IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 4 July 2022, accepted 21 July 2022, date of publication 16 August 2022, date of current version 22 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3198992

RESEARCH ARTICLE

A Distance Metric for Uneven Clusters of Unsupervised K-Means Clustering Algorithm

MOSTAFA RAEISI AND ABU B. SESAY, (Life Senior Member, IEEE)
Department of Electrical and Software Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada

Corresponding author: Mostafa Raeisi (mostafa.raeisiziaran@ucalgary.ca)

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada; and in part by the Schulich School of Engineering, University of Calgary.

ABSTRACT In this paper, we propose a new distance metric for the K-means clustering algorithm. Applying this metric in clustering a dataset, forms unequal clusters. This metric leads to a larger size for a cluster with a centroid away from the origin, rather than a cluster closer to the origin. The proposed metric is based on the Canberra distances and it is useful for cases that require unequal size clusters. This metric can be used in connected autonomous vehicle wireless networks to classify mobile users such as pedestrians, cyclists, and vehicles. We use a combination of mathematical and exhaustive search to establish its validity as a true distance metric. We compare the K-Means algorithm using the proposed distance metric with five other distance metrics for comparison. These metrics include the Euclidean, Manhattan, Canberra, Chi-squared, and Clark distances. Simulation results depict the effectiveness of our proposed metric compared with the other distance metrics in both one-dimensional and two-dimensional randomly generated datasets. In this paper, we use three internal evaluation measures namely the Compactness, Sum of Squared Errors (SSE), and Silhouette measures. These measures are used to study the proper number of clusters for each of the K-Means algorithms and also select the best run among multiple centroid initializations. The elbow method and the local maximum approach are used alongside the evaluation measures to select the optimal number of clusters.

INDEX TERMS Canberra distance, chi-squared distance, clustering algorithm, distance metrics, Euclidean distance, K-means algorithm, unsupervised learning.

I. INTRODUCTION

Clustering is an algorithm in unsupervised learning to classify data points into multiple groups based on points' similarity. There are several clustering algorithms such as K-means [1], K-medians [2], Mean-shift clustering [3], Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [4]. K-means is the most widely used algorithm for clustering.

K-means algorithm is used to classify points of a dataset into K sub-groups, based on their similarities, in an iterative approach. The K-means algorithm's clustering results depends on two factors, the initial centroid points and distance metric. The distance metric is the metric applied to measure the distance between data points and cluster centroids [5].

The choice of distance metric depends on the application, dataset and the desired output. Datasets may contain numerical or categorical values. For categorical data, some researchers propose distance metrics to improve the clustering algorithm's performance [6] and [7]. On the other hand, the Euclidean, Manhattan, Chebychev, Canberra, Chi-Squared and some other distance metrics are used in numerical data types.

A K-means algorithm with Euclidean distance as a metric, sets the separation boundary between two adjacent clusters, equidistant from the two centroids and form equal size clusters. The Manhattan and Minkowski metrics, that can

The associate editor coordinating the review of this manuscript and approving it for publication was Chao Tong.

have equal size clustering results similar to the Euclidean distance, have been evaluated in [5] and their performance compared with the Euclidean distance metric for the K-means algorithm. These metrics are presented in the next section with more details.

Creating unequal size clusters has been investigated in some papers. In [8] the authors use the K-means clustering algorithm and then by either merging or splitting the clusters, formed unequal size clusters. They applied their proposed clustering method in routing protocols of wireless sensor networks for Internet of Things (IoT) applications. Unequal length clustering for improved wind power prediction is studied in [9]. The Canberra distance proposed by Lance and Williams in [10] is a common distance metric that can be used in K-Means algorithm to form unequal size clusters. The application of Canberra distance for clustering of different databases is studied in [11].

Finding an appropriate distance metric for a K-means algorithm to properly classify a dataset is challenging. Some researchers have proposed the use of learning algorithms to find the metric iteratively [1], [12], [13], [14], [15], [16]. In [12], Xing *et al.* propose a metric learning algorithm by learning a scaling matrix. They use a training dataset to learn similar sample relations and then find the coefficients of their scaling matrix to re-arrange the dataset. This algorithm separates the data points and reduces the data dimensions if needed. Nguyen and De Baets in [1] propose a distance metric learning method based on kernels in a non-linear feature space. A learning method for semi-supervised clustering using background information under prior knowledge is proposed by Jing *et al.* in [13].

Although learning algorithms may find a proper distance metric in some cases, they have some issues and limitations. These algorithms need a training dataset to provide the algorithm with similar and non-similar points. Then, they use these training data to find the pattern and metric. Therefore, they have a supervised learning phase to first determine the metric. Providing a training dataset is not always a possible or available option as there could be significant changes in the input data characteristics. Moreover, learning imposes more processing tasks that use up additional resources and could increase the computational burden and time complexity.

Road users' clustering has been investigated in some papers to improve road efficiency, [14] and [15]. In [15], authors cluster road users based on their speeds to predict the traffic velocity more accurately. They use matrix factorization of the observed speed matrix for user clustering in space and time. In this study, their dataset consists of the average velocity of 1190 road sections in Pittsburgh and 1091 segments in Washington, D.C. in five-minute intervals for a month. Therefore, they only consider cars and cyclists, not pedestrians and their average speed over a five-minute time span, and not individual vehicles or bicycles separately.

