Pacific Northwest
NATIONAL LABORATORY

PNNL-35888

Augmented Human Analysis (AHA)

April 2024

Jonathan H. Tu
Rogene M. Eichler West
Emily Ellwein
Jason M. Vann

U.S. DEPARTMENT OF ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from
the Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062
www.osti.gov
ph: (865) 576-8401
fox: (865) 576-5728
email: reports@osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
or (703) 605-6000
email: info@ntis.gov
Online ordering: http://www.ntis.gov

PNNL-35888

Augmented Human Analysis (AHA)

April 2024

Jonathan H. Tu
Rogene M. Eichler West
Emily Ellwein
Jason M. Vann

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99354

Abstract

Radio frequency (RF) signal monitoring generally emphasizes intentionally generated signals, such as WiFi, Bluetooth, or cellular transmissions. However, electronic devices also produce unintended radiated emissions (UREs), which could also be useful in RF spectrum analysis. In either case, deriving intelligence from RF signals is typically a human-intensive process requiring significant domain knowledge. In the Augmented Human Analysis (AHA) project, we investigate the utility of dimensionally aligned signal projection (DASP) and machine learning (ML) algorithms for accelerating RF analysis workflows. We find that while DASP algorithms can indeed highlight signal characteristics relevant for classification tasks, the choice of algorithmic hyperparameters greatly affects performance. To address this challenge, we evaluate the quality of DASP outputs using the silhouette score, which measures how well data points cluster; high silhouette scores indicate good clustering, and thus good hyperparameter values. This approach is critical for machine learning pipelines as the DASP parameters cannot be directly optimized during model training. By identifying good DASP parameters, and thus good DASP outputs, as a preprocessing step, we can decrease the amount of effort required for downstream ML model training. We demonstrate our workflow using a dataset of UREs from common household devices, showing that even without the aid of ML, proper selection of DASP parameters enables clustering by device type.

Acknowledgments

This research was supported by the **National Security Directorate Mission Seed** under the Laboratory Directed Research and Development (LDRD) Program at Pacific Northwest National Laboratory (PNNL). PNNL is a multi-program national laboratory operated for the U.S. Department of Energy (DOE) by Battelle Memorial Institute under Contract No. DE-AC05-76RL01830.

Acronyms and Abbreviations

AHA – Augmented Human Analysis
EM - Electromagnetic
RF – Radio frequency
ML – Machine learning
DASP - Dimensionally Aligned Signal Projections
UREs - Unintended Radiated Emissions
t-SNE – t-distributed stochastic neighbor embedding
DL – Deep learning
NN – Neural network
DNN – Deep neural network
DCC - Deep Continuous Clustering
CC - Contrastive Clustering (CC)
C3 - Cross-instance guided Contrastive Clustering
DCDC - Doubly Contrastive Deep Clustering
DeepDPM - Deep Clustering with An Unknown Number of Clusters
GCML - Generalized Clustering and Multi-Manifold Learning
ProPos - Learning Representation for Clustering via Prototype Scattering and Positive Sampling
TCL - Twin Contrastive Learning for Online Clustering
MNIST – Modified National Institute of Standards and Technology numerical image dataset
CIFAR-10 – Canadian Institute for Advanced Research 10-class image dataset
STFT - Short-Time Fourier Transform
FASP – Frequency-aligned Signal Projection
HASP – Harmonically-aligned Signal Projection
MASP – Modulation-aligned Signal Projection
CMASP – Cross-modulation-aligned Signal Projection
FHASP – Frequency-based Harmonically-aligned Signal Projection
SCAP – Spectral Correlation-aligned Signal Projection
ACC – Accuracy
NMI - Normalized Mutual Information
ARI - Adjusted Rand Index

Contents

Figures

Tables

1.0 Introduction

Effective monitoring of the electromagnetic (EM) spectrum for radio frequency for (RF) devices requires the ability to detect and distinguish between allowed, disallowed, and unknown devices, all of which may be transmitting intentionally or unintentionally. The range of frequencies to be monitored and the growing number of devices contributing to the RF spectrum make this task increasingly prohibitive for human analysts. Consequently, there is an urgent need to develop automated signal processing and machine learning (ML) technologies that can rapidly identify signals that require additional analyst attention. Doing so is an ongoing and active area of research due to numerous technical challenges, including spectral and temporal overlap of signals, variability in signal characteristics even within a single class of devices, and the impact of the background environment.

