TRAFFIC MONITORING WITH DISTRIBUTED ACOUSTIC SENSING

B. CHAN, E. R. KORFANTY, S. NATAJ, J. PARK, B. PENG, AND J. ZHANG

Industry mentor:
Matt McDonald, Chief Technology Officer, Fotech

Academic mentors:
Prof. Parimala Thulasiraman and Ying Ying Liu
Department of Computer Science, University of Manitoba

Abstract. Using vehicle traffic data collected with *Distributed Acoustic Sensing* (DAS) facilities from Fotech, we demonstrate that k-means clustering and Kalman filtering can be used to determine the locations of individual vehicles, and track the trajectories of vehicles over time. The implementation provided can handle traffic data for several vehicles travelling in the same direction, with no car passing another. We consider more complex traffic flow situations as future directions for improvement to our method.

1. Background and Overview

Smart city applications have experienced a notable increase in interest over the last few years. A smart city is a metropolis that employs a variety of electronic *Internet of things* sensors to collect data. The data is used to improve the operations, resources and services across the city [8]. One of the branches of smart city development consists of the development of real-time traffic monitoring with the aim of reducing traffic congestion. Processing of fiber-optic distributed acoustic sensing (DAS) data is in high demand within different smart city applications [15], [11]. DAS signal data measures the strain in fiber optic cables that are placed underground. For traffic monitoring applications, a fiber optic cable is run along a road. When vehicles pass by, mechanical vibrations are created in the ground, causing slight deformities in the fibre optic cable. These deformities cause a phase differences in the back-scattered light from laser pulses propagating through the fiber optic cable. The DAS system is able to detect these minute differences, and creates a signal linearly proportional to the strain in the cable at each point along its length [16]. DAS technologies have significant potential for traffic monitoring applications within a city. DAS has proven to be successful in the similar topic of train monitoring, though DAS signals from a road can be much more complex than DAS signals from trains. In [16], an algorithm was presented for using DAS signals to track the positions of trains over time, provided that there is a sufficiently large separation between trains

This work was funded by PIMS and Mitacs.

at all times. The algorithm applies machine learning techniques to detect the train edges, uses a distance-based optimization method for assigning train edges to train objects, and uses a Kalman filter to track the train objects over time. DAS signals from vehicles on a road, however, often appear in greater quantity, and in closer proximity than those from trains. This presents challenges in the application of this algorithm to vehicle traffic monitoring.

The goal of this project is to investigate a new methodology for identifying and tracking vehicles, by making use of clustering algorithms and Kalman filtering techniques in data mining and signal processing literature. We hope to provide a foundation which can be extended upon to handle more complex traffic flow settings. In Section 2 the proposed methodology for the solution of this problem is discussed. In Section 3, the result of implementing this approach is presented. DAS data is provided by Fotech, and Python libraries and packages are used for coding. In Section 4 the future direction and possible improvements are discussed. Finally, a summary of the project is given in Section 5.

2. Approach and Methodology

Our proposed methodology consists of three main steps: locating peaks in the DAS signals, using clustering methods to determine vehicle positions, and tracking these vehicle positions over time. Investigations relating to these three steps are presented in Section 3. Here, we provide an overview of the process by which these three steps can be combined into a real-time traffic monitoring method, summarized in Figure 1.

DAS signals are read one at a time. A single DAS signal reading is referred to as a *shot*, and consists of one sample from each position along the fiber optic cable, for a given instant in time. A fixed number of shots, which will be referred to as the *window size*, are analyzed at any given time. When a new shot is read, the analysis window is shifted by one time step, in order to include the most recent *window size* many shots.

Kalman filter models can be used to track vehicle positions and velocities over time. In the case of more than one vehicle, we propose the use of more than one Kalman filter model simultaneously. For each of the vehicles identified, a tracker object is created, and equipped with a Kalman filter model for that vehicle. These tracker objects will be referred to as *Kalman trackers*. Each Kalman tracker must be initialized with the position of the corresponding vehicle. Initial velocities and positions are set to null, and are corrected as new data becomes available in each time step.

