www.adeepakpublishing.com

Sigue, M. L. et al. (2022): JoSS, Vol. 11, No. 1, pp. 1125–1141
(Peer-reviewed article available at www.jossonline.com)

www. JoSSonline.com

On Designing Algorithms for Augmentation of On-Orbit CubeSats Sensor Data

Matthew L. Sigue
LCSEE, West Virginia University
Morgantown, WV, US

Thirimachos Bourlai
MILAB, E.C.E., University of Georgia
Athens, GA, US

Max Spolaor
NASA Independent Verification and Validation (IV&V),
Independent Test Capability (I.T.C.) Jon McBride Software Testing & Research (JSTAR) Laboratory
Fairmont, WV, US

Abstract

In this study, we propose an alternative methodology for verifying the relevancy of augmented data from different on-orbit sensors. This methodology is based on scalar and image-based augmentation, rule-based similarity metrics, and clustering algorithms. Scalar and image-based augmentation techniques are used to augment the original data multiple times during the augmentation process. With jittering, values changes (increases and decreases), permutations, rotations, and flipping, a set of augmented data is created for use with the rule-based similarity metrics. The Structural Similarity Index Measure (SSIM) and the Root Mean Squared Error (RMSE) are calculated for each data augmentation technique used and compared against a user-selected threshold value (bound). While the decision rule based on the bound declares the augmentations that pass the rule, the SSIM- and RMSE-computed errors are inputted into the spectral clustering algorithm. In this algorithm, the cluster that contains the original data would be compared as the baseline against the output of the rule-based system to ensure that the output from the rule-based system is relevant. Our rule-based method achieves an output similarity to the clustering algorithm of 93.46% in cases where the input data was forced to be 100% unique. To the best of our knowledge, this is the first time that rule-based similarity scores are supporting clustering algorithms to determine the relevancy of augmented data from an on-orbit satellite.

1. Introduction

Accurate and reliable data produced by on-orbit small satellites is a unique and expensive commodity, but it is crucial to expedite and streamline the CubeSat's development lifecycle. Numerical simulations have been created to aid with the research and development process of small satellites. However, for simulations to work, there needs to be sufficient raw (original data) collected and available to be processed, for the environment to be adequately simulated. Machine Learning-driven data augmentation techniques can provide a solution to overcome data shortages in such testing environments. These techniques involve a set

Corresponding Author: Matthew Sigue – ms0172@mix.wvu.edu
Publication History: Submitted – 11/13/20; Revision Accepted – 12/28/21; Published – 02/08/22

of processes that allow us to generate augmented datasets that can be hundreds of times larger in size than the original one (input) and preserve the authentic data properties and patterns that make the original on-orbit data so valuable for software assurance testing. The augmented data would aim to potentially cover edge cases, including data and their features not appearing in the original data, while still maintaining the realistic and relevant characteristics of the raw dataset. As such, the more data generated, the more potential situations (scenarios) can be analyzed, and thus, when an edge case occurs, more potential scenarios can be processed.

Data augmentation (DA) is a strategy that enables Machine Learning (ML) and Deep Learning (DL) practitioners to significantly increase the diversity of data that are available for processing, including training different models, without having to collect new data. DA techniques can be applied to many original data types, including data captured by 1D (sound waves; Cui, 2015) and 2D (images; Ho, 2019) sensors. In the latter case, while there are a certain number and types of data augmentation techniques, e.g., in images, it includes cropping, padding, and horizontal flipping, most of these techniques are considered basic types of augmentation. In the former case, i.e., on 1D raw sensor data (defined as numerical data that has not been preprocessed before), standard data augmentation techniques are flipping, rotations, jittering, permutations (Cagli, 2017; Davis, 2018; Kubota, 2016; Shorten, 2019; Um, 2017). These techniques can be helpful in CubeSat data, and are not limited to the dimensionality that the original input data, i.e., when working with 1D data, we can convert them into a 2D representation (image). The techniques for 2D dimensionality can be used. There are situations in which certain DA techniques cannot be used on specific data, for example, data that demonstrate a pattern repetition in 1D over time, in which cases 2D augmentations processes such as rotations cannot be used as data output will not contain that original temporal pattern anymore. This raises the question: how can we tell if the data we are given is relevant to the original information?

The relevancy of the output data can be determined using different data similarity metrics. The primary statistical measures for similarity are comparing the standard deviation, mean, median, and range of the original data to the augmented data. Another measure that has been used to compare the similarity of reference (of good or ideal quality) versus a degraded image is the Structural Similarity Index Measure (SSIM), which is considered an improved version of the universal image quality index proposed before (Rajkumar, 2016; Silva, 2007; Wang, 2005; Wang, 2004). The use of SSIM in this project is selected due to various studies that demonstrated that SSIM, when compared to mean squared error (MSE) and peak signal to noise ratio (PSNR), provides improved results in various studies (Horé, 2010; Silva, 2007; Martin and Bourlai, 2018; Abaza et al., 2012 and 2014; Bourlai et al., 2011; Narang and Bourlai, 2015). The root means squared error (RMSE) is another standard way to measure the error of a model in predicting quantitative data (Chai, 2014). In this work, each of the measures of similarity discussed above is used to compare different data augmentation approaches. The question is whether the augmented data maintain the realistic and relevant characteristics of the original raw dataset. Thus, we used a rule-based system and a set of clustering algorithms that are discussed below.

Clustering algorithms have been used to group data into clusters of similar data based on their features (Saha, 2019). Many algorithms have been developed over the years, including Hierarchical K-Means, K-Medoids, and many more (Hartigan, 1979; Jain, 2010; Kung, 2009; Celebi, 2013; Qi, 2017; Vo, 2013). Each algorithm uses different approaches to determine the cluster placement of each input data. A common way that clustering algorithms determine the cluster to be assigned is to determine the distance that point has to another central point (e.g., the center of different clusters), which is usually randomly assigned (Jain, 2010; Park, 2009; Qi, 2017). K-Means is often used to solve clustering problems using specific features, for example, using different similarity metrics (SSIM, RMSE, PNSR, and others) as the defining feature for a record (Choose Cluster, 2020; Ye, 2012; Huang, 1998; Wang, 2004). Clustering algorithms are often used to verify groups within a large dataset. For example, in the past, hierarchical clustering was used to determine the

grouping of plants within a water network (Boyacioglu, 2007).

Clustering has also been used to process voltage, current, and acceleration data, such as the one-dimensional data within the on-orbit SmallSat telemetry data (Madadi, 2019; Vo, 2013). In large-scale datasets, different clustering algorithms are used to determine the most efficient data management approach. For that reason, there is a comparison of six clustering algorithms for the dataset used in our study (Guyeux, 2019). The researchers used the idea to verify a rule-based system for similarity with a clustering algorithm in gene expression (Sethi, 2010). This gave us the idea to use the same approach to verify if the augmentations are relevant to the original data, i.e., by using our rule-based system and clustering algorithms. This algorithm was developed specifically for SmallSat data, as the clustering algorithms and augmentation techniques are chosen based on the specific nature of the sensor data and what would work best for the data. This method is innovative, as compared to other data generation methods as it attempts to use different dimensional techniques for the data; by comparing image structure of conventionally one-dimensional data, the process aims to generate relevant data from non-image image structure. Details of our proposed methodology are provided in the section below.

The rest of the paper is organized as follows: Section II presents the process for creating one-hundred-times augmented data using our rule-based system and clustering verification. The experimental procedure used to evaluate the proposed method and the databases used is described in Section III. Discussions on the experiments are included in Section III, and conclusions and future work are presented in Section IV.

2. Methodology

This section will describe the process (see Figure 1) for selecting the augmentations that would be considered relevant for 100 times the original data. Note that 100× augmentation was chosen as a reasonable target for this study. This target is set by the operator and can be easily increased or decreased.