The standard approach to analyzing an RF signal is to study its spectral characteristics, such as its carrier frequency and/or bandwidth, among others. This is almost universally done using Fourier-based methods (e.g., spectrograms), or in some cases, wavelet-based equivalents (e.g., scaleograms). Recently, dimensionally aligned signal projections (DASP) algorithms have been proposed for this purpose [1]. DASP algorithms augment traditional Fourier analysis by highlighting signal characteristics that are inherent to the physical implementation of engineered RF devices, making it easier to identify features such as harmonics or modulations. However, the performance of DASP algorithms is highly dependent on hyperparameters like the suspected carrier frequency, modulation frequency, or bandwidth.

In this work, we develop a systematic framework for selecting DASP hyperparameters based on the silhouette score. (Alternatively, other clustering metrics like the Dunn index, Davies–Bouldin index, or C-index [2-4] might also be effective.) First, raw signals are collected from a number of known classes; here we use an experimental dataset comprising RF signals capturing unintended radiated emissions (UREs) from household devices. Each signal is then transformed into an image-like object using a DASP algorithm with a particular set of hyperparameters, each of which can be thought of as a point in a point cloud in the high-dimensional DASP output space. Then the silhouette score is computed to characterize the presence of clusters in that point cloud, with a more positive score indicating larger distances between clusters and smaller distances within clusters. As such, a more positive silhouette score also indicates that the proposed DASP algorithm and hyperparameters are effective in highlighting spectral features that help distinguish between signal classes. Using t-SNE (t-distributed stochastic neighbor embedding) as a visualization tool, we show that indeed, DASP outputs that achieve higher silhouette scores better separate different signal classes into distinct clusters.

While our silhouette score analysis is useful on its own for evaluating the utility of DASP algorithms for RF signal processing, we note that it may also be very useful as part of a deep learning (DL) pipeline. Specifically, we note that there are many recently proposed methods that utilize deep neural networks (DNNs) to transform data into a representation space where different classes separate into distinct clusters, including Deep Continuous Clustering (DCC) [5], Contrastive Clustering (CC) [6], Cross-instance guided Contrastive Clustering (C3) [7], Doubly Contrastive Deep Clustering (DCDC) [8], Deep Clustering With An Unknown Number of Clusters (DeepDPM) [9], Generalized Clustering and Multi-Manifold Learning (GCML) [10], Learning Representation for Clustering via Prototype Scattering and Positive Sampling (ProPos) [11], and Twin Contrastive Learning for Online Clustering (TCL) [12]. In our initial efforts to reproduce published results on benchmark datasets like MNIST and CIFAR-10, we found many of these methods to be sensitive to hyperparameters. Our later efforts to apply these methods to

RF data were similarly challenging. We propose the silhouette score as a diagnostic tool that can help determine if the model input data are reasonably clusterable. In other words, starting with data with a high silhouette score may allow researchers to focus on fine-tuning the neural network itself, rather than wondering if the input data contain enough information to achieve the desired clustering.

2.0 Methods

2.1 Corona Duff URE dataset

The Corona Duff dataset comprises URE measurements of 25 common household devices (see Table 1) collected in an office environment. Current and voltage data were collected using a Behringer UMC202HD recording device with a Pearson 411C probe at a sample rate of 192 kHz. Each collection event spanned ten minutes. For further details see the description of a similar data collection in [1]. To better enable machine learning, we implemented a custom PyTorch dataset class (see Appendix A) for loading and interacting with the Corona Duff data.

Table 1. Household devices included in Corona Duff dataset.

Alarm clock	HP monitor	Roku 2 XS	Vizio Blu-Ray
Angle grinder	HP printer	Rotary tool	Wii U
APC UPS	Kano computer	Space heater	Wired router
Clothing Iron	Lasko standing fan	Stand blender	Wireless router
CyberPower UPS	LED bulb	Table fan	
Dell monitor	Massage pad	Upright vacuum	
Gateway laptop	Power meter	USRP B210	

2.2 DASP algorithms

DASP algorithms are Fourier-based methods designed to emphasize signal characteristics commonly found when analyzing RF emissions from electronic circuits, including harmonics, modulations, and spectral correlations. They produce image-like outputs, similar to spectrograms. As part of this work, we implemented a number of DASP algorithms (see Table 2) in a new Python package (see Appendix A).