In this paper, we propose a new distance metric to be used in the K-Means algorithm to group numerical datasets with unequal clusters. Unlike the K-Means with the Euclidean metric which set the separate-boundary of two adjacent clusters in the middle point of the imaginary direct line between their cluster centroids, the K-Means with our new metric sets the decision boundary towards the centroid closer to the origin. In all the K-Means algorithms in this paper, the centroid of each cluster is calculated using the arithmetic means of the assigned data points in that cluster. Compared to the Canberra metric, the proposed metric is computationally more intensive because it has an extra square root in the denominator. In terms of cluster sizes, our proposed metric's cluster areas get wider as the centroids get further away from the origin compared with the Euclidean metric while it is smaller than the Canberra metric. We use an exhaustive search to demonstrate the validity of the proposed distance metric. This metric can be used in autonomous vehicles' wireless communication to distinguish low-velocity pedestrians from fast-speed vehicles. Our main contributions are as follows:

- Proposal of a new distance metric that can be used in the K-Means clustering algorithm in applications such as wireless networks and vehicular communications.
- Distance criterion has been investigated for the proposed distance metric and proved as a valid metric.
- The performance of the proposed metric is investigated and compared with similar metrics using generated datasets and a model-based dataset through simulation.
- Investigation of three different evaluation measures for selecting the appropriate number of clusters.
- Investigation of the Canberra, Clark, and Chi-Squared metrics issues with opposite sign values.

The rest of this paper is organized as follows. Section II presents the preliminary information about the K-Means algorithm and different distance metrics used in this paper. It also provides the basics of the three evaluation measures including the Compactness, SSE, and the Silhouette criterion. In Section III, the proposed distance metric is presented and an investigation of the distance metric criterion is provided. Simulation results to compare the performance of our proposed metrics and the Euclidean, Canberra, Manhattan, Chi-squared, and Clark metrics are provided in Section IV. Finally, the last section provides the conclusion of the paper.

A. NOTATIONS

$X = \{x_1, x_2, \ldots, x_I\}$ denotes a data set of size I where each component is of dimension N, that is, $x_i = \{x_{i_1}, x_{i_2}, \ldots, x_{i_N}\}$. Also, let $C = \{\mu_1, \mu_2, \ldots, \mu_K\}$ denote a set of K cluster centroid positions where each centroid component value is of dimension N, that is, $\mu_i = \{\mu_{i_1}, \mu_{i_2}, \ldots, \mu_{i_N}\}$. C_k is the k^{th} cluster with I_k data points in it which is a subset of C.

II. PRELIMINARIES

Clustering and classification are both algorithms for grouping similar points or objects. However, Classification is a supervised learning algorithm while clustering is an unsupervised learning algorithm. Clustering algorithms are divided into

five major groups named partitioning-based, hierarchical-based, density-based, grid-based, and model-based methods. K-Means, K-Medoids, K-Modes, PAM, CLARANS, CLARA, FCM, and CluStream are the main partitioning-based algorithms.

K-MEANS

K-Means algorithm is a method for partitioning unlabelled data points into K groups called clusters. This algorithm consists of five steps that recursively searches for the local optimum point for cluster centres also known as centroids as listed below.

1) Select number of clusters K
2) initialize centroids
3) Distance calculation
4) point assignment
5) update centroid location

The number of clusters in some applications is predetermined and the first step is already solved. However, in many other applications, it is not fixed and a range of numbers is acceptable for K. In these cases, using the clustering evaluation metrics helps to find an efficient K based on the conditions that are explained in the rest of this section.

After selecting the number of clusters, K centroids are picked. There are multiple methods to select the start points of centroids. These methods include randomly selecting K points in an N dimensional space, randomly selecting K data points from an input data set, or picking the K most likely points for centroids and letting the algorithm adjust them. The latter method applies in cases where there is knowledge about the most expected starting point for centroids. To avoid eliminating a cluster, these K points should not be equal to each other at the initialization step. Since K-means is a non-convex algorithm, it does not always guarantee the optimal clusters. Therefore, it is recommended to initialize the centroid points randomly, multiple times and run the algorithm and then, select the best run according to the evaluation metrics.

The K-Means algorithm uses a distance metric to calculate the distance between each of the I input data points and each of the K centroids and assign the i^{th} point (x_i) to the centroid with minimum distance. The distance metric, in the majority of the application, is the Euclidean distance and works fine. However, in some applications, it is required to use a distance metric that enhances the K-Means performance. The five distance metrics that are used in this paper, are described in this section.

After distance calculation between each data point and all the centroids, the algorithm assigns each point to a cluster with the minimum distance to its centroid. Hence, the selection of a proper distance metric plays an important role in distance values, assignment of the points to each cluster, and the overall size of each cluster.

The last step is updating the cluster centroids according to the assigned data points in step four. In this step we find the arithmetic mean of the assigned points and set it as the updated centroid location. If there is insignificant difference between all the centroids' previous and new locations, the algorithm has reached a stable condition. However, if there is a significant difference between the last and updated location of each centroid, the algorithm goes to step 3 and repeats the process. So, the K-Means is a recursive algorithm that repeats the last three steps till it converges to a stable condition.

A. DISTANCE METRICS

The most suitable distance metrics to compare our performance with are the Euclidean, Manhattan, Canberra, Chi-Squared, and Clark distances.

1) EUCLIDEAN

The Euclidean distance, also known as the Pythagorean distance, is a special case of the Minkowski distance with an order of two ($p = 2$). The Euclidean distance between two N dimensional points (x and y) is the square root of the sum of squared errors in each dimension as presented in (1).

$$d(x, y) = \sqrt{\sum_{n=1}^{N} (x_n - y_n)^2} \tag{1}$$

2) MANHATTAN

The Manhattan distance given in (2), is the sum of absolute errors over all dimensions. It is known by many other names such as taxicab, city block, or L_1 distance.

$$d(x, y) = \sum_{n=1}^{N} |x_n - y_n| \tag{2}$$

3) CANBERRA

This distance was first developed and modified by Lance and Williams in 1966 and 1967 respectively without the absolute values in the denominator. The current Canberra formula with absolute values in the denominator was also presented in their 1967 paper and called it Adkins. However, in the literature, the Canberra distance is defined as follows.

$$d(x, y) = \sum_{n=1}^{N} \frac{|x_n - y_n|}{|x_n| + |y_n|} \tag{3}$$

The Canberra distance has a problem with opposite sign numbers along an axis (e.g., $x_n = |x_n|$, $y_n = -|y_n|$) and the distance is always 1 over that dimension regardless of their absolute values. Therefore, in an N dimensional space, for any two points in opposite quarters with respect to the origin, the distance is N. It also happens for a distance of any point to the origin. Hence, the triangle inequality condition of a valid distance is not satisfied for the Canberra distance and proves that it is not a valid distance metric for opposite sign values. This problem is further explained in Section IV.