The four salient steps of our methodological process are discussed below:

(i) Preprocessing: To get the data into a form where we could use it, first, the data is checked to see if the size of the daily dataset is large enough to do image-based augmentations. If it were not, the next day's dataset would be appended to it to increase the total size. Next, the data is padded to get the data to the next square value; this allows for the image-based augmentations to be done on the data. Next, the data would have extreme outliers removed by forcing values that are outside ±median2 to be replaced with ±median2 and stored to be inserted randomly after the augmentation process is complete. Finally, the data is converted into an image and then normalized to a greyscale image. The output from the preprocessing process is a greyscale image for each feature for a set of data.

(ii) Data Augmentation Process: The input for the augmentation process is preprocessed data created in the previous step. We randomly selected different augmentation techniques based on the total number of augmentation techniques that we decided were relevant to our processes to be used (see discussion in Section 1). The selected augmentation techniques are split into 1D (vector) and 2D (image-based) data augmentations. In the case where the data that is being augmented is defined as categorical, the data will be converted into a 2D matrix to be augmented using image-based augmentation techniques. In all other cases, the data go through a vector augmentation process, which includes jitter, and increase or decrease in all values based on a randomly generated number. A check is performed to ensure that the original maximum and minimum values selected are within an acceptable, predefined range. The image-based augmentations selected include the flipping of rows/columns, image rotations, and permutations. Each option has an equal chance to be applied to the dataset. The image rotations are only bounded to 1° in the positive and negative direction to maintain original data in the augmentation as the higher the rotation percentage within 45° would include more filler zeros values that will skew the data and the output of the similarity metrics. The output from the augmentation process is set to be 1000 augmentations and is ready to be compared against the rules in the next step.

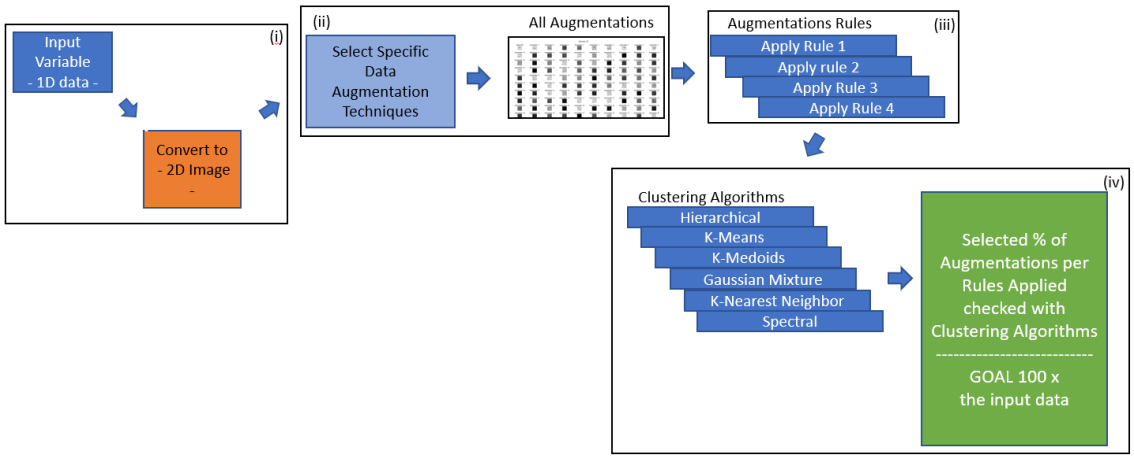

Figure 1: An overview of the data augmentation methodology we used for CubeSat Sensor Data Augmentation.

(iii) Data Augmentation Rules: We define four rules to determine the relevancy of each augmentation to the original/raw data. The input would be all the augmentations generated in the previous step. Each rule is used independently in testing the relevancy of each augmentation.

- Rule 1 involves the Structural Similarity Index (SSIM) metric that extracts three key features from an image: luminance, contrast, and Structure. The comparison between the two images (a reference and a target) is performed based on these key features. Figure 2 below shows the Structural Similarity Measurement system flow, where signals X and Y refer to the reference and sample/target images, respectively.

This system computes the SSIM between two images, a value between -1 and +1 (Wang, 2004), as discussed above. A value of +1 means that the compared images are very similar or the same, while a value of -1 indicates the images are very different. Please note that sometimes these values are normalized to be in the range [0, 1], where the extremes hold the same meaning.

- Rule 2 involves using basic statistical metrics, namely standard deviation, mean, median, and range, where each statistic is equally weighted.

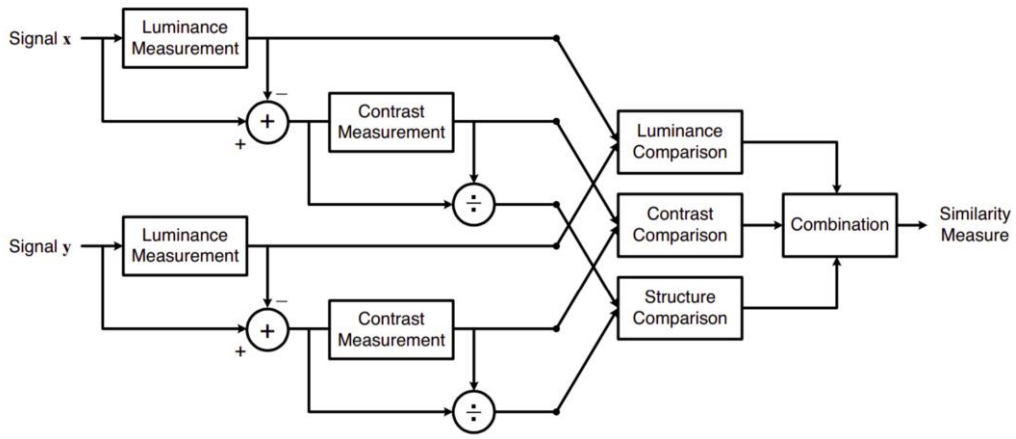

Figure 2: The Structural Similarity Measurement System. Source: Wang, 2004.

- Rule 3 is the combination of rules 1 and 2, with the output being required to pass both rule standards simultaneously.
- Rule 4 involves the RMSE metric, which is thought of as the normalized distance between the vector of predicted values and the vector of observed values. RMSE values can range between 0 to 1, with the smaller values indicating lower errors in the data. The output from each rule would be considered the portion passed for the rule and will be compared to the clusters in the next section.

(iv) Clustering Algorithms: The input for the clustering would be all the augmentations generated by the previous process. For the clustering algorithms, the complete set of augmentations is passed into six (empirically selected) different clustering algorithms, namely the Hierarchical, K-Means, K-Medoids, Gaussian Mixture, K-Nearest Neighbor, Spectral Clustering. These algorithms determine the data group, which corresponds to the same area as the augmentations generated by the rule-based system. In our proposed approach, the output from the rule-based system, along with the assessment from the clustering algorithms, should indicate relevant augmentations. The relevancy is determined by comparing the shared values between the rule and the cluster, where the cluster that would be compared is the cluster that contains the original non-augmented data. The final output generated by our methodological four-step process is selected to be the best 100 from the 1000 augmentations in terms of similarity to the original data.

3. Experiments

In this section, we will discuss the datasets that are used in the experiments. The datasets are original on-orbit data generated by the STF-1 CubeSat mission. We refer the reader to the article "NASA Operational Simulator for Small Satellites (NOS3): The STF-1 CubeSat case Study" by Geletko et al. (2019) for the details of the small satellite mission. The first dataset is a limited subset of the STF-1 CubeSat on-orbit telemetry data. In the file used, this dataset contained 132 sensors (voltage, current, temperature, gyroscope, and magnetic field for different positions are sensor clusters around the STF-1 satellite), which each included 6530 data points. This dataset is a single days' worth of data from the satellite, allowing the data to show an entire 24-hour time frame, the position, orientation, and internal sensor values that correspond to those specific locations and orientations. The second of the two datasets is the complete daily uploaded version of the original data, containing two years' worth of data. In the case of the second dataset, we focus on the same file containing 132 sensors but combined daily uploads to be in larger groups around 10,000 points. We combine data into more massive datasets due to the regular uploads sometimes not containing enough data to create high enough resolution images for the augmentation techniques and the rules we incorporated in the experiments.