Table 2. DASP algorithms implemented for AHA.

Frequency-aligned signal projection (FASP)
Harmonically-aligned signal projection (HASP)
Modulation-aligned signal projection (MASP)
Cross modulation-aligned signal projection (CMASP)
Frequency-based harmonically-aligned signal projection (FHASP)
Spectral correlation-aligned signal projection (SCAP)

The basis of all DASP algorithms is the short-time Fourier transform (STFT). To fit the naming convention, we call this a frequency-aligned signal projection (FASP).

The harmonically-aligned signal projection (HASP) is designed to align a specified center frequency with its respective harmonics, with perfect harmonics appearing as a vertical line. For a fixed-type computation (HASP-F), harmonics of other frequencies will appear as diagonal lines. This can be adjusted using decimation or interpolation (HASP-D or HASP-I) so that all harmonic structures appear as vertical lines.

Modulation-aligned signal projection (MASP) applies an FFT to each frequency of an STFT. The resulting values capture how the energy content in a particular frequency band changes over time. The MASP output is similar to a spectrogram, but instead of plotting spectral content as a function of time and frequency, we plot as a function of carrier frequency and modulation frequency. Cross modulation-aligned signal projection (CMASP) is similar, but also aligns the harmonic content of the modulation sidebands of the carrier frequencies.

The frequency-based harmonically-aligned signal projection (FHASP) is a previously unpublished DASP algorithm that was further refined as part of AHA. It computes the instantaneous frequency of a signal using a Hilbert transform. Then HASP is applied to the result, enabling analysis of harmonics in the instantaneous frequency, rather than the raw signal. Similarly, spectral correlation-aligned signal projection (SCAP) applies HASP as a secondary computation to the autocorrelation of the Fourier transform of a signal. This can be used to align fixed frequency spacings between spectral peaks, which can result from frequency mixing, harmonics, or modulations.

2.3 DASP hyperparameter optimization

The performance of a DASP algorithm can be highly dependent on the specified hyperparameters. For instance, if the correct center frequency is chosen for a given signal, then HASP-F will produce a vertical line, but a slight error in the center frequency will instead yield a diagonal line, and a large error may produce no result at all. In this work, we use the silhouette score to systematically evaluate and select DASP hyperparameter values. The silhouette score incorporates inter- and intra-cluster distances to evaluate how well a dataset clusters. Values range from -1 to 1, where more positive values indicate better clustering.

For the Corona Duff dataset, we evaluate the FASP, HASP-D, MASP, and FHASP-D algorithms, sweeping through the hyperparameters listed in Table 3. To do so, we collect all signals in the dataset and divide them into fixed-length chunks. These signal chunks are then fed into the DASP algorithms to compute image-like outputs. All of the DASP outputs are flattened into one-dimensional vectors and collected into a single matrix, for which the silhouette score is computed. To visually confirm the results of the silhouette score analysis, we compare a t-SNE embedding of the raw signal chunks to one of the DASP outputs, with the expectation that DASP computations with a higher silhouette score will yield t-SNE embeddings with clearer clusters.

We note that an initial attempt at using a genetic algorithm to optimize the DASP hyperparameters failed to converge. Analysis of the fitness landscape determined that it was highly nonsmooth, which motivated the brute force hyperparameter sweep.