4) CHI-SQUARED

The Chi-Squared distance or squared chi-Squared distance is a distance metric that can form unequal size clusters.

In this metric, the denominator slows down the increase in the distance when the absolute values of x and y increase, which results in uneven clustering when used in the K-Means algorithm. This distance is defined as:

$$d(x, y) = \sum_{n=1}^{N} \frac{(x_n - y_n)^2}{x_n + y_n} \tag{4}$$

The Chi-squared distance has an issue with two symmetric points with respect to the origin and the distance go towards infinity in these cases ($x = -y + \epsilon$). As a result, the triangle inequality of a valid distance is not satisfied for opposite signed values. Also, without an absolute value in the denominator, it violates the non-negative distance condition of a valid distance.

5) CLARK

The Clark distance or the coefficient of divergence is the square root of the sum of separate squared terms in Canberra distance. In other words, it is the square root of the sum of squared normalized errors over all dimensions as illustrated in (5) according to [17].

$$d(x, y) = \sqrt{\sum_{n=1}^{N} \left(\frac{x_n - y_n}{|x_n| + |y_n|}\right)^2} \tag{5}$$

Since each term of the Clark distance is the square of the equivalent term of the Canberra distance, it has the same issues as the Canberra distance.

B. EVALUATION MEASURES

In clustering algorithms such as the K-Means, mostly the centroids are initialized randomly and on each run, the clustering can end up with a different local optimum. To compare the performance of an algorithm in various initializations or to compare multiple algorithms' proficiency, evaluation measures are used. Compactness, Sum of Squared Errors (SSE), and Silhouette are the internal measures methods that are used in this paper to evaluate the results.

In many applications and datasets, the best choice of the number of clusters is not easy to guess. Local maximum, local minimum, and the elbow method are the rules to select the number of clusters but each of them works for a set of evaluation measures. The elbow method is used with the compactness and the SSE measures while the local maximum is used for the Silhouette to find the proper number of clusters. In this paper, we use the elbow method and the local maximum to investigate the appropriate number of clusters for our dataset.

1) COMPACTNESS

Compactness or cluster cohesion defined in (6), is a measure that relies only on input variables without labels.

$$\text{CMP} = \frac{1}{K} \sum_{k=1}^{K} \sum_{\substack{i=1 \\ x_i \in C_k}}^{I_k} \frac{d(x_i, \mu_k)}{I_k} \tag{6}$$

In (6), μ_k is the centroid location of the k^{th} cluster (C_k) and I_k is the number of points in C_k. The ultimate goal of the clustering algorithm is to minimize the compactness value by selecting the best location for the centroids.

2) SUM OF SQUARED ERRORS (SSE)

SSE is another evaluation measure that is similar to the compactness but instead of the averages, it adds up the squared distances. It should be noted that to keep the fairness among K-Means with various distance metrics, we use the same distance metric to measure the error accordingly.

$$\text{SSE} = \sum_{k=1}^{K} \sum_{\substack{i=1 \\ x_i \in C_k}}^{I_k} d^2(x_i, \mu_k) \tag{7}$$

Similar to the compactness measure, in the clustering algorithm that uses the SSE measure, the target is the minimum value of SSE by selecting the optimal centroid locations.

3) SILHOUETTE

Silhouette is a measure used to evaluate the integrity and quality of the clusters. The Silhouette equations are summarized in (8)-(11).

$$\text{SIL} = mean(\text{SIL}(x_i)) \qquad i = 1, 2, \ldots, I \tag{8}$$

$$\text{SIL}(x_i) = \frac{b(x_i) - a(x_i)}{\max\{a(x_i), b(x_i)\}} \tag{9}$$

$$a(x_i) = \frac{\sum_{j=1, j\neq i}^{I_k} d(x_i, x_j)}{I_k - 1} \tag{10}$$

$$b(x_i) = \min_{1 \leq k' \leq K, k' \neq k} \left\{ \frac{\sum_{j=1, j\neq i}^{I_{k'}} d(x_i, x_j)}{I_{k'}} \right\} \tag{11}$$

In (8), $I = \sum_k I_k$ is the total number data points in the dataset and I_k in (10), is the number of points in k^{th} cluster that x_i exists. Also, $I_{k'}$ is the number of data points in k'-th cluster to which x_i is not assigned. According to (9), the Silhouette measure calculates I values. Therefore, it is difficult to compare two clustering algorithms with Silhouette measure while we have I values for each of them. In this paper to have a single value for silhouette measure to compare the different distance metrics, we use the average of silhouette values of all the points according to (8).