In order to use DAS signal data to identify the positions of vehicles at a given point of time, peaks in the signal strength must first be located. The goal of this procedure is to separate the signals representing vehicles from background noise. The detected peaks can then be further analyzed to infer vehicle positions.

A clustering algorithm can be used to identify individual vehicles using the detected peaks data. Many clustering algorithms provide a point, or points, representative of each cluster. These are often referred to as *cluster centers*, or *cluster centroids*. Cluster centroids can be excellent candidates for vehicle positions.

To track the positions of identified vehicles, the locations of cluster centers can be used to update the Kalman trackers. If there is more than one Kalman tracker, an optimization problem must be solved in order to associate a given cluster center with an existing tracker. The strategy for assignment of cluster centers to trackers is an important aspect managing more than one Kalman tracker simultaneously.

Finally, the process of reading a shot, shifting the analysis window, finding and clustering peaks, and assigning cluster centers to trackers for trajectory correction, can be repeated in a loop for real-time tracking of vehicle positions.

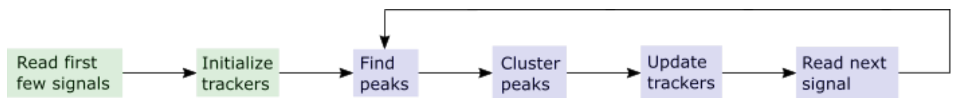


FIGURE 1. Flowchart of the proposed method for identifying and tracking vehicles from DAS signals in real-time.

3. Implementation and results

In our approach, we analysed small subsets of the data collected from Fotech instruments, to reduce the computational overhead. When comparing clustering algorithms, we looked at vehicle trajectories for both smaller subsets of 10 shots, and larger subsets of a few hundred shots. When using Kalman trackers to trace out vehicle trajectories, we used circular shifting, and analyzed 5 to 20 shots worth of data at a time. This technique is used to achieve the results in Section 3.3, where an example implementation is provided to illustrate how our methodology can be used to handle simultaneous tracking of several vehicles.

In preparation, we first detected peaks across the aforementioned segments of data, as described in Section 3.1. In section 3.2, clustering algorithms are employed to cluster the detected peaks. Furthermore, we discuss whether the resulting clusters align with our objectives. In particular, we investigate whether the resulting clusters effectively distinguish one vehicle from another. Using these observations, we make conclusions about which clustering methods may be more effective for identifying locations of vehicles from peaks in DAS signals.

3.1. **Peak detection techniques.**

The elastic strain wave found in each shot contains noise which is unnecessary for our purpose [10]. Hence, only the positions at which a compression wave obtains a local extremum exceeding a certain threshold, which we call a peak, were processed through the clustering methods presented in the Section 3.2. Once we found the peaks in each shot (Figure 2), we collected these peaks over the time period of interest (Figure 3a and 4a).

There are various peak detection methods available on Python `scipy` library:

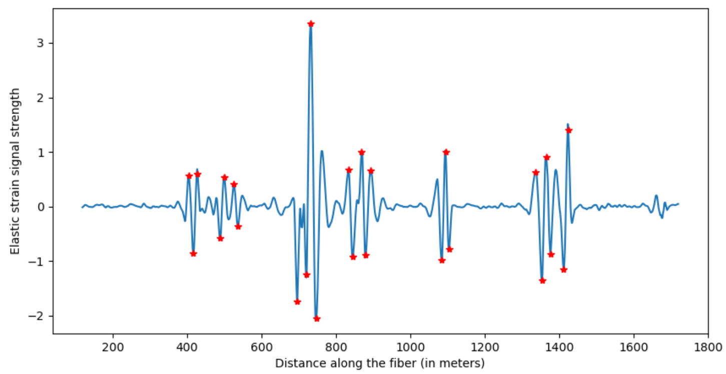
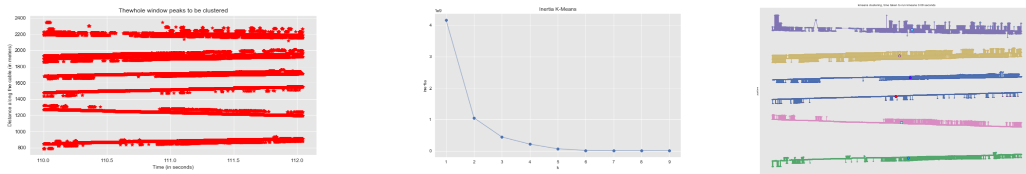