Table 3. Summary of DASP hyperparameter sweep

Algorithm	Hyperparameter values
FASP	channel: current, voltage chunk length [s]: 1, 2, 3, …, 60
HASP-D	channel: current chunk length [s]: 1 center frequency [Hz]: 20, 40, 60, 80, 100, 120, 500, 700, 800, 900, 1000, 1200, 2000, 4000, 8000, 10000 bandwidth [Hz]: 20, 30, 40, 60, 80, 90, 100, 120, 150, 160, 200, 240, 500, 700, 750, 800, 900, 1000, 1050, 1200, 1350, 1400, 1500, 1600, 1800, 2000, 2400, 3000, 4000, 6000, 8000, 10000, 12000, 15000, 16000, 20000 max harmonics: 28, 120, 540
MASP	channel: current chunk length [s]: 1, 2, 3, 4, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60 modulation frequency [Hz]: 50, 100, 150, …, 1000
FHASP-D	channel: current chunk length [s]: 1, 2.5, 5, 15, 30, 60 center frequency [Hz]: 10, 50, 100, 200, 500, 1000, 2000, 5000, 10000, 20000 bandwidth [Hz]: 10, 15, 50, 75, 100, 150, 200, 300, 500, 750, 1000, 1500, 2000, 3000, 5000, 7500, 10000, 15000, 20000, 30000 max harmonics: 28

2.4 Deep clustering algorithms

We consider eight different deep clustering algorithms for RF signal monitoring: Deep Continuous Clustering (DCC) [5]; Contrastive Clustering (CC) [6]; Cross-instance guided Contrastive Clustering (C3) [7]; Doubly Contrastive Deep Clustering (DCDC) [8]; Deep Clustering With An Unknown Number of Clusters (DeepDPM) [9]; Generalized Clustering and Multi-Manifold Learning (GCML) [10]; Learning Representation for Clustering via Prototype Scattering and Positive Sampling (ProPos) [11]; and, Twin Contrastive Learning for Online Clustering (TCL) [12]. Of these, we were able to successfully run DCC, CC, DCDC, and C3. For each, we evaluate the performance of the algorithm on standard reference datasets like MNIST and CIFAR-10, attempting to reproduce published results. Then we apply the algorithms to the Corona Duff dataset.

We note that the backbone for five of the algorithms is a ResNet-like model, and two rely on denoising autoencoders (DCC, GCM). DeepDPM describes its backbone as “clustering and subclustering networks." Furthermore, DeepDPM is the only algorithm we considered that does not require the user to first specify the anticipated number of clusters. All of the algorithms we considered had PyTorch implementations available on GitHub, which we used with minimal modifications, changing only hyperparameter values.

3.0 Results

3.1 DASP hyperparameter optimization

We consider four different DASP algorithms in our hyperparameter study: FASP, HASP-D, MASP, and FHASP-D. For each of these, we compute the silhouette score for a variety of hyperparameter values. These values are summarized above in Table 3. Figure 1 shows a histogram of the silhouette scores for each algorithm. We see that for all algorithms, there is a noticeable distribution of values, and close to half of all computations yield a negative silhouette score. Of the four algorithms, FASP performs worst, with nearly all negative silhouette scores. MASP and FHASP-D produce the most positive scores, with the highest single silhouette score of 0.365 achieved by a MASP computation with measured electrical current, 1 second signal chunks, and a modulation frequency of 50 Hz. The rarity of high silhouette scores shows how important it is to choose good hyperparameter values.

Though it is clear from Figure 1 that the vast majority of our Corona Duff DASP computations yield relatively low silhouette scores, the raw scores alone do not show how much of an improvement good hyperparameters provide over bad ones. To get a feel for this, we use t-SNE as a visualization tool, transforming a collection of high-dimensional DASP outputs into a two-dimensional representation that attempts to preserve the original high-dimensional distribution. Figure 2 shows four such visualizations, for the highest performing hyperparameter values for each DASP algorithm. In each t-SNE plot, every point corresponds to the DASP output for a single signal chunk, colored by the device type. For FASP, the best silhouette score is 0.002. There is very little discernable clustering in the t-SNE embedding, with many overlapping colors and multiple small clusters for a single device. This is slightly improved for HASP-D with a top silhouette score of 0.068, but there are still many regions of overlap. The t-SNE clustering is much improved for FHASP-D (top silhouette score of 0.204) and MASP (top silhouette score of 0.365); the clusters are well separated with little overlap, even though the silhouette scores are not near the theoretical maximum value of 1. These results show the utility of silhouette score analysis in evaluating the performance of DASP algorithms. The results with higher silhouette scores provide better initial class separation that likely require less effort to separate and classify using downstream methods like deep learning.