III. PROPOSED DISTANCE METRIC

The proposed distance metric is the sum of absolute values of the difference between two points divided by the square root of the summation of their absolute values in each dimension as shown in (12). This metric is obtained by adding a square root in the denominator of each term in the Canberra distance, which solves the issue of the Canberra distance as explained before. Moreover, the proposed distance metric does not have the issues of the Clark and Chi-Squared distances as explained above. The addition of the square root also decelerates the increase of the denominator while the numerator has not changed. Therefore, the distance in our metric is bigger

than the Canberra when the absolute value of the sum of points is larger than one ($|x_{i_n}| + |\mu_{k_n}| \geq 1$). As a result, clusters' region of the K-Means with our metric are wider than clusters with the Canberra distance for clusters close to the origin.

$$J_{ik}(x_i, \mu_k) = \sum_{n=1}^{N} \frac{|x_{i_n} - \mu_{k_n}|}{\sqrt{|x_{i_n}| + |\mu_{k_n}|}} \tag{12}$$

The metric for one dimension is shown in (13).

$$d(x, y) = \frac{|x - y|}{\sqrt{|x| + |y|}} \qquad \forall\, x, y \neq 0 \tag{13}$$

Theorem 1: Equation (13) is a distance metric if on a given set S, a function d, maps $S \times S \rightarrow \mathbb{R}$, where $\mathbb{R}$ denotes the set of real numbers. Also, d must satisfy the following conditions:

- $d(x, y) \geq 0$
- $d(x, y) = 0$ if and only if $x = y$
- Symmetric: $d(x, y) = d(y, x)$
- Triangle inequality: $d(x, z) \leq d(x, y) + d(y, z)$

In this paper, we consider x and y to be real numbers. However, both of them cannot be zero at the same time $\{x, y \in S = \mathbb{R} | x, y \neq 0\}$.

The mapping space from S to real numbers and proofs for the first three conditions of a distance metric are provided in Appendix V. Validity of the triangle inequality is investigated below using a Brute-force algorithm.

A. TRIANGLE INEQUALITY

To prove the triangle inequality, we need to show that $d(x, z) \leq d(x, y) + d(y, z)$ is true. However, the square root in this distance metric, makes direct mathematical solution tremendously difficult. Hence, we use an exhaustive search approach to validate the triangle inequality through simulations. We compute $d(x, y)$ for all values of x and y queuerated. The distance between two corresponding values of x and y on x-axis and y-axis is depicted along z-axis in Figure 1.

As expected, this distance metric is symmetrical with respect to the $x = y$ line which also confirms the third condition in Theorem 1 as we provided in Lemma 4 of Appendix V. The distance values are always greater or equal to zero as in the first condition of Theorem 1 and in accordance with Lemma 2 of Appendix V. The most important feature is its behaviour with increasing x and y. The growth rate of this distance decreases with increasing x and y. Accordingly, the edge of the plane bends downwards on the far right and far left of Figure 1 which means less distance for larger numbers compared to the Euclidean distance.

This characteristic can be used to confine the cluster size for clusters closer to the origin. To clarify this feature, consider a point with equal Euclidean-distance from two centroids where one centroid is closer to the origin and the other one is further away. With the Euclidean distance metric, this data point can be assigned to each of these two clusters. While, with our distance metric in (12), for a centroid closer

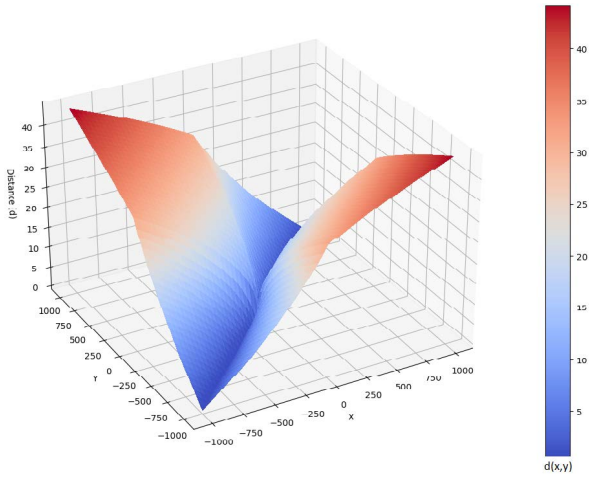

FIGURE 1. Distance between x and y according to the proposed distance metric-II.

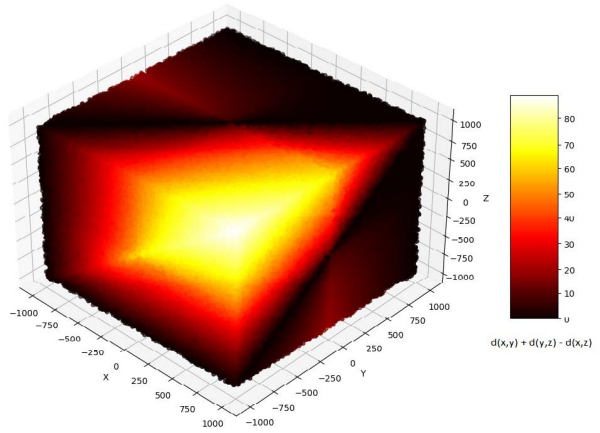

FIGURE 2. 4D plot of triangle inequality for the proposed metric with three one-dimensional points x, y, and z. The color shows the value of $d(x, y) + d(y, z) - d(x, z)$.

to the origin, the denominator is smaller as the value of μ_{k_n} is smaller. Hence, the distance between the data point and the cluster closer to the origin is bigger than the other cluster, so the data point will be assigned to the cluster further away from the origin. Therefore, the cluster away from the origin covers wider area than the closer one.