What follows is a discussion of the data augmentation portion of the program. Data augmentation is used in two forms for the data that we imported from each of the datasets. Data augmentation is performed using 1D and 2D data, as discussed above. Each option had an equal chance to be chosen for the augmentation +.1,-.1,±: 1, none. For the matrix-based augmentation techniques, each technique had the same odds to be selected, flip rows, flip columns, positive or negative rotation within one degree, or permutation of columns.

This section describes how each rule was used for the program. Four rules were developed to determine the relevancy of each augmentation. Each rule was designed with different data features in mind. Rule 1 is based on SSIM, which is calculated on the original image versus the augmented image to find if the augmented data is within the 80% similarity of the original. All augmentations that had a value higher than .8 SSIM was passed through this rule. Rule 2 is based on the statistical metrics defined as the standard deviation, mean, median, and range of the original data compared to the augmented data. All four parameters must be within the scope of 20% around the original data. Rule 3 is the combination of Rule 1 and Rule 2 is done by passing the augmentations that pass Rule 1 at 80% similarity into Rule 2 using 20% as the range around the original data. To pass Rule 3, the augmentation must pass through both Rule 1 and Rule 2. Rule 4 is based on the RMSE of the augmentation.

RMSE is used as a measure of how close the data is from the line of best fit. Any augmentation that has an RMSE of less than .015 passes through this rule.

What follows is a set of figures demonstrating the outcomes of the experiments conducted, assuming that there could be some level of the same data. The trials were when we used the set of clustering algorithms, namely K-Means, K-Medoids, K-Nearest Neighbor, Hierarchical, Gaussian Mixture Method, and Spectral to determine which one verified that the rules were appropriately tuned. We generated the 100 times original data by selecting the cluster region that contains the SSIM values closest to 1, indicating 100 percent similarity.

We determined the number of clusters that we would use for each of the algorithms by using the silhouette values for K-Means, Hierarchical, and Gaussian Mixture Models were evaluated using the built-in MATLAB evalclusters command for k-clusters 1-9 (Evalclusters. 2020). Using this command, the Optimal K value for the three clustering algorithms was between 4-6 depending on the run of augmentations; therefore, the k-value of 5 was chosen for the number of clusters when a user input for the number of clusters was requested for any of the following algorithms (Evalclusters, 2020). For the number of neighbors for K-Nearest Neighbors, we used a value of 100 neighbors, which with the datasets we were given at 1000 augmentations, led to similar clusters as the 5-cluster input for the other algorithms.

Figure 3 shows that Hierarchical Clustering clusters ranging sizes but show a sizable region in the top left designated Cluster 4 that we consider the 'good' cluster. This clustering algorithm seems not to be the best choice for verifying our augmentations due to the clusters being completely different sizes. In almost all other clustering algorithms, the results were areas of about equal.

Figure 7 shows the K-Nearest Neighbor clustering algorithm; the results are slightly different from the previous figures due to the algorithm determining that there would be six clusters for the data compared to the five previously used.

This changes the clustering algorithm's region to consider 'good' tighter towards the SSIM and RMSE values that indicate identical (1 and 0, respectively). Figure 8 shows how the results of the Spectral Clustering algorithm show similar results for clusters as the other algorithms but find the area within Cluster 3 to be its group, which indicates a group of outliers that deviate from the rest of the data.

Each clustering algorithm behaves differently but returns roughly the same output, with certain regions being similar between the algorithms. Outliers can be easily determined with the Spectral clustering algorithm as the green region of Cluster 3 in Figure 8. Compared to the clusters in Figure 3, the clusters do not determine what we would consider outliers. Still, they contain smaller clusters, which may need to be added into the future's “good” augmentations. Figures 9-14 are like Figures 3-8, but each of the outputs includes data augmentations that are 100% unique. Unique data is included to show new relevant augmentations that do not contain duplicate information.

In Tables 1 and 2, a comparison of each of the clustering algorithms for a set of 1000 augmentations were made comparing the number of augmentations accepted by the rule versus each clustering algorithm. Table 1 includes the number of augmentations passed through the rule and the number of augmentations passed by the clustering algorithm.

As well as the number of augmentations that were shared by the two, and a percentage of the total number passed from the rule of all augmentations that were passed from the rule and cluster combined. Table 2 contains the same data but contains the data from 100% unique data augmentations.

4. Discussion: Data Augmentation Process and Target

In order to create 100 times the data, the augmented data's relevancy must be checked. The rule-based system for monitoring the relevancy of the augmentations by assessing different features of the augmentations. The features being SSIM, statistical measures (STD, mean, median, range), and RMSE, allowed for different types of information to be assessed to determine the relevancy of the augmentation. In our testing, we found that the

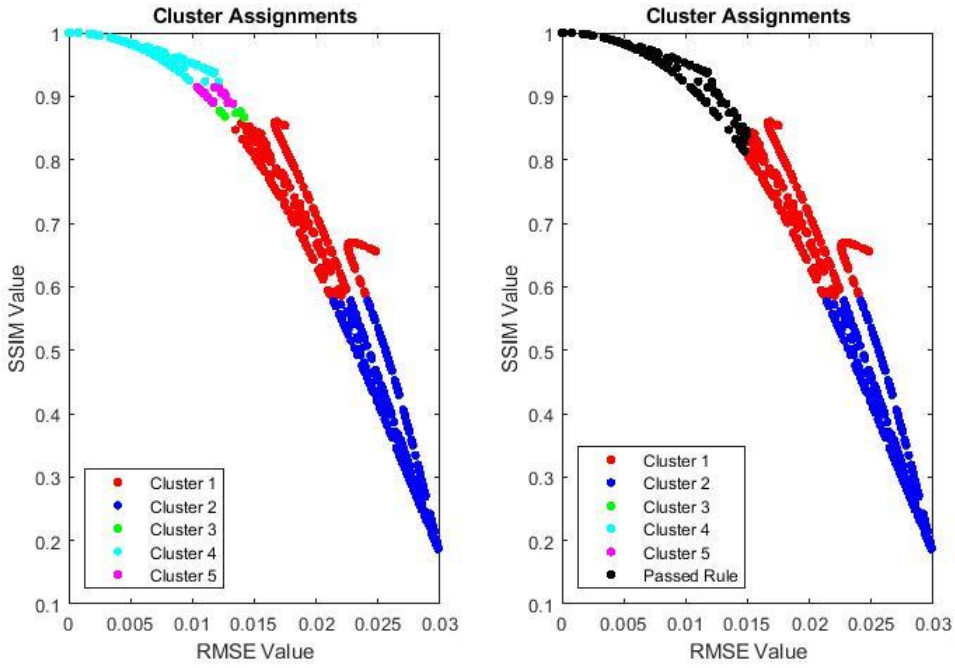

Figure 3. Comparison of Hierarchical Clustering Results (left) and the output of rule layered over clustering data (right).