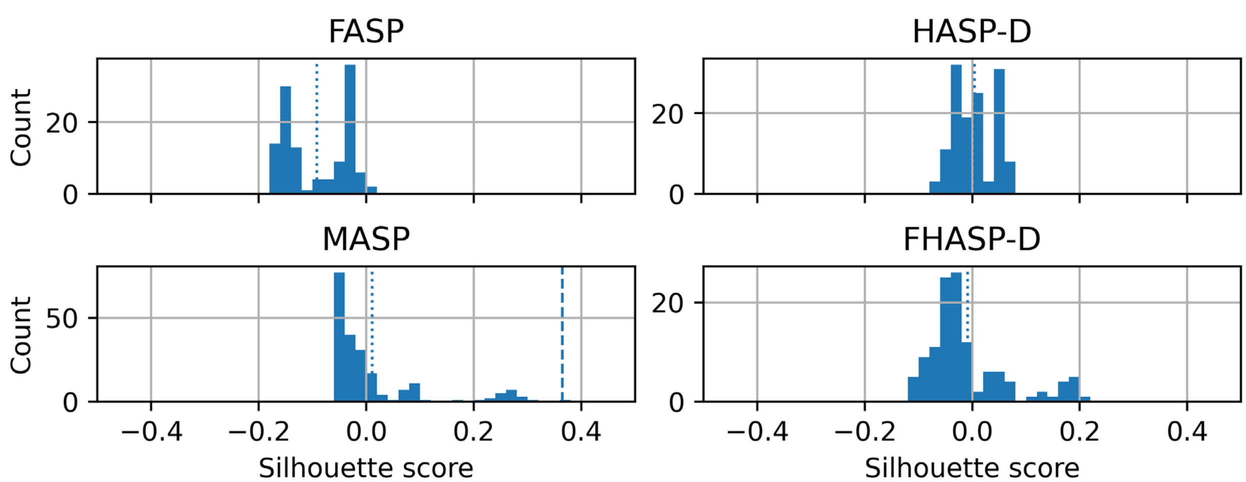

Figure 1. Histograms of silhouette scores for Corona Duff DASP computations for different hyperparameter values. The mean score for each algorithm is marked with a dotted line. The highest score across all algorithms is 0.365 and is marked with a dashed vertical line.

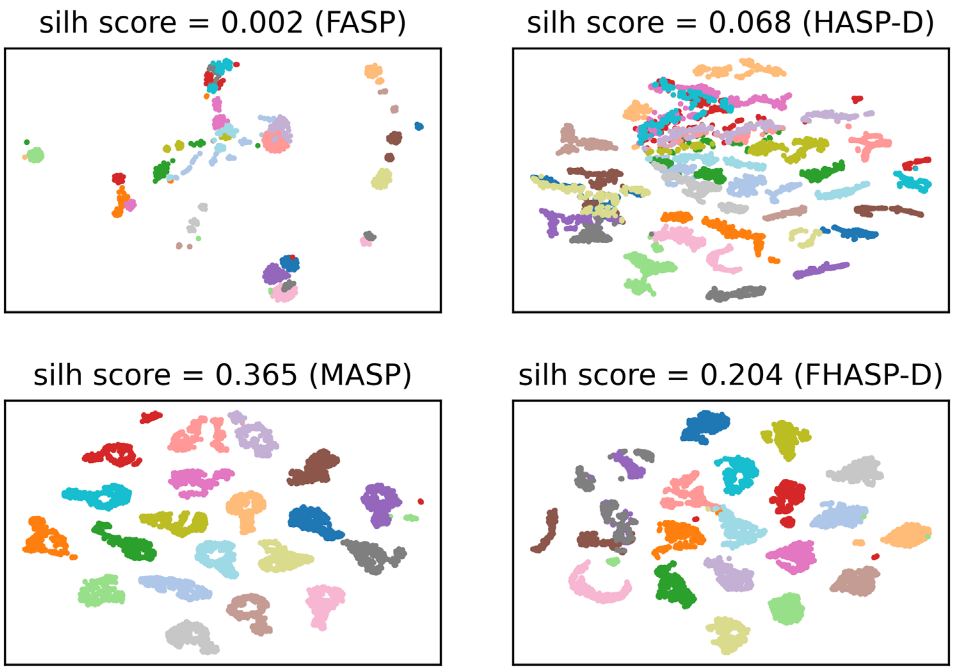

Figure 2. t-SNE embeddings of Corona Duff DASP outputs, computed for the best case silhouette scores. Each point corresponds to a single signal chunk and is colored based on the device type. Higher silhouette scores are predictive of more distinct t-SNE clusters.