In Figure 2, the difference between the sum of two arbitrary sides of an x-y-z triangle and the third side according to the (14), based on the triangle inequality, is illustrated for one million random points of x, y, and z. Without loss of generality, each of x, y, and z points are randomly selected in a range of -1000 to 1000 ($x, y, z \in [-1000, 1000]$). This range can be generalized to any arbitrary ranges without negatively impacting the validity of triangle inequality as proved in Appendix V. Figure 2 also shows the values of (14) in color where the darker color indicates smaller values. As the colorbar on the right side of Figure 2 shows, there is no single point with a negative difference in (14). This means that the

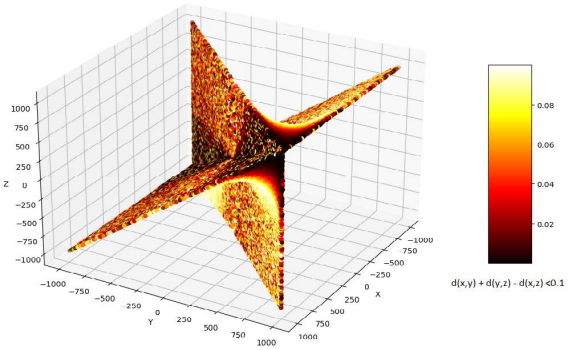

FIGURE 3. 4D plot of triangle sides difference for three one-dimensional points provided it is less than 0.1.

triangle inequality is always valid for the proposed metric.

$$d\,(x, y) + d\,(y, z) - d\,(x, z) \tag{14}$$

To further investigate the validity of the triangle inequality, we focus only on areas in Figure 2 that have very small values. Using the same concept as in Figure 2, we generate a 4D plot where the values of (14) are constrained to an upper limit of 0.1. This time, to increase the sample space, we generated five million random numbers and only displayed results that are less than 0.1 (close to zero) and the result shown in Figure 3. Based on the proposed distance metric in (12) and Figure 3, near zero values of (14) are around two hyperplanes and small areas around their intersection. These two hyperplanes are $x = y$ and $y = z$.

We use Algorithm 1 for a broader range of numbers in the above mentioned low difference areas in Figure 3. This algorithm helps to further investigate the validity of the triangle inequality for the proposed metric. To avoid unnecessary searches and increase the accuracy, this algorithm only focuses on near zero areas described above and visually investigated in Figure 3 with higher resolution.

Algorithm 1 Triangle Inequality Test Algorithm

```
for _ in range(1e9): do
    x = rand(−1e8, 1e8);
    if randn() > 0 then
        y = randn() ∗ x ∗ 0.01 + x;
        z = rand(−1e8, 1e8);
    else
        y = rand(−1e8, 1e8);
        z = randn() ∗ y ∗ 0.01 + y;
    end
    if d(x, y) + d(y, z) >= d(x, z) then
        Continue;
    else
        print("Error");
        break;
    end
end
```

In Algorithm 1, in the main for-loop, one billion data points, uniformly distributed, are generated for x. Then, y and z random values based on x around one of the two planes of $x = y$ and $y = z$ are generated. This algorithm also covers the areas around the intersection of these two planes that could have small distance differences. Finally, this algorithm checks the triangle inequality for each set of points. If a set (x, y, z) of points exists that does not satisfy the triangle inequality, the algorithm prints an error message and breaks the loop. Implementation of Algorithm 1 using Python programming language proved the triangle inequality for the proposed distance metric with no error returned.

Since the proposed metric satisfies the distance criteria presented in Theorem 1, using this metric in K-Means algorithm with deterministic point assignment guarantees the convergence of the algorithm.

IV. SIMULATION RESULTS FOR CLUSTERING

In this section, we compare the clustering results of the K-means algorithm using our proposed metric and other metrics such as the Euclidean, Manhattan, Canberra, Chi-Squared, and the Clark metrics on various datasets representing the road users of a cellular network. Also, we evaluate the clustering validity using three different internal clustering evaluation measures for all the distance metrics. These evaluation measures are Compactness, Sum of squared error, and Silhouette as explained in section II.

Since these metrics' performance on clustering a dataset are tightly similar to each other, using a real dataset can not reveal and illustrate the differences very well. Moreover, to the best of our knowledge, there is no dataset in our targeted application that consists of all pedestrians, bicycles, and cars of different velocities on a road and it is not possible for authors to collect such a dataset. Therefore, let us first assume that speeds of cars, cyclists and pedestrians can all be modelled as uniformly distributed random numbers. We then generate a dataset with 10, 000 data points that are uniformly distributed in a range of $[-100, 100]$ to represent road users' speeds (in Km/h) on a two-way road in a very busy condition. The positive range represents one direction of movement (e.g., from left to right) and the negative range represents the opposite direction (e.g., from right to left). The 10, 000 points are generated to span, without gaps, the entire range $[-100, 100]$. It is difficult to find a road that has large number of cars with speeds up to 100 Km/h, cyclists with a wide range of speeds, and a big group of people moving at various speeds in both directions of the road while the entire dataset follow the uniform distribution. However, we shall use such a scenario to test how the various metrics can cluster these speeds.

One objective is to cluster into three groups ($K = 3$) with lower speed users, such as pedestrians, low speed cyclists, and stationary/slow cars, in any direction into one cluster while higher speed users, such as fast driving cars and cyclists, in opposite directions are clustered into two separate clusters.

The second objective is to cluster into four groups ($K = 4$) where slow moving users such as pedestrians, stationary/slow cars, and slow cyclists in opposite directions are clustered into two separate clusters while fast moving objects, including cars and cyclist, in opposite directions are clustered into two separate clusters.

Figures 4 and 5 illustrate clustering results of the K-means algorithm using six different distance metrics that cluster the dataset into three and four clusters respectively. The clusters are represented by the colors red, green, blue, and cyan. In Figure 4, blue represents the cluster of points representing high-speed users in one direction while red represents the cluster of points for high-speed users in the opposite direction. Green represents the cluster of points for slow moving objects. The centroid of each cluster is indicated with a black vertical mark in the middle of each cluster. In Figure 5, cyan represents the cluster of points representing high speed users in one direction while red represents the cluster of points for high speed users in the opposite direction. Blue represents the cluster of points for slow moving objects in one direction while green represents slow moving objects in the opposite direction.