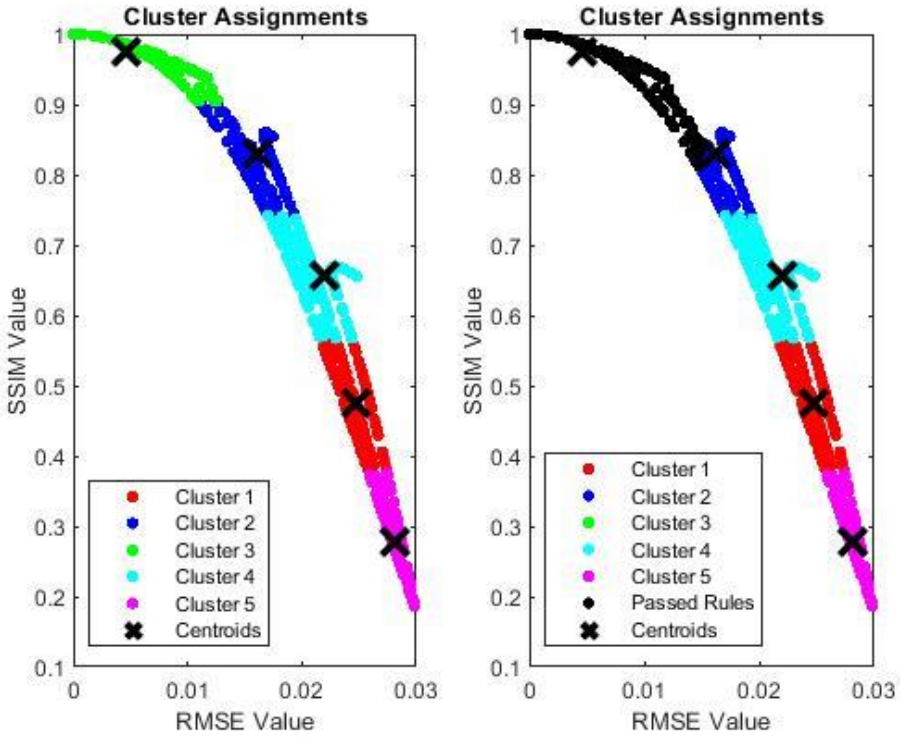

Figure 4. K-Means clustering (left) with rule passing augmentations layered over clustering data (right).

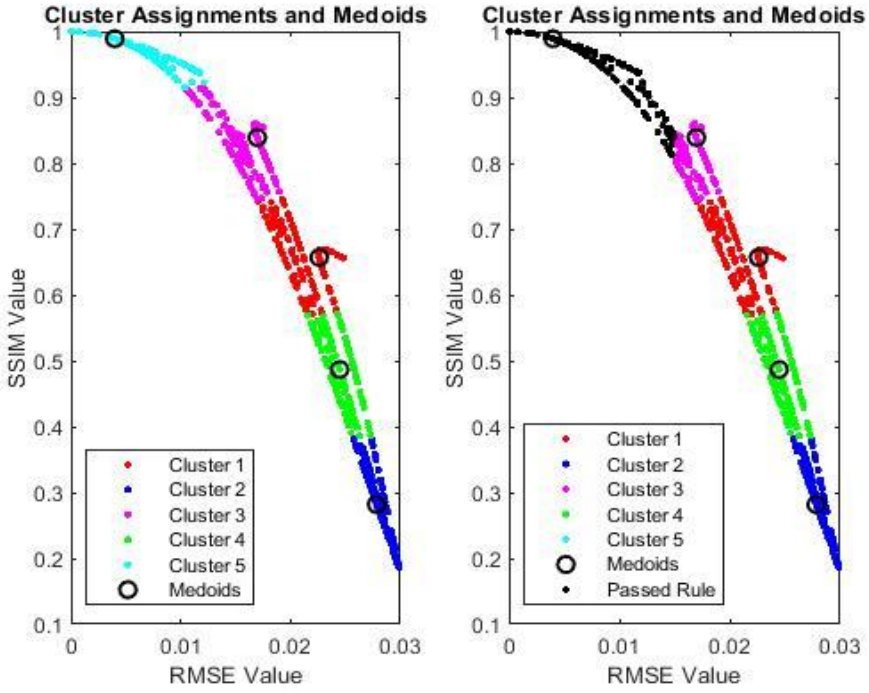

Figure 5. K-Medoids clustering (left) with rule passing Black dots layered over clustering data (right)

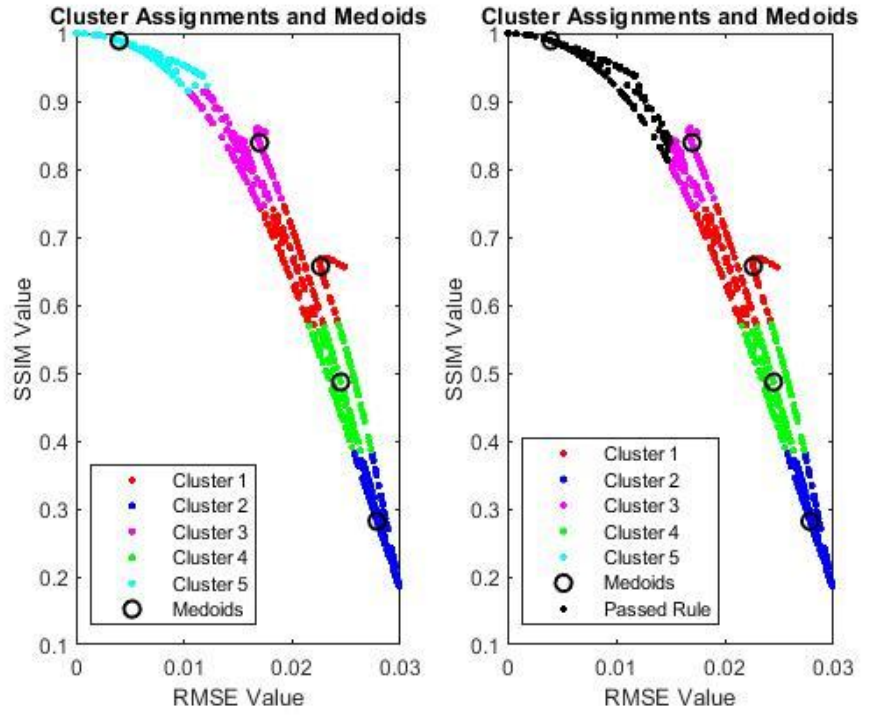

Figure 6. Gaussian Mixture Model Clustering with Passed Rule (black dots) layered over clusters in a series of different settings for Sigma and Shared Covariance.

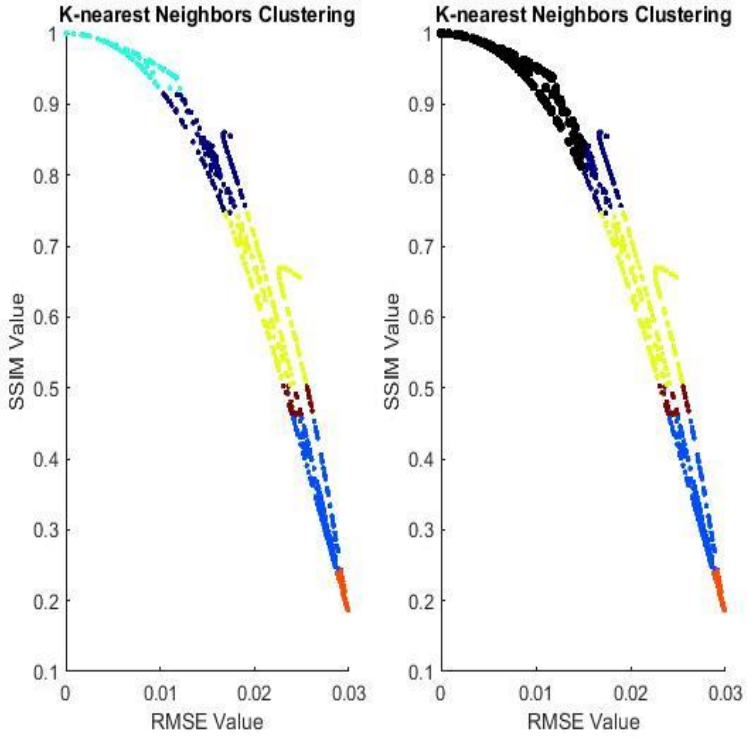

Figure 7. K-Nearest Neighbor Clustering data (left) with rule passing data layered over clustering data (right).