FIGURE 2. An example of the peaks (marked with red stars) on an elastic strain wave

- `scipy.signal.find_peaks_cwt`,
- `scipy.signal.find_peaks`,
- `scipy.signal.argrelextrema`.

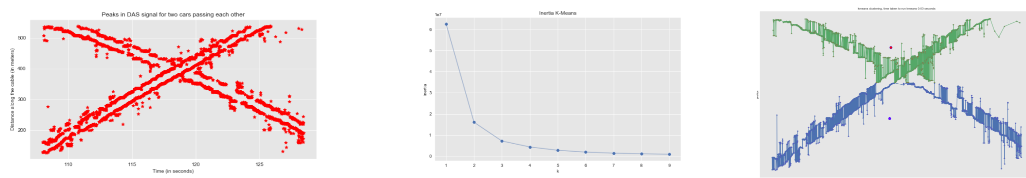
In our implementation, we used signal processing `scipy.signal.find_peaks_cwt` to find the peaks for the subsets of DAS signal data to be analyzed. This method is based on a wavelet transformation [4]. The algorithm performs the discrete convolution of the dilation and translation of the Ricker wavelet function with the raw data. We have tested various peak detection methods, including three listed above, but a thorough comparison between the peak detection methods is beyond the scope of this investigation, and is included in the future work for improving our implementation.

3.2. **Comparison of clustering algorithms.** After collecting the peaks data, we applied clustering techniques to partition the peaks data, in hopes of achieving partitions representative of individual vehicles. We required that the number k be equal to the number of cars in the data segment to be determined. The clustering used the data for the positions and times of peaks. We employed the Python scikit-learn library for implementing clustering algorithms [12]. One promising approach to clustering for this DAS application is k-means clustering. Although this algorithm is computationally intensive [6], implementing it is straightforward. The first difficulty in applying any clustering algorithm is to obtain a priori knowledge of the number of clusters in each section of streaming data. To overcome this challenge, we apply the algorithm in a loop to compare the performance of the model for different numbers of clusters. In other words, we run the algorithm for different values of k, starting at $k = 1$ and increasing k by one each iteration before every re-run. The algorithm stops when the inertia attributes of the model no longer decrease significantly (inertia is sum of squared distances of samples to their closest cluster center). Figure 3b depicts the result of applying k-means clustering in a loop to find the best number



(A) The data set (B) The loop (C) k-means clustering

FIGURE 3. k-means clustering of the data set containing of 6 vehicles passing the road in the same direction



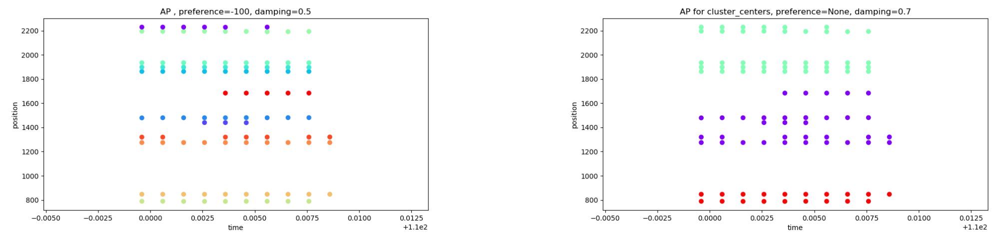
(A) The data set (B) The loop (C) k-means clustering

FIGURE 4. k-means clustering of the data set containing 2 vehicles passing the road in different directions

of clusters in the data set (Figure 3a). This data set represents six cars moving in same direction. Applying k-means in a loop shows that the best number of clusters is $k = 6$. Figure 3c shows that k-means algorithm successfully clusters the peak data into six clusters. After this, the centroids of the clusters can be used for in applying Kalman filters.