3.2 Evaluation of deep clustering algorithms

Of the eight algorithms considered, we were only able to get four to run: DCC, CC, DCDC, and C3. For each of these, we attempted to reproduce published results testing the algorithms against common benchmark datasets; these results are summarized in Table 4. We note that even with published code repositories and manuscripts, it was non-trivial to reproduce these results. For instance, we had to adjust learning rates away from published values. For DCC, we underperformed by 0.173 on MNIST compared to the published value of 0.913; dataset analysis identified a few pixel-level differences between the authors' copy of MNIST and copies available in standard libraries like PyTorch. For DCDC, we underperformed by 0.116 compared to the published value of 0.699. For CC our observed accuracy was only 0.032 below the published value of 0.790. For C3 we outperformed the published value of 0.836 by 0.005. Table 4 also includes normalized mutual information (NMI) and adjusted rand index (ARI) values, which also show mixed results in our attempts to reproduce the benchmark computations.

Given the difficulty in analyzing simple benchmark datasets like MNIST and CIFAR10, it is perhaps not surprising that we had difficulty applying these deep clustering methods to the Corona Duff dataset. Our initial computations made minimal changes to the deep clustering architectures and hyperparameters. These did not yield any reasonable clustering accuracies, at which point we explored different learning rates (to debug the DNN) and different DASP hyperparameters (to debug in the input data). Through all of our exploration, we were unable to achieve reasonable clustering performance, and as such do not report any results. These difficulties are what motivated our silhouette score analysis, as it was unclear if our focus should have been to continue modifying the deep clustering model (or its training hyperparameters), or instead to improve the training data by tuning the DASP hyperparameters.

Table 4. Evaluation of deep clustering algorithms on benchmark datasets.

Algorithm	Dataset	Metric	Published	Ours	Difference
DCC	MNIST	ACC	0.913	0.740	-0.173
		NMI	0.915	0.851	-0.064
		ARI	--	0.737	--
CC	CIFAR10	ACC	0.790	0.758	-0.032
		NMI	0.705	0.671	-0.034
		ARI	0.637	0.596	-0.041
DCDC	CIFAR10	ACC	0.699	0.583	-0.116
		NMI	0.585	0.516	-0.069
		ARI	0.506	0.413	-0.093
C3	CIFAR10	ACC	0.836	0.841	+0.005
		NMI	0.743	0.750	+0.007
		ARI	0.703	0.709	+0.006

4.0 Conclusions

For the AHA project, we evaluated the utility of DASP and deep clustering algorithms for unsupervised monitoring of RF signals using the Corona Duff dataset of UREs. We found that the choice of DASP hyperparameters greatly impacted whether or not the DASP signal representations were consistently similar for like devices and distinct for unlike ones, significantly impacting clustering performance. We also observed that silhouette score analysis is highly predictive of clustering performance, as visualized by t-SNE embeddings, and this can help with the hyperparameter selection process. This insight is especially useful as we also found that state-of-the-art deep clustering methods are still lacking in maturity. We make this claim because it was non-trivial to reproduce published results for these algorithms on benchmark datasets like MNIST and CIFAR10. We further found that the advertised capabilities of deep clustering algorithms do not generalize, at least not without a lot of hyperparameter tuning, as clustering accuracy was very poor when the methods were adapted to DASP outputs. The use of silhouette scores to evaluate the utility of a dataset can provide researchers direction in whether next steps to improve performance might be directed toward tuning the DASP hyperparameters to produce better data representations, or whether the deep clustering pipeline (architectures, hyperparameters) requires refinement.

5.0 References

[1] J. M. Vann, T. P. Karnowski, R. Kerekes, C. D. Cooke, and A. L.Anderson, 'A dimensionally aligned signal projection for classification of unintended radiated emissions', IEEE transactions on electromagnetic compatibility, vol. 60, no. 1, pp. 122–131, 2017.[2] M.