For the 3-cluster case in Figure 4, the proposed distance (Figure 4-a), the Euclidean distance (Figure 4-b) and the Manhattan distance (Figure 4-d) produce the desired clusters, i.e., a cluster for both direction of slow moving objects (the green cluster) and two clusters for high speed users in each direction (the red and blue clusters). The Canberra (Figure 4-c), the Chi-Square (Figure 4-e) and the Clark (Figure 4-f) metrics combine both slow moving and fast moving cars, in the positive direction, into one cluster (the blue), which is not as explained above. On the opposite side, the red clusters of these three metrics, cover high speed cars as is desired while the green clusters are not acceptable because they only cover part of slow-moving users.

Figure 5 shows the clustering results for the 4-cluster case, as an only simulation with an even number of clusters in this paper, to have a reference for all six metrics' performances for even number of clusters. In this figure, all the metrics produce two clusters in positive values of X and two clusters for negative values of X as desired. The Canberra and Clark distances in Figures 5-c and 5-f respectively, have the smallest clusters around the origin for low speed users (the green and blue clusters). Whereas, the Euclidean and Manhattan distances (Figures 5-b and 5-d) have largest clusters in the middle. The proposed and the Chi-Squared distances' clusters for low speed users (in Figures 5-a and 5-e) are smaller than the Euclidean and Manhattan distances and larger then the Canberra and Clark distances. However, to separate slow-moving road users such as pedestrians from fast-moving objects like cars, an even number of clusters is not desirable as we need all pedestrians in one group. A pedestrian can stop or change direction at any moment and it is better to just put them all in one group.

The cluster widths for all metrics are, however, different in Figures 4 and 5. The proposed distance, Euclidean, and

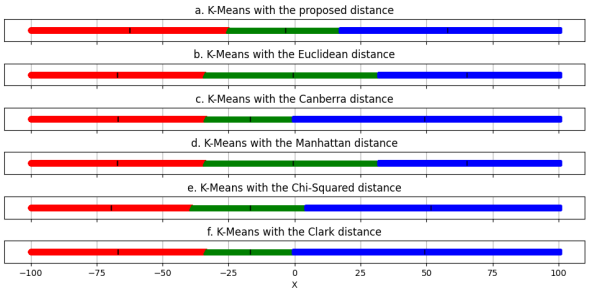

FIGURE 4. One-dimensional data clustering using K-Means algorithm for three clusters.

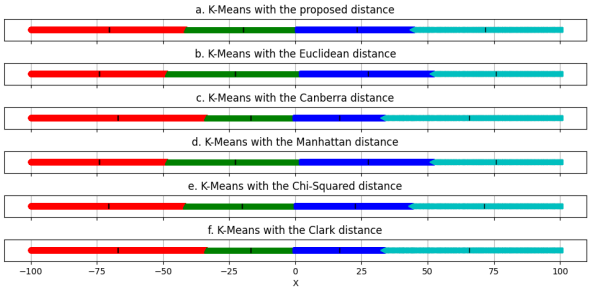

FIGURE 5. One-dimensional data clustering using K-Means algorithm for four clusters.

Manhattan metrics are only metrics that have desired results in both cases of an even and odd number of clusters. The proposed metric produces smaller cluster around zero compared with the other two metrics, which works well for applications such as separating low speed users from high speed users. The Euclidean and the Manhattan metrics produce wider clusters. The Canberra and the Clark metrics' clusters are shifted considerably to one side with respect to the origin (for odd number of clusters like in Figure 4) because of these metrics' issue along any dimension in cases of either two points with opposite signs or one point equal to zero and another point taking on any other value. The problem with the Chi-Squared metric in Figure 4-e, is also with negative numbers, which for two points that have very close absolute values but different signs ($x = -y + \epsilon$), the distance tends towards infinity and results improper clustering. The Chi-Squared metric has another problem with negative numbers that can results in negative distances, which happens if the denominator is less than zero. For the simulations in this paper, we modified the Chi-Squared metric by using the absolute value in the denominator.

To better compare the clustering results of the K-Means algorithm with the proposed distance metric and the other five distance metrics, we generate a dataset, based on real models of road users velocities investigated in the literature. This dataset represent a short section of a road with 500 cellular users in total which include pedestrians, cyclists, and cars. Based on [18], human walking speed on a sidewalk follows the normal distribution with mean values of 4.5 and 4.9 Km/h and standard deviations of 1.5 and 0.75 Km/h in two different sidewalks with different conditions. Therefore, we generate

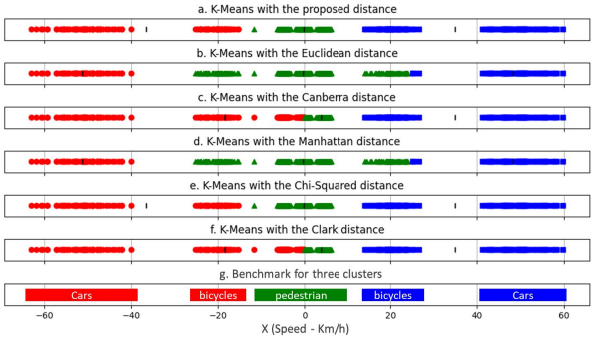

FIGURE 6. K-Means clustering of six metrics for three clusters.