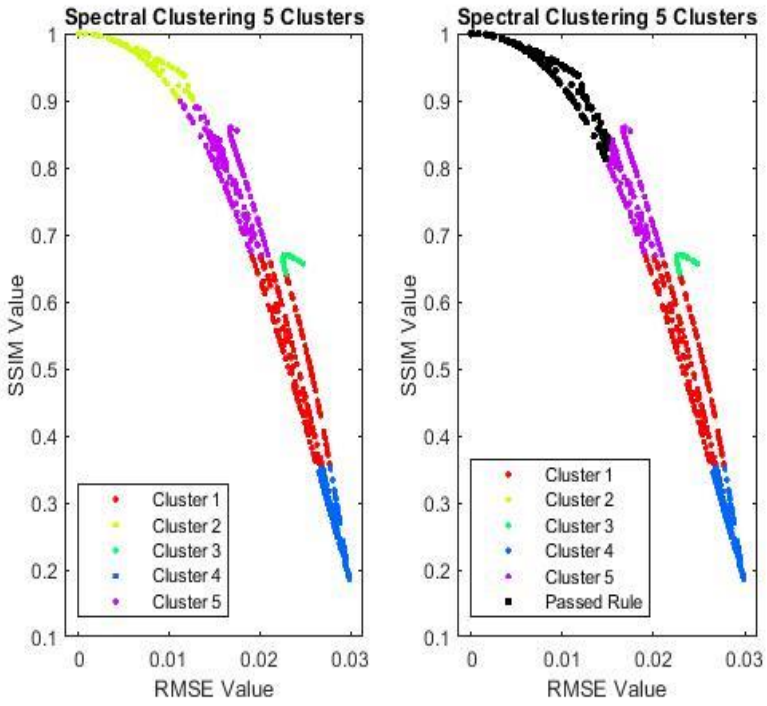

Figure 8. Spectral Clustering data (left) with rule passing data layered over clustering data (right).

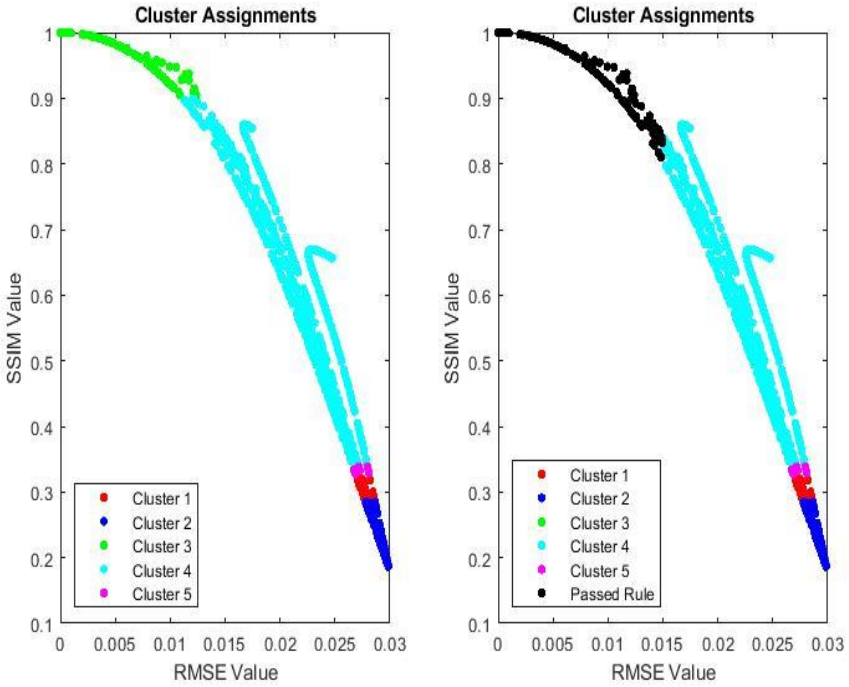

Figure 9. Hierarchical Clustering data (left) with rule passing data over clustering data (right) with unique augmentation data.

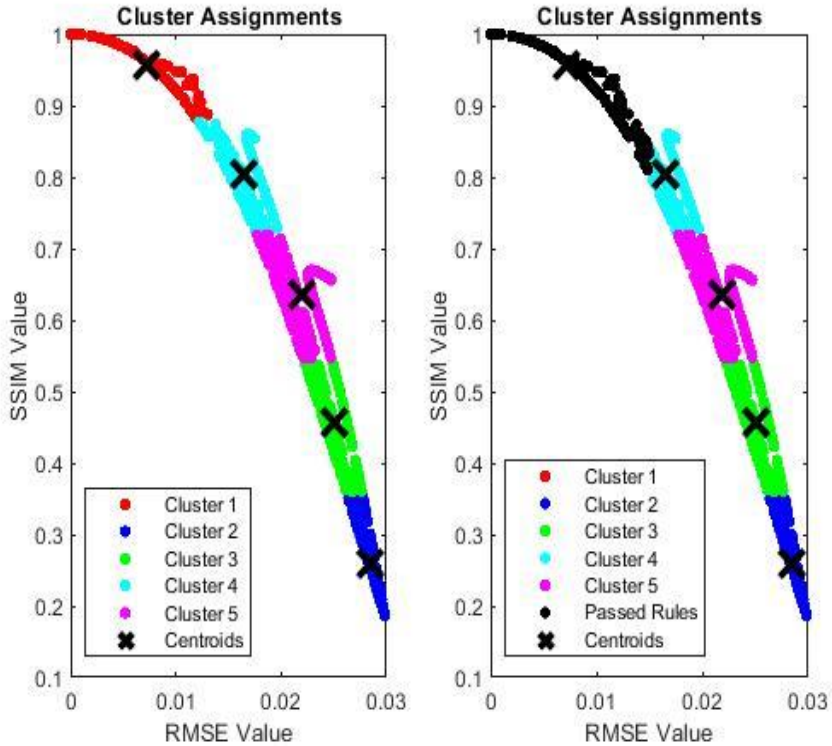

Figure 10. K-Means Clustering data (left) with rule passing data layered over clustering data (right) with unique data augmenta-

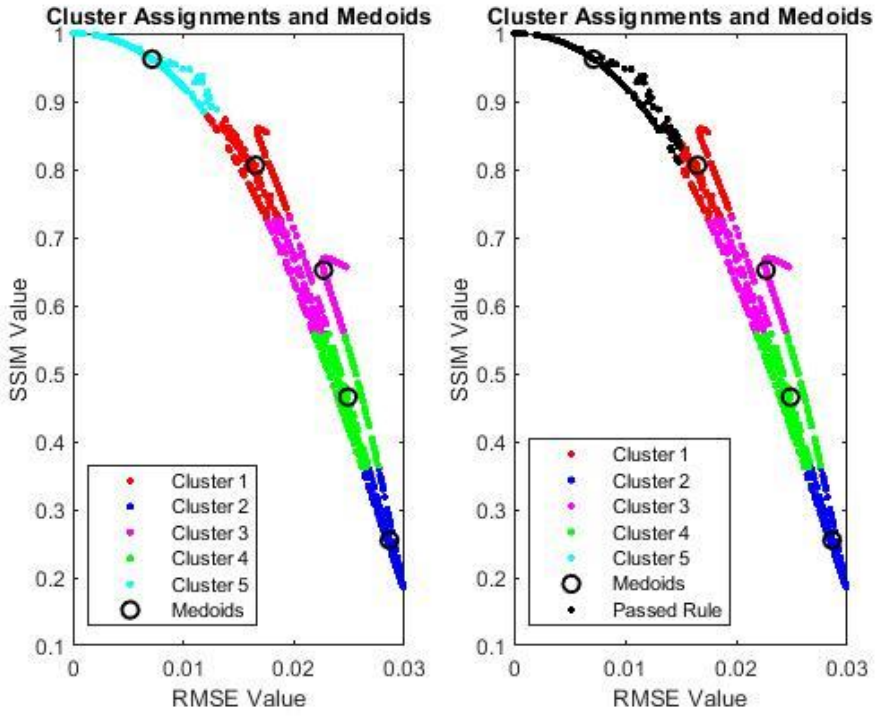

Figure 11. K-Medoids Clustering data (left) with rule passing data layered over clustering data (right) with unique data augmentations.