Another challenge in implementing k-means clustering in our data set is that the algorithm does not give an appropriate number of clusters for data segments presenting cars moving in different directions; it can not accurately distinguish vehicles passing each other on the road. This happened because k-means clustering usually cannot handle non-convex sets. By combining k-means with hierarchical clustering, we may overcome this difficulty [9]. Figure 4 shows the result of applying k-means in a data set with two vehicles passing other, for which it is not successful in finding the right number of clusters. However, this might be expected, because a pair of intersecting lines consists of a single connected component. More specifically, k-means is best suited to convex sets, but a pair of intersecting, thick lines is not convex.

Another potential algorithm to use for clustering in this DAS application is Affinity Propagation (AP). Unlike k-means, AP does not require the number of clusters to be determined before running the algorithm. This method finds *exemplars* (members of the input set that are representative of clusters). As an input, the algorithm requires some parameters to be provided: *similarity* (affinity function), *preference* and *damping factor*. Similarity defines how well-suited a point is to be the exemplar for another. Here we define it as negative Euclidean distance, which is the negative



(A) first time, using peak positions

(B) second time, using center samples

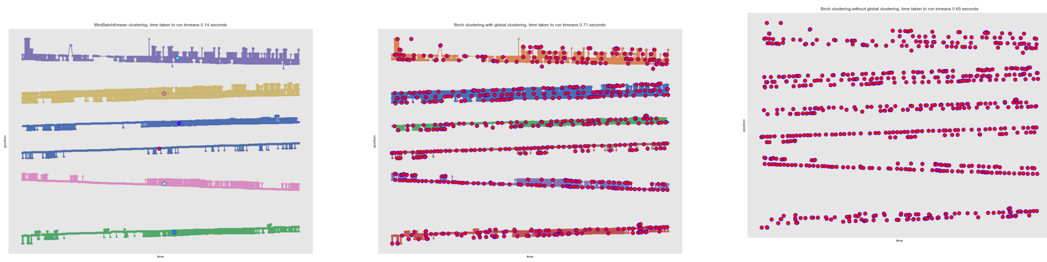
FIGURE 5. AP clustering of the data set containing 6 vehicles passing the road in the same direction

squared distance between the two data points. Preference represents the suitability of each data point to be an exemplar; points with larger preference values are more likely to be chosen as exemplars. Unfortunately, AP is very slow for a larger window size, such as the data shown in Figure 3a, due to the computational expense of the algorithm [13]. When examining AP, we apply it for a smaller data set. Figure 5a shows the result of applying AP in for a small window size, when the preference is any number between -10 and -100, and the damping factor is between 0.5 and 1. Since it gave more clusters than expected, we ran the algorithm a second time to merge clusters together using the cluster centers, presented in Figure 5b, which results in four clusters; this is still not representative of the number of vehicles, even for this small time window.

Some other clustering algorithms that could be useful in this DAS application are BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) and Mini Batch k-means methods. BIRCH applies hierarchical clustering on large data sets. This method is memory efficient; it can typically find a good clustering with a single scan of the data [17]. Mini Batch k-Means has less computational cost in comparison to k-means, as it uses random batches of data as opposed to using all available data [14]. The implementation of Mini Batch k-Means and BIRCH algorithms are available in the scikit-learn library [12]. The partial fit method of these models, included in the sklearn package, provides a way to do online clustering for steaming data. For references about online clustering, see [3] and the references therein. Figure 6 compares these two algorithms for the same data set as in Figure 3a.

Finally, it should be noted that for any of these clustering methods, tailoring the similarity measurement to this specific application may significantly improve results. For example, [7] provides a path-based similarity measure for AP. This, and the incorporation of velocities and accelerations, are potential improvements to the similarity measurements.