[2] Gagolewski, M. Bartoszuk, and A. Cena, 'Are cluster validity measures (in) valid?', Information Sciences, vol. 581, pp. 620–636, 2021.

[3] S. Petrovic, 'A comparison between the silhouette index and the davies-bouldin index in labelling ids clusters', in Proceedings of the 11th Nordicworkshop of secure IT systems, 2006, vol. 2006, pp. 53–64.

[4] N. Bolshakova and F. Azuaje, 'Cluster validation techniques for genome expression data', Signal processing, vol. 83, no. 4, pp. 825–833, 2003.

[5] S. A. Shah and V. Koltun, 'Deep Continuous Clustering', arXiv:1803.01449, 2018.

[6] Y. Li, P. Hu, Z. Liu, D. Peng, J. T. Zhou, and X. Peng, 'Contrastive clustering', in Proceedings of the AAAI Conference on Artificial Intelligence,2021, vol. 35, pp. 8547–8555.

[7] M. Sadeghi, H. Hojjati, and N. Armanfard, 'C3: Cross-instance guidedContrastive Clustering', arXiv preprint arXiv:2211. 07136, 2022.

[8] Z. Dang, C. Deng, X. Yang, and H. Huang, 'Doubly Contrastive DeepClustering', arXiv [cs.CV]. 2021.

[9] M. Ronen, S. E. Finder, and O. Freifeld, 'DeepDPM: Deep Clustering With An Unknown Number of Clusters', in Conference on Computer Vision and Pattern Recognition, 2022.

[10] L. Wu, Z. Liu, J. Xia, Z. Zang, S. Li, and S. Z. Li, 'Generalized Clustering and Multi-Manifold Learning with Geometric Structure Preservation', in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 139–147.

[11] Z. Huang, J. Chen, J. Zhang, and H. Shan, 'Learning Representation for Clustering via Prototype Scattering and Positive Sampling', IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022.

[12] L. Yunfan, Y. Mouxing, P. Dezhong, L. Taihao, H. Jiantao, and P. Xi,'Twin Contrastive Learning for Online Clustering', International Journal of Computer Vision, 2022.

[13] J. M. Vann, T. Karnowski, and A. L. Anderson, 'Classification of unintended radiated emissions in a multi-device environment', IEEETransactions on Smart Grid, vol. 10, no. 5, pp. 5506–5513, 2018

Appendix A – Software products

As part of the AHA project, we developed a number of software products. One is a library to enable easier use of the Corona Duff dataset for machine learning. There are a number of methods to help parse the Corona Duff metadata, and most importantly, a custom PyTorch dataset for loading the data and feeding it to PyTorch machine learning models. The library has autodocumentation and is set up for easy installation using pip or conda. It is available at https://signals.pnl.gov/aha/coronaduff.

We also implemented the DASP algorithms from Table 2 as part of a new Python library. This library improves upon the original DASP code provided by Michael Vann, for instance replacing for loops with vectorized computations for speed, adding numerous error checks to guide the user toward selecting valid hyperparameters. Where possible, tests were done to ensure the new implementation generated outputs matching those of the old one. One key change is that the FHASP algorithm was altered to use the numerical derivative of the instantaneous phase, rather than just a difference. This is more consistent with the intent of the algorithm. The new dasp library is available at https://signals.pnl.gov/aha/dasp.

To enable code-free exploration of the Corona Duff dataset using DASP algorithms, we implemented a webapp called the Corona Duff Explorer. The webapp is implemented in Python using the Bokeh library. Using a variety of widgets (drop down menus, slider bars, text entry fields), webapp users can load Corona Duff signals and apply DASP algorithms. Figure 3 shows a screenshot of the webapp, which is available at https://signals.pnl.gov/aha/coronaduff-explorer.

Finally, we implemented ClusterSelect, a simple webapp that allow users to visually inspect and label two-dimensional point clouds. This webapp was also implemented using Python and Bokeh. It allows users to load x-y coordinates and then draw on the plot (e.g., using a computer mouse) to select points that belong to a cluster. The labels can then be exported for future use. This allows users to easily label data without need to write any code. The webapp is available at https://signals.pnl.gov/aha/clusterselect. See Figure 4 for a screenshot.

Figure 3. Screenshot of Corona Duff Explorer webapp. Users can apply DASP algorithms to Corona Duff signals without writing any code.