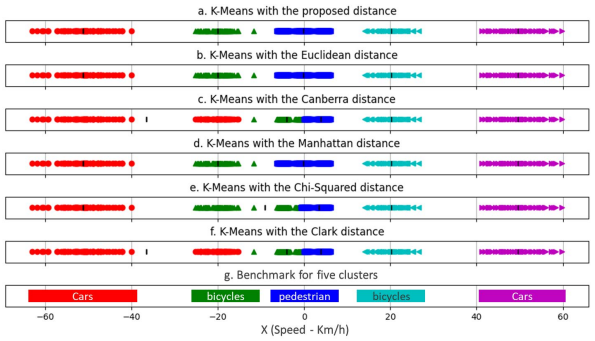

FIGURE 7. K-Means clustering of six metrics for five clusters.

three groups of pedestrians with normal distribution (considering users that are not walking dedicatedly in a specific direction) with mean values of 0.0, 4.5 and -4.9 Km/h and standard deviations of 1.5, 0.75 and 0.68 Km/h respectively. Two of these three groups are for pedestrians walking in each direction of the road on sidewalks and the third group is for people that are standing still or shopping. For cyclists, we use models proposed in [19], [20] and we generate two groups of data points for each direction of the road with normal distribution and mean values of 19.3 and -19.8 Km/h and standard deviations of 3.16 and 2.45 Km/h respectively. Based on the model proposed in [21], cars velocity also follows normal distribution. So for cars, we generate two groups of data points with a normal distribution and mean value of 50 and -55 Km/h and standard deviation of 5 Km/h. Therefore, the dataset is created by merging the above mentioned seven groups of data points together where each group has one-seventh portion of the total 500 users (almost 71 users in each group).

The objective is to separate and group users on the road into three and five clusters. For three clusters ($K = 3$), the objective is to group fast-moving users (cars and cyclists) in opposite directions into two separate clusters while slow-moving users (pedestrians) in either direction are combined into one cluster. For five clusters ($K = 5$), the objective is to group cars in opposite directions into two separate clusters, cyclists in opposite directions into two separate clusters, and pedestrians in any direction into one cluster. This ideal clustering (the benchmark) is illustrated in Figure 6-g and

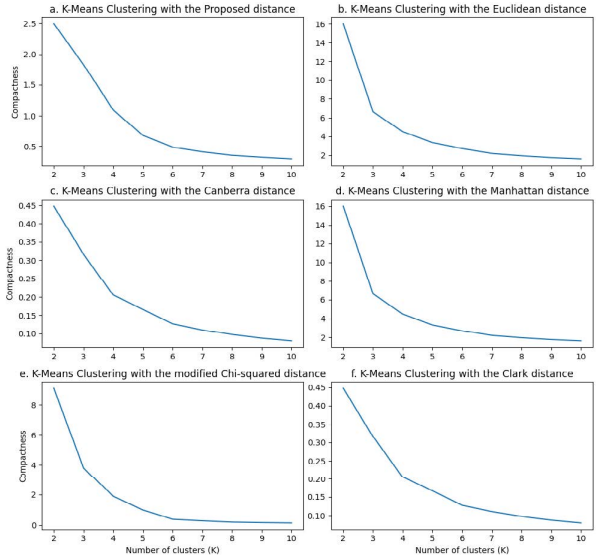

FIGURE 8. Clustering evaluation using Compactness method for K-Means with six different metrics.

Figure 7-g for three and five clusters, respectively. We notice that there are gaps between clusters in the generated dataset and accordingly in the ideal results of the benchmark. This is a result of the relatively small number of data points in the generated dataset (500 users in total, that is, about 71 in each group) that does not cover all speeds. However, if we generated a considerably large number of data points, the gaps between clusters will be filled.

The clustering results for the model-based dataset are shown in Figure 6 for three clusters and in Figure 7 for five clusters. In these simulations, we randomly initialize the centroids for 50 times and select the best initialization based on the compactness evaluation measure. The K-Means clustering results with the proposed distance metric in Figures 6-a and the chi-squared metric in Figure 6-e are accurate results and are the closest to the ideal results. The Euclidean-based and Manhattan-based clustering, shown in Figure 6-b and 6-d, combine the cyclists and pedestrians into one cluster and cars in opposite directions into two separate clusters, which is not the goal. Also, these two metrics group part of the cyclists with the cars in the positive direction (the small blue section attached to the right of the green group in Figures 6-b and 6-d) that is not desirable and makes problems. The clustering results for the Canberra and Clark distances in Figure 6-c and 6-f are also not desirable because of the misclustered points around the origin.

In Figure 7 the clustering results for five clusters (K = 5) is presented and the clustering results of the proposed distance, Euclidean, and Manhattan metrics are the closest to the ideal results. All three distance metrics provide the desired clusters. The Canberra and the Clark distance metrics have similar clustering results. They both provide incorrect clustering results for cyclists (shown in red) and pedestrians (shown in green) in the negative direction. The clustering result of the Chi-Squared distance is also incorrect for low speed

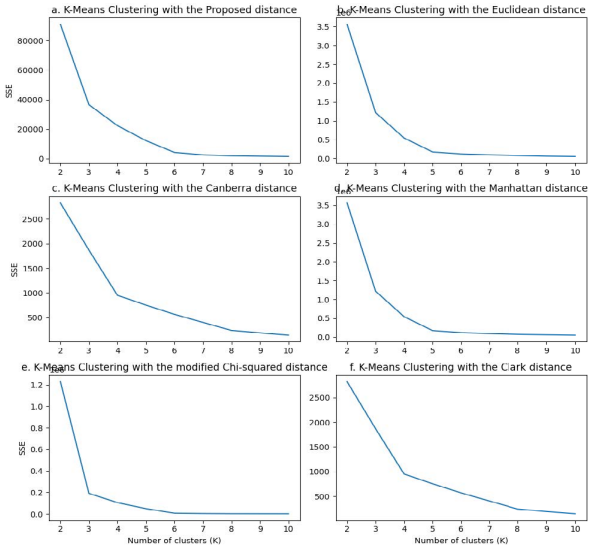

FIGURE 9. Clustering evaluation using SSE method for K-Means with six different metrics.