3.3. **The real-time process.** In this section, we provide an example to illustrate the feasibility of the methodology described in Section 2, making use of a circular shifting technique to mimic a real-time implementation. The proposed method was



(A) Mini Batch k-means (B) BIRCH (C) sub cluster centres in BIRCH

FIGURE 6. Mini Batch k-means and BIRCH clustering of the data set containing 6 vehicles passing the road in different directions

applied to a subset of DAS data consisting of five vehicles travelling along the fiber optic cable. The vehicles do no pass each other, and travel at similar velocities. The sample data consists of 5 seconds' worth of DAS signals measured over a 1529 meter length of fiber optic cable. For the particular DAS system, 5 seconds corresponds to 500 shots. A window size of 5 shots was used for analysis.

Peaks in the DAS signals were detected using `find_peaks_cwt`, and the method described in Section 3.1. Peaks for which the absolute value of the signal strength was less than 10 times the signal reading median were deemed to be noise, and were excluded from the computations.

The k-means clustering algorithm was used, with $k = 5$ set to match the number of vehicles. As illustrated in Section 3.2, the k-means algorithm was effective, efficient, and reliable. It was chosen for this implementation for these reasons, and to provide a clear example which can be used as a starting point for further investigations. See Section 4 for remarks on generalizing to a varying number of vehicles.

Five Kalman trackers were created, and the initial positions were set to be those of the cluster centers found from the peaks in the first 5 shots. The simultaneous use of more than one Kalman tracker comes with a challenge; namely, how does one decide which cluster center corresponds to which vehicle in subsequent time steps? In this example, the rule for deciding which cluster center should be used for updating which Kalman tracker was based on distance, and can be summarized as follows: If cluster center c is the closest cluster center to tracker t, and there is no other tracker t' which is closer to c than t, use c to correct tracker t. Otherwise, do not correct tracker t.

Figure 7 shows the results of this implementation. The five trackers are successful in tracing trajectories through collections of detected peaks corresponding to vehicles. This shows that a clustering-based approach to DAS signal data may provide a feasible solution for the tracking of vehicle positions and velocities. In particular, the use of cluster centers for the simultaneous management of several Kalman trackers was shown to be successful in this example, and the method has enough flexibility to serve as a starting point for treatment of more complex traffic flow situations.

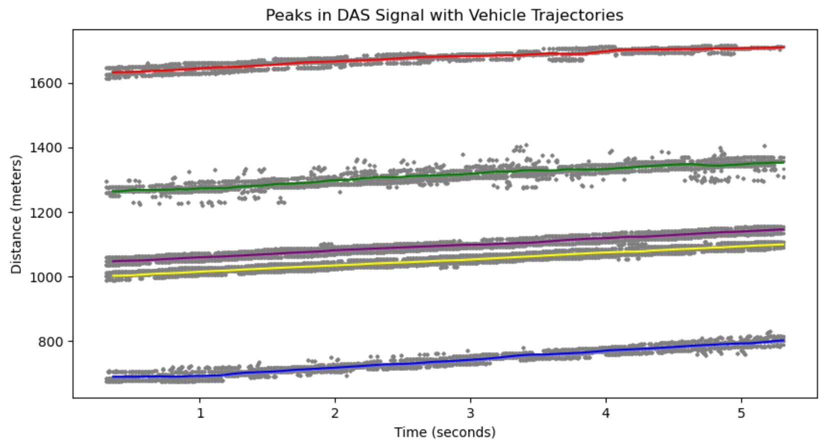


FIGURE 7. Trajectories of five vehicles tracked simultaneously using the proposed methodology. Detected peaks in DAS signal are plotted in grey, and the five colours show the Kalman tracker trajectories.

4. DISCUSSION

In Section 3.3, we successfully implemented the proposed methodology using k-means clustering for subsets of data with simple traffic flow; specifically, for a fixed number of vehicles moving along the length of the fiber optic cable, in one direction, with no vehicle passing another. There are many ways in which this example can be improved. The main area of flexibility is in the choice of clustering algorithm. It may be beneficial to compare the performance of this tracking method when implemented using different similarity measurements, online clustering methods, hierarchical clustering methods, and methods which do not require the number of clusters to be specified. The pre-processing and peak-finding techniques used may affect which clustering algorithms are the most successful when tracking, and should be considered in a thorough comparison. It may also be interesting to consider the use of Hough lines instead of peaks [1].