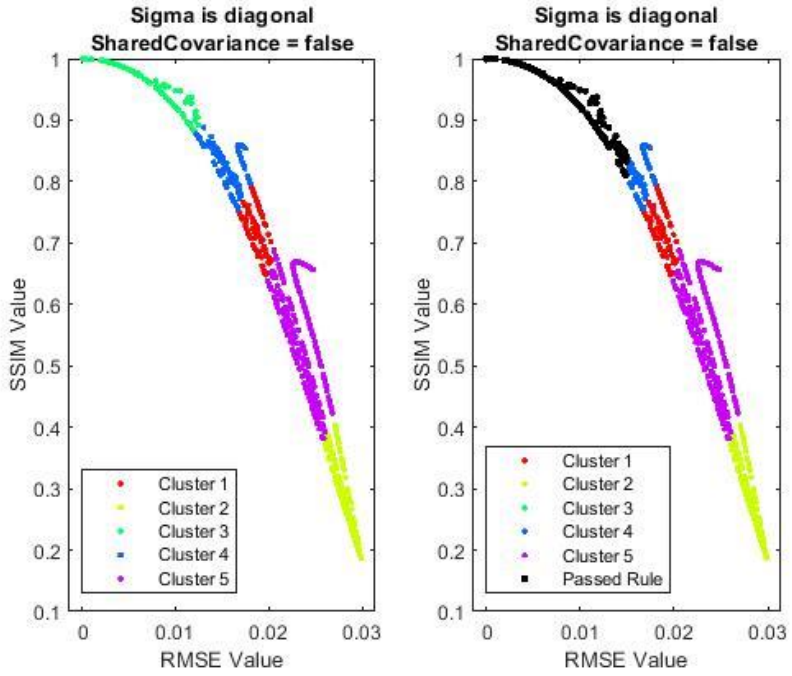

Figure 12. Gaussian Mixture Model clustering data (left) with rule passing clustering data (left) layered over clustering data.

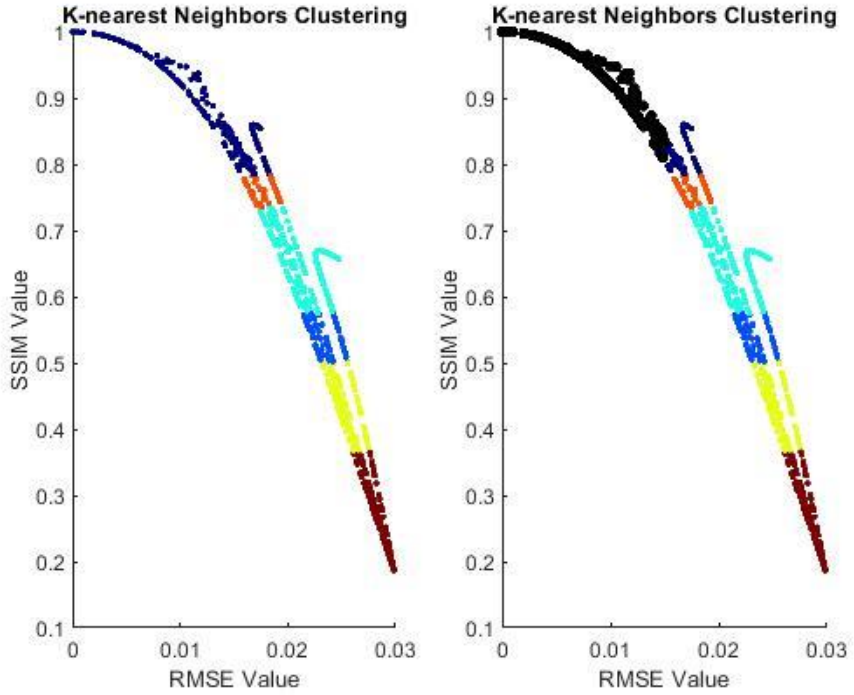

Figure 13. K-Nearest Neighbors clustering data (left) with rule-based data layered over clustering data (right) with unique data augmentations.

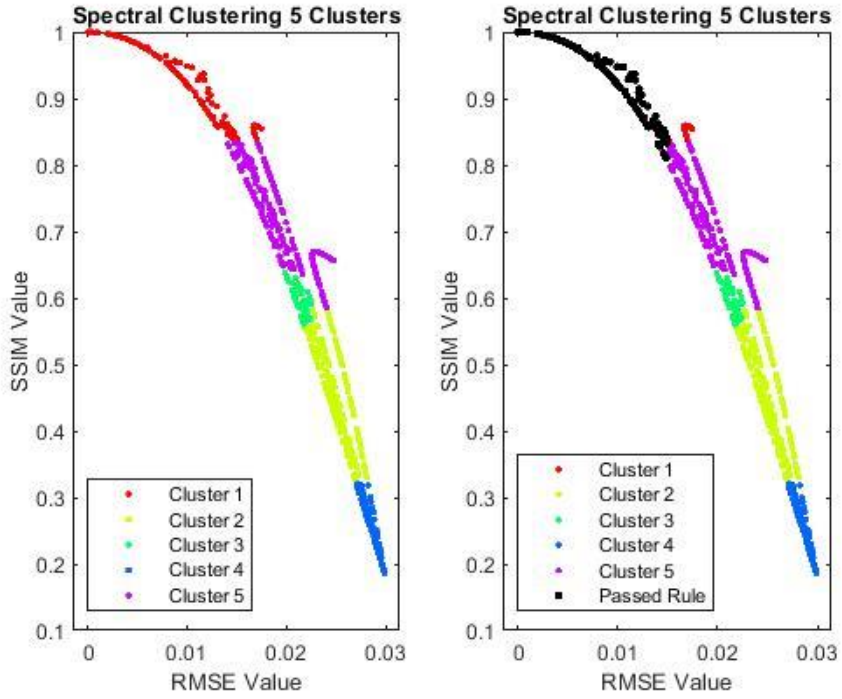

Figure 14. Spectral Clustering data (left) with rule-based data layered over clustering data (right) with unique data augmentations.

Table 1: Overview of the Rule-Based System Compared to the Clustering Assessment

Clustering Algorithm	# Augmentations Passed Rule	# Augmentations Passed Clustering	# Augmentations Shared Rule and Clustering	Shared augmentations/total unique augmentations (%)
Hierarchical	215	157	157	73.02
K-Means	215	167	167	77.67
K-Medoids	215	158	158	73.49
Gaussian Mixture Model	215	174	174	80.93
K-Nearest Neighbor	215	157	157	73.02
Spectral	215	169	169	78.60

Table 2: Overview of the Rule-Based System Compared to the Clustering Assessment, Given 100% Unique Data

Clustering Algorithm	# Augmentations Passed Rule	# Augmentations Passed Clustering	# Augmentations Shared Rule and Clustering	Shared augmentations/total unique augmentations (%)
Hierarchical	153	97	97	63.40
K-Means	153	107	107	69.93
K-Medoids	153	106	106	69.28
Gaussian Mixture Model	153	109	109	71.24
K-Nearest Neighbor (KNN)	153	257	153	59.53*
Spectral	153	143	143	93.46

* KNN's cluster contained more augmentations than the pass-rule. All other clustering algorithms included more augmentations in the cluster than the rule.

Table 3: Runtime of Each Clustering Algorithm, Excluding any Plotting (Average over 100 Runs)

	Hierarchical	K-Means	K-Medoids	Gaussian Mixture Model	K-Nearest Neighbor	Spectral Clustering
Runtime(sec)	0.117	0.021	0.356	0.293	0.677	0.305

\

features SSIM and RMSE were perfect for determining the relevancy of the data. As they gave a specific value on a known scale, so the interpretation of the information was straightforward. The use of statistical measures was the opposite, as the outputs were not promising as the outputted augmentations were at best identical copies of the original, or there were none. Statistical measures would be better suited for internal checks to ensure the data is close to the original information but should not be used as a rule, as it did not provide helpful output.

For each of the clustering algorithms, the shared augmentations over total unique augmentations were calculated. This measure was to determine how accurate the clustering algorithms and rules contained the same data. In the case where duplicate values were allowed, percentages hovered around the 75% range with notable Gaussian mixture model clustering containing 80% similarity. Where no duplicates were approved, the measure was closer to the 60-70% mark except for Spectral clustering, which maintained a 93% shared augmentations per total augmentations.