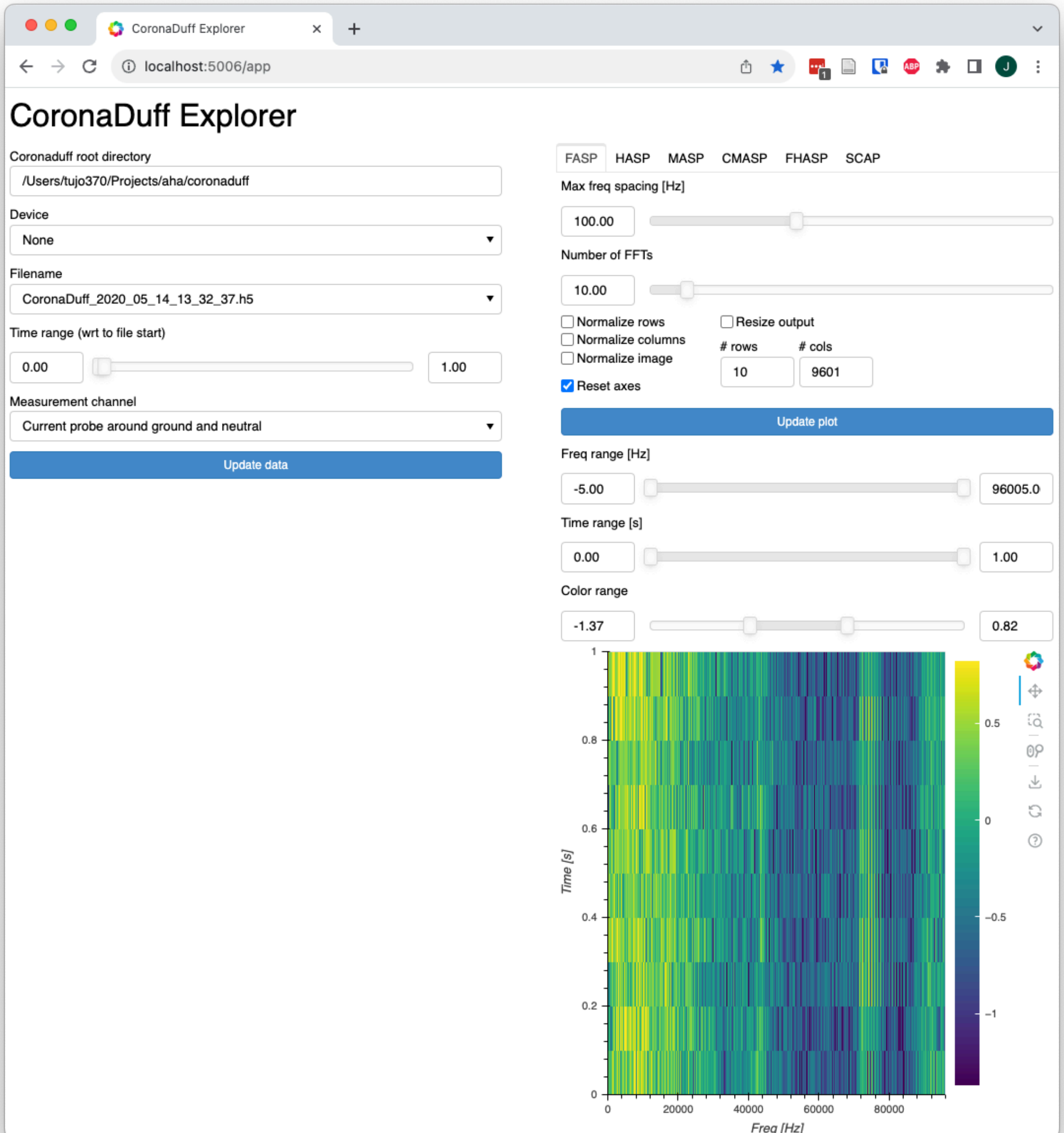

Figure 4. Screenshot of ClusterSelect webapp. Users can load two-dimensional point clouds and then interact with the webapp to select and label points.

Pacific Northwest National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354

1-888-375-PNNL (7665)

www.pnnl.gov