It is possible that the most effective clustering algorithm could be one that requires the number of clusters to be specified. A possible method for using such an algorithm with a changing number of cars is the following: If the current number of trackers is k, compare the results of finding $k-1$, k, and $k+1$ many clusters, and pick the best result. Further investigation is required to determine a method for comparing clustering results, and deciding which should be used.

Another clustering algorithm of interest is CURE [5]. The benefit of CURE is that it allows for more flexibility in the shapes and sizes of the clusters, and is also robust in the presence of outliers. This could make CURE a suitable clustering algorithm to use for tracking, as peak detection methods can result in outliers.

When vehicles pass each other, the trajectories intersect, appearing as the crossing of lines of peaks in DAS signal. The algorithm for correcting trackers provided in

Section 3.3 does not include logic to consistently achieve accurate tracking in this situation. A potential improvement to the algorithm would be to weight corrections to trackers more lightly when approaching an intersection of trajectories.

It is also necessary to allow for the creation and deletion of trackers. This would allow the method to be applied to a situations with a varying number of vehicles. One possible approach is the following: If a cluster center is not assigned to a tracker, create a new tracker starting from that position. If a tracker is not updated for a specified number of iterations, delete the tracker.

5. Conclusion

Among the new techniques rising with the smart city trend, DAS technology provides a feasible solution for monitoring of traffic and transportation. In this work, we investigated a methodology for identifying positions of vehicles from DAS signal data, and tracking the positions and velocities of vehicles over time. The method for identifying positions was based on the application of clustering algorithms in the data mining literature to peaks in DAS signal, and the method for tracking the positions was based on the simultaneous use of several Kalman filter models. Peaks in signal data were detected using a standard signal processing tool based on a wavelet transform, and low intensity peaks were excluded as noise. The performance of various clustering algorithms was compared for a subset of peaks. k-means clustering was used to successfully identify and track five vehicles, in the case where no vehicle passes another. The implementation of the proposed approach showed how several Kalman filter models can be managed simultaneously, by using cluster centers to correct the trackers' trajectories. This investigation has provided the groundwork for the creation of a tool that uses DAS signals to locate and track individual vehicles. The method offers flexibility in the specific implementation, and has lead to many ideas for further investigation. In particular, the comparison of clustering algorithms suggested that Mini Batch k-means clustering can be applied to reduce computational cost of the implementation provided for k-means. We hope that this method will be improved upon, and lead to a new application of DAS technology to real-time traffic monitoring.

Acknowledgements

The team would like to thank Fotech and our industry mentor Matt McDonald for introducing and supporting our industry project and for his mentorship and advice throughout this project. In particular, we thank his guidance through the intricacies of the DAS system and for teaching us approaches for detecting vehicle locations, and tracking vehicle trajectories, and also for providing us the data and primary codes. We would also like to extend our thanks to our academic mentor Prof. Parimala Thulasiraman (Department of Computer Science, University of Manitoba) for her mentorship and introducing to us approaches that related our industry problem in the data-mining literates. Furthermore, we would like to thank Ying Ying Liu (Department of Computer Science, University of Manitoba) for providing us with resources that helped us with the project. Lastly, we thank PIMS, and Mitacs and the organizing committee who put the PIMS $Math^{Industry}$ workshop together for making this investigation possible.

References

[1] N. Aggarwal and W. C. Karl. *Line detection in images through regularized hough transform.* IEEE Transactions on Image Processing, vol. 15, no. 3, pp. 582-591, 2006.

[2] F. Auger, M. Hilairet, J. M. Guerrero, E. Monmasson, T. Orlowska-Kowalska and S.Katsura. *Industrial Applications of the Kalman Filter: A Review.* IEEE Transactions on Industrial Electronics, vol. 60, no. 12, pp. 5458-5471, 2013.