Based on Figures 3-8 and Table 1, the clustering algorithms that we decided are the best suited for our use case are K-Means and K-Nearest Neighbors. K-Means was selected due to its ease of use and understanding. The output from K-Means gives us the cluster that we can use while maintaining a useful similarity between the groups so each cluster can be easily understood. K-Nearest Neighbors was selected as a second choice because it did not require user input for the number of clusters but the number of nearest neighbors, leading to a cluster that was different from the K-Means that were still interpretable.

With K-Nearest Neighbors, since there were six clusters, it leads to a tighter cluster in the first region. Based on Table 3, K-mean maintained across one hundred runs the lowest runtime at .021 seconds, which was factored into our decision as the amount of time used in larger datasets scales with the data. As

such, we factored the value of having a lower time to run in our choice. K-Nearest Neighbors.

Based on Figures 9-14 and Table 2, Spectral clustering was the clustering algorithm chosen to assess each rule's output. This is due to when the data is 100% unique; the Spectral clustering algorithm returned the highest shared augmentation per total unique augmentations. At 93.46%, the region contained by the rules and the clustering algorithm were incredibly similar. The algorithm's runtime is not the fastest, but in terms of performance as an assessment tool, it was the first choice for our process.

Our second choice is K-Means, like in the previous experiments. It was selected for being the easiest to use to get the same level of results. In terms of other effects, K-Means was middle of the pack in terms of shared augmentations per total unique augmentations. This factored with the ease of use, and having the fastest runtime, the difference in shared augmentations per total unique augmentations, we decided it was an excellent second choice. There are a few limitations of the approach; five clusters for all clustering algorithms were used based on a range of 4-6 being the optimal cluster value based on K-Means, Hierarchical, and Gaussian Mixture Model algorithms. With how different each run of augmentations the optimal K value changes a lot and is not the best for all other clustering algorithms; more testing needs to be done to find the optimal clustering algorithm for each of the different algorithms. Another limitation is that the number of inputs augmentations for the clusters affects the cluster positions moving the “good” cluster by up to .1 SSIM in cases where there is not enough data.

5. Conclusions

We studied the problem of creating accurate, augmented small satellite operational data from original on-orbit datasets. The method we developed was designed to work with the sensor data from an on-orbit satellite and return data of similar values. We were tasked with two scenarios: the data could contain duplicate values and another where the data was one hundred percent unique. Our results differ based on each scenario; specifically, our method's similarity was on average 76.12% given data with duplicate values and an average of 71.08% similarity with one hundred percent unique data. Our recommendations given the choice of a specific algorithm for both scenarios are based on each algorithm's runtime and cluster similarity.

Achievement of our work was determining what we believe is the best choice for the rule in the rule-based portion of our method, as well as the best clustering algorithm. We propose using rules based in SSIM or RMSE as they produced predictable values that followed trends that allowed for easy assessment. In the first scenario, we determined that the K-mean clustering algorithm was the best option for creating 100 times the original data.

The average runtime of 0.021 seconds was considerably faster than the other options, with the similarity performance being on average 77.67%. In the second scenario, we recommend using Spectral clustering as its similarity performance was considerably higher than the other algorithms at 93.46%. The runtime of Spectral clustering is slower than K-Means at 0.305 seconds compared to K-Means at 0.021 seconds. Still, the average similarity score being outperformed by over 20% leading to Spectral clustering being the choice for the second scenario.

References

Abaza, A., Harrison, M. A., and Bourlai, T. (2012): Quality Metrics for Practical Face Recognition, presented at IAPR Int. Conf. on Pattern Recognition (ICPR), Tsukuba Science City, Japan. Available at: https://ieeexplore.ieee.org/document/6460821 (accessed Oct. 17, 2021).

Abaza, A., Harrison, M. A., Bourlai, T., and Ross, A. (2014): Design and Evaluation of Photometric Image Quality Measures for Effective Face Recognition, I.E.T. Biometrics. doi: 10.1049/iet-bmt.2014.0022. Available at: https://www.cse.msu.edu/~rossarun/pubs/AbazaFaceQuality_IET2014.pdf (accessed Oct. 17, 2021).

Bourlai, T., Ross, A., and Jain, A. (2011): Restoring Degraded Face Images for Matching Faxed or

Scanned Photos. IEEE Transactions on Information Forensics and Security, Vol. 6, No. 2, pp. 371–384. doi: 10.1109/TIFS.2011.2109951. Available at: https://ieeexplore.ieee.org/document/5706362 (accessed Oct. 17, 2021).

Boyacioglu, Hayal and Boyacioglu, Hülya. (2007): Surface Water Quality Assessment by Environmetric Methods. Env. Monitoring and Assessment., pp. 371–376. doi: 10.1007/s10661-006-9482-4. Available at: https://www.researchgate.net/publication/6690558_Surface_Water_Quality_Assessment_by_Environmetric_Methods (accessed Oct. 17, 2021).

Cagli, E., Dumas C., and Prouff E. (2017): Convolutional Neural Networks with Data Augmentation Against Jitter-Based Countermeasures. Lecture Notes in Computer Science Cryptographic Hardware and Embedded Systems, pp. 45–68. doi: 10.1007/978-3-319-66787-4_3. Available at: https://link.springer.com/chapter/10.1007/978-3-319-66787-4_3 (accessed Oct. 17, 2021).

Celebi M. E., Kingravi H. A., and Vela P. A. (2013): A Comparative Study of Efficient Initialization Methods for the K-Means Clustering Algorithm. Expert Systems with Applications, Vol. 40(1), pp. 200–210. doi: 10.1016/j.eswa.2012.07.021. Available at: https://arxiv.org/abs/1209.1960 (accessed Oct. 17, 2021).

Chai, T., and Draxler R. R. (2014): Root Mean Square Error (RMSE) or Mean Absolute Error (M.A.E.)? – Arguments against Avoiding RMSE in the Literature. Geoscientific Model Development, Vol. 7, No. 3, pp. 1247–1250. doi: 10.5194/gmd-7-1247-2014. Available at: https://gmd.copernicus.org/articles/7/1247/2014/ (accessed Oct. 17, 2021).

Choose Cluster Analysis Method. (2020): Available at: https://www.mathworks.com/help/stats/choose-cluster-analysis-method.html.

Cui X., Goel V., and Kingsbury B. (2015): Data Augmentation for Deep Neural Network Acoustic Modeling. IEEE/ACM Trans. Audio, Speech and Lang., Vol. 23, pp. 1469–1477. doi: 10.1109/TASLP.2015.2438544. Available at: https://ieeexplore.ieee.org/abstract/document/7113823 (accessed Oct. 17, 2021).

Davis, N. and Suresh, K. (2018): Environmental Sound Classification Using Deep Convolutional Neural Networks and Data Augmentation. 2018 IEEE Recent Advances in Intelligent Computational Systems (RAICS). doi: 10.1109/RAICS.2018.8635051. Available at: https://ieeexplore.ieee.org/document/8635051 (accessed Oct. 17, 2021).

Evalclusters (2020): Available at: https://www.mathworks.com/help/stats/evalclusters.html (accessed Oct. 17, 2021).

Geletko, D. M. et al. (2018): NASA Operational Simulator for Small Satellites (NOS3): The STF-1 CubeSat Case Study. JoSS, Vol. 7, No. 3, pp. 789-800. Available at: https://jossonline.com/letters/nasa-operational-simulator-for-small-satellites-nos3-the-stf-1-cubesat-case-study/ (accessed Oct. 17, 2021).

Guyeux, C. et al. (2019): Introducing and Comparing Recent Clustering Methods for Massive Data Management on the Internet of Things. J. of Sensor and Actuator Networks. doi: 10.3390/jsan8040056. Available at: https://www.mdpi.com/2224-2708/8/4/56 (accessed Oct. 17, 2021).