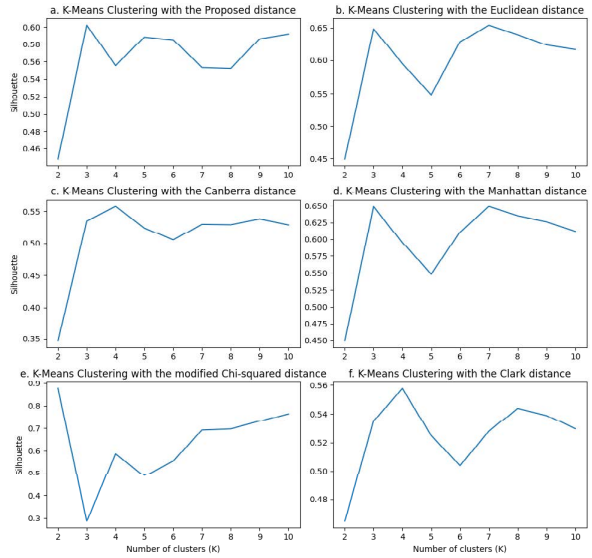

FIGURE 10. Clustering evaluation using Silhouette method for K-Means with six different metrics.

users and it is not desirable. The above simulation results and analyses confirm that the proposed distance metric can improve the K-Means clustering performance in applications with unequal cluster sizes like the road users' example.

In Section II, the basics of evaluation measure and their equations along with their applications were explained. In Figures 8-10 we use these measures to investigate the proper number of clusters for each of the six K-Means algorithms. In these figures, the horizontal axis shows the number of clusters K and the vertical axis is the evaluation measure value. To generate these figures, we simulated 10,000 data points according to the models used in Figures 6 and 7. The number of initializations varies between 100 and 400 times linearly changing for different K values (100 initializations for two clusters that linearly increases to 400 initializations

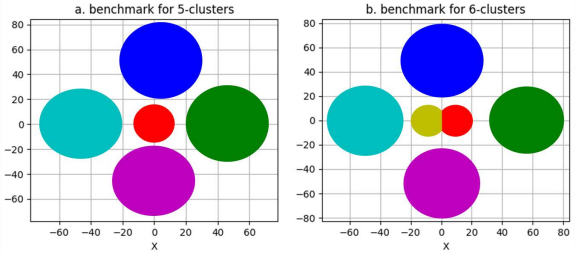

FIGURE 11. Two-dimensional data clustering benchmarks for five and six clusters.

for ten clusters). We also average the evaluation value over the best six percent results for each number of clusters.

Figure 8 shows the compactness evaluation measure values for K-Means clustering with six different distance metrics including our proposed metric for a range of K from 2 to 10 clusters. Using the elbow method we can select the proper number of clusters in each clustering algorithm in Figure 8. In these figures, it is hard to find the elbow point because the input data points are close to each other and it is difficult to confidently select one value for the number of clusters K. However, these figures are still useful to have an idea about the proper value for K in each the clustering algorithms.

In Figure 8-a which is the evaluation result of the K-Means with the proposed metric, five clusters can be considered as the elbow point which satisfies the requirement of an odd number of clusters. In K-Means with the Euclidean and Manhattan distances in Figures 8-b and 8-d, three clusters is a choice based on the graph but it still has a high evaluation measure value with respect to the subsequent number of clusters. For the Canberra, Chi-Squared, and Clark distances in Figures 8-c, 8-e, and 8-f, four and six clusters are the two elbow points. As was expected, their measures for an odd number of clusters are relatively high and these two metrics are not appropriate for these conditions.

Figure 9 shows the clustering evaluation results using the SSE for the six K-Means algorithms in a range of two to ten clusters. Using the SSE measure, the results are slightly different from the compactness method. For clustering with the proposed distance in Figure 9-a, the elbow point could be either three or six but also four and five also could be other good choices as the graph is almost linear between three and six. For the chi-squared in Figure 9-e the elbow point is three. The elbow for the Euclidean and the Manhattan distances is four in Figures 9-b and 9-d, whereas for the Canberra and the Clark distances it could be considered as six in Figures 9-c and 9-f.

Figure 10 illustrates the Silhouette measure results for the K-Means algorithms in a wide range of clusters from two up to ten clusters. As explained above, for the Silhouette measure, the maximum point in the graph shows the proper number of clusters for that dataset. The maximum point in Figure 10-a occurs at three clusters and it shows that the Silhouette measure is a proper measure for cases where a low number of clusters are needed. In Figures 10-b and 10-d

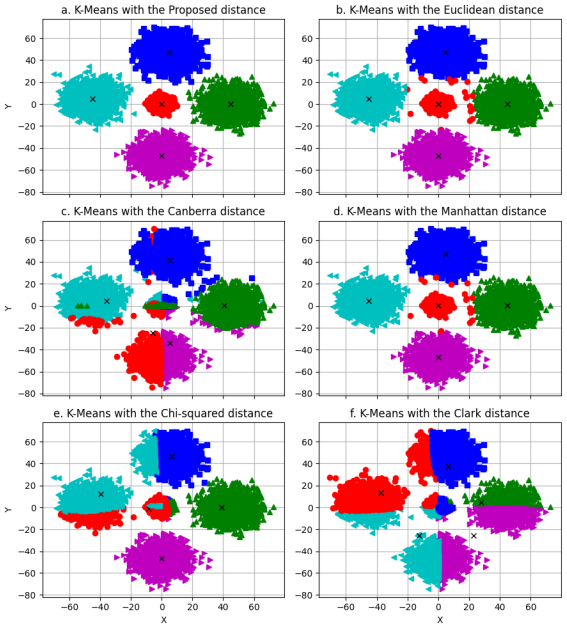

FIGURE 12. Two-dimensional data clustering using K-Means algorithm for five clusters.

