Rubin, Oregon State University Cascades, for his key conversations during the beginning of this research.

References

[1] "US Bureau of Reclamation," http://www.usbr.gov, accessed: 2015-3-3.

[2] "Aging water resource infrastructure in the United States," http://www.usbr.gov/newsroom/testimony/detail.cfm?RecordID=2441, accessed: 2015-3-3.

[3] "Internal erosion (piping)," http://hyd.uod.ac/internal-erosion-piping/, accessed: 2015-3-3.

[4] M. Foster, R. Fell, and M. Spannagle, "The statistics of embankment dam failures and accidents," *Canadian Geotechnical Journal*, vol. 37, no. 5, pp. 1000–1024, 2000.

[5] A. Pyayt, I. Mokhov, A. Kozionov, V. Kusherbaeva, N. Melnikova, V. Krzhizhanovskaya, and R. Meijer, "Artificial intelligence and finite element modelling for monitoring flood defence structures," *Proceedings of the IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS)*, 2011.

[6] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, "Top 10 algorithms in data mining," *Knowledge and Information Systems*, vol. 14, no. 1, pp. 1–37, 2008.

[7] P. Berkhin, "A survey of clustering data mining techniques," in *Grouping Multidimensional Data*. Springer, 2006, pp. 25–71.

[8] R. Xu, D. Wunsch *et al.*, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[9] G. Chen, S. A. Jaradat, N. Banerjee, T. S. Tanaka, M. S. Ko, and M. Q. Zhang, "Evaluation and comparison of clustering algorithms in analyzing ES cell gene expression data," *Statistica Sinica*, vol. 12, no. 1, pp. 241–262, 2002.

[10] H. Zhang, S. C. Lenaghan, M. H. Connolly, and L. E. Parker, "Zebrafish larva locomotor activity analysis using machine learning techniques," *Proceedings of the 2013 12th International Conference on Machine Learning and Applications*, vol. 1, pp. 161–166, 2013.

[11] C. S. Hilas and P. A. Mastorocostas, "An application of supervised and unsupervised learning approaches to telecommunications fraud detection," *Knowledge-Based Systems*, vol. 21, no. 7, pp. 721–726, 2008.

[12] T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.

[13] A. Köhler, M. Ohrnberger, and F. Scherbaum, "Unsupervised feature selection and general pattern discovery using self-organizing maps for gaining insights into the nature of seismic wavefields," *Computers & Geosciences*, vol. 35, no. 9, pp. 1757–1767, 2009.

[14] A. Köhler, M. Ohrnberger, and F. Scherbaum, "Unsupervised pattern recognition in continuous seismic wavefield records using self-organizing maps," *Geophysical Journal International*, vol. 182, no. 3, pp. 1619–1630, 2010.

[15] R. V. Rinehart, M. L. Parekh, J. B. Rittgers, M. A. Mooney, and A. Revil, "Preliminary implementation of geophysical techniques to monitor embankment dam filter cracking at the laboratory scale," *Proceedings of the 6th Annual International Conference on Software Engineering (ICSE)*, 2012.

[16] O. Lartillot and P. Toiviainen, "A matlab toolbox for musical feature extraction from audio," *Proceedings of the 10th International Conference on Digital Audio Effects*, 2007.

[17] M. J. Rubin, "Efficient and automatic wireless geohazard monitoring," Ph.D. dissertation, Colorado School of Mines, 2014.

[18] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2002.

[19] D. S. Hochbaum and D. B. Shmoys, "A best possible heuristic for the k-center problem," *Mathematics of Operations Research*, vol. 10, no. 2, pp. 180–184, 1985.

[20] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1027–1035, 2007.

[21] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[22] J. H. Ward Jr, "Hierarchical grouping to optimize an objective function," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 236–244, 1963.

[23] C. D. Manning, P. Raghavan, H. Schütze *et al.*, *Introduction to Information Retrieval*. Cambridge University Press Cambridge, 2008, vol. 1.

[24] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.

[25] "Flood Control IJkdijk," http://www.floodcontrolijkdijk.nl/nl/, accessed: 2015-7-3.

[26] M. A. Mooney, M. L. Parekh, B. Lowry, J. Rittgers, J. Grasmick, A. R. Koelewijn, A. Revil, and W. Zhou, "Design and implementation of geophysical monitoring and remote sensing during a full scale embankment internal erosion test," *Proceedings of the GeoCongress*, 2014.

[27] J. Rittgers, A. Revil, T. Planes, M. Mooney, and A. Koelewijn, "4-D imaging of seepage in earthen embankments with time-lapse inversion of self-potential data constrained by acoustic emissions localization," *Geophysical Journal International*, vol. 200, no. 2, pp. 756–770, 2015.

[28] "Smart levee guideline," http://www.smartlevee.nl/projects-and-cases/totaallijst/colijnsplaat/, accessed: 2015-8-3.