Hartigan, J. A., and Wong, M. A. (1979): Algorithm AS 136: A K-Means Clustering Algorithm. J. of the Royal Statistical Society, pp. 100-108. doi: 10.2307/2346830. Available at: https://www.jstor.org/stable/2346830 (accessed Oct. 17, 2021).

Ho, D., Liang, E., Stoica, I., Abbeel, P., and Chen, X. (2019): Population Based Augmentation: Efficient Learning of Augmentation Policy Schedules. Available at: https://arxiv.org/abs/1905.05393 (accessed Oct. 17, 2021).

Horé, A. and Ziou, D. (2010): Image Quality Metrics: PSNR vs. SSIM. 20th Int. Conf. on Pattern Recognition, pp. 2366-2369. doi: 10.1109/ICPR.2010.579. Available at: https://ieeexplore.ieee.org/document/5596999 (accessed Oct. 17, 2021).

Huang, Z. (1998): Extensions to the K-Means Algorithm for Clustering Large Data Sets with Categorical Values. Data Mining and Knowledge Discovery, Vol. 2(3), pp. 283–304. doi: 10.1023/A:1009769707641. Available at: https://link.springer.com/article/10.1023/A:1009769707641 (accessed Oct. 17, 2021).

Jain, A. (2010): Data Clustering: 50 Years Beyond K-Means. Pattern Recognition Letters, Vol. 31(8), pp. 651–666. doi: 10.1016/j.patrec.2009.09.011. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0167865509002323 (accessed Oct. 17, 2021).

Kubota K., Chen, J., and Little, M. (2016): Machine Learning for Large-Scale Wearable Sensor Data in Parkinson Disease: Concepts, Promises, Pitfalls, and Futures. Movement Disorders, Vol. 31(9), pp.1314-1326. doi: 10.1002/mds.26693. Available at: https://www.researchgate.net/publication/306004209_Machine_learning_for_large-scale_wearable_sensor_data_in_Parkinson%27s_disease_Concepts_promises_pitfalls_and_futures (accessed Oct. 17, 2021).

Kung, H. and Vlah, D. (2009): A Spectral Clustering Approach to Validating Sensors via Their Peers in Distributed Sensor Networks, in Proc. of the 18th Int. Conf. on Computer Communications and Networks, San Francisco, CA, Aug. 3–6.

Madadi, S., Mohammadi-Ivatloo, B., and Tohidi S., (2019): A Data Clustering Based Probabilistic Power Flow Method for AC/VSC-MTDC. IEEE Systems Journal, Vol. 13, No. 4, pp. 4324-4334. doi: 10.1109/JSYST.2019.2918234. Available at: https://ieeexplore.ieee.org/document/8734148 (accessed Oct. 17, 2021).

Martin, M. and Bourlai, T. (2018): Enhanced Tattoo Image Quality Assessment Through Multi-Spectral Sensing. IEEE Sensors J. doi: 10.1109/LSENS.2017.2768326. Available at: https://ieeexplore.ieee.org/document/8094320 (accessed Oct. 17, 2021).

Narang, N. and Bourlai, T. (2015): Face Recognition in the SWIR Band When Using Single Sensor Multi-Wavelength Imaging Systems. Image and Vision Computing, Vol. 33, pp. 26-43. ISSN 0262-8856. Available at: https://doi.org/10.1016/j.imavis.2014.10.005.

Park, H. S. and Jun, C. H. (2009): A Simple and Fast Algorithm for K-Medoids Clustering. Expert Systems with Applications, Vol. 36, pp. 3336–3341. doi: 10.1016/j.eswa.2008.01.039. Available at: https://www.sciencedirect.com/science/article/abs/pii/S095741740800081X (accessed Oct. 17, 2021).

Qi, J., Yu, Y., Wang, L., Liu, J., and Wang Y. (2017): An Effective and Efficient Hierarchical K-Means Clustering Algorithm. Sage J., Vol. 13, No. 4. doi:10.1177/1550147717728627. Available at: https://journals.sagepub.com/doi/full/10.1177/1550147717728627 (accessed Oct. 17, 2021).

Rajkumar, S. and Malathi, G. (2016): A Comparative Analysis on Image Quality Assessment for Real Time Satellite Images. Indian J. of Science and Technology, Vol. 9, pp. 22-38. doi:10.17485/ijst/2016/v9i34/96766. Available at: https://www.researchgate.net/publication/308667046_A_Comparative_Analysis_on_Image_Quality_Assessment_for_Real_Time_Satellite_Images (accessed Oct. 17, 2021).

Saha, R., Tariq, M. T., Hadi, M., and Xiao, Y. (2019): Pattern Recognition Using Clustering Analysis to Support Transportation System Management, Operations, and Modeling. J. of Advanced Transportation, Vol. 2019, pp. 1-12. doi:10.1155/2019/1628417. Available at: https://www.researchgate.net/publication/338256676_Pattern_Recognition_Using_Clustering_Analysis_to_Support_Transportation_System_Management_Operations_and_Modeling (accessed Oct. 17, 2021).

Sethi, P. and Alagiriswamy, S. (2010): Association Rule-Based Similarity Measures for the Clustering of Gene Expression Data. The Open Medical Informatics J., Vol. 4, pp. 63–73. doi:10.2174/1874431101004010063. Available at: https://www.researchgate.net/publication/51156759_Association_Rule_Based_Similarity_Measures_for_the_Clustering_of_Gene_Expression_Data (accessed Oct. 17, 2021).

Silva E. et al. (2007): Quantifying Image Similarity Using Measure of Enhancement by Entropy. Mobile Multimedia/Image Processing for Military and Security Applications 2007, Vol. 6579, pp. 1-48. doi: 10.1117/12.720087. Available at: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.160.6126&rep=rep1&type=pdf (accessed Oct. 17, 2021).

Shorten, C. and Khoshgoftaar,T. (2019): A Survey on Image Data Augmentation for Deep Learning. J. of Big Data, Vol. 6, No. 1. doi: 10.1186/s40537-019-0197-0. Available at: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0 (accessed Oct. 17, 2021).

Um T. et al. (2017): Data Augmentation of Wearable Sensor Data for Parkinson's Disease Monitoring Using Convolutional Neural Networks, in Proc.19th A.C.M. Int. Conf. on Multimodal Interaction - ICMI 2017, Vol. 1, pp. 216–220. doi: 10.1145/3136755.3136817. Available at: https://arxiv.org/pdf/1706.00527.pdf (accessed Oct. 17, 2021).

Vo, Q. V., Hoang, M. T., and Choi, D. (2013): Personalization in Mobile Activity Recognition System Using K-Medoids Clustering Algorithm. Int. J. of Distributed Sensor Networks, Vol. 9(7). doi: 10.1155/2013/315841. Available at: https://journals.sagepub.com/doi/full/10.1155/2013/315841 (accessed Oct. 17, 2021).

Wang, Z. and Simoncelli E. (2005): Translation Insensitive Image Similarity In Complex AW, in Int. Conf. on Acoustics, Speech, and Signal Processing, ICASSP-88, 2:573–576. doi: 10.1109/ICASSP.2005.1415469. Available at: https://www.researchgate.net/publication/4137003_Translation_Insensitive_Image_Similarity_in_Complex_Wavelet_Domain (accessed Oct. 17, 2021).

Wang, Z. et al. (2004): Image Quality Assessment: From Error Visibility to Structural Similarity, in IEEE Transactions on Image Processing, Vol. 13, No. 4, pp. 600-612. doi: 10.1109/TIP.2003.819861. Available at: https://www.cns.nyu.edu/pub/lcv/wang03-preprint.pdf (accessed Oct. 17, 2021).

Ye, P. and Doermann, D. (2012): No-Reference Image Quality Assessment Using Visual Codebooks, in IEEE Transactions on Image Processing, Vol. 21(7), pp. 3129-3138. doi: 10.1109/TIP.2012.2190086. Available at: https://ieeexplore.ieee.org/document/6165361 (accessed Oct. 17, 2021).