[3] H. Cho and M. K. An. *Co-Clustering Algorithm: Batch, Mini-Batch, and Online.* International Journal of Information and Electronics Engineering, vol. 4, no. 5, 2014.

[4] J. Gregoire, D. Dale and R. Bruce van Dover. *A wavelet transform algorithm for peak detection and application to powder x-ray diffraction data.* The Review of scientific instruments, vol. 82, 015105, 2011.

[5] S. Guha, R. Rastogi, and K. Shim. *Cure: An efficient clustering algorithm for large databases.* Information Systems, vol.26, pp. 35-58, 2001.

[6] M. Inaba, N. Katoh, and H. Imai. *Applications of Weighted Voronoi Diagrams and Randomization to Variance-Based k-Clustering: (Extended Abstract)*, Proceedings of the tenth annual symposium on Computational geometry (SCG '94), Association for Computing Machinery, New York, NY, USA, pp. 332–339, 1994.

[7] Y. Jiang, Y. Liao, and G. Yu. *Affinity propagation clustering using path based similarity.* Algorithms, vol. 9, no. 3, p. 46, 2016.

[8] C. S. Lai, Y. Jia, *et al A Review of Technical Standards for Smart Cities.* Clean Technologies, vol. 2, no. 3, pp. 290–310, 2020.

[9] R. Lior, and O. Maimon. *Clustering methods. Data mining and knowledge discovery handbook.* Springer US, pp. 321-352, 2005.

[10] H. Liu, J. Ma, W. Yan, W. Liu, X. Zhang and C. Li. *Traffic Flow Detection Using Distributed Fiber Optic Acoustic Sensing.* IEEE Access, vol. 6, pp. 68968-68980, 2018.

[11] A. Masoudi and T. P. Newson. *Contributed review: Distributed optical fibre dynamic strain sensing.* Review of Scientific Instruments, vol. 87, 011501, 2016.

[12] F. Pedregosa, G. Varoquaux, *et al*, *Scikit-Learn: Machine Learning in Python.* Journal of Machine Learning Research, vol. 12, 2012.

[13] R. Refianti, A. B. Multiara, and S. Gunawan. *Time affinity complexity comparison between affinity propagation algorithms.* Journal of Theoretical and Applied Information Technology, vol. 95, no. 7, pp. 1497-1505, 2017.

[14] D. Sculley. *Web-Scale k-Means Clustering.* Proceedings of the 19th international conference on World wide web (WWW '10), Association for Computing Machinery, New York, NY, USA, pp. 1177–1178, 2010.

[15] L. Shiloh, A. Eyal and R. Giryes. *Efficient Processing of Distributed Acoustic Sensing Data Using a Deep Learning Approach.* Journal of Lightwave technology, vol. 37, no. 18, pp. 4755-4762, 2019.

[16] C. Wiesmeyr, M. Litzenberger, M. Waser, A. Papp, H. Garn, G. Neunteufel, and H. Döller. *Real-Time Train Tracking from Distributed Acoustic Sensing Data*, Applied Sciences, no. 2, p. 448, 2020.

[17] T. Zhang, R. Ramakrishnan, and M. Livny. *BIRCH: an efficient data clustering method for very large databases.* Proceedings of the 1996 ACM SIGMOD international conference on Management of data, Association for Computing Machinery, vol. 25, no. 2, pp. 103–114, 1996.

Brian Chan, Department of Mathematics, UBC
E-mail address: btchan@telus.net

Emily R. Korfanty, Department of Mathematics and Statistics, University of Victoria
E-mail address: emilykorfanty@uvic.ca

Sarah Nataj, Department of Mathematics, University of Manitoba
E-mail address: sarah.nataj@umanitoba.ca

Jaeun Park, Department of Mathematics, University of Wisconsin-Madison
E-mail address: park367@wisc.edu

Boya Peng, Department of Statistics, University of Alberta
E-mail address: pboya@ualberta.ca

Jianou Zhang, Department of Statistics, University of Saskatchewan
E-mail address: jiz127@usask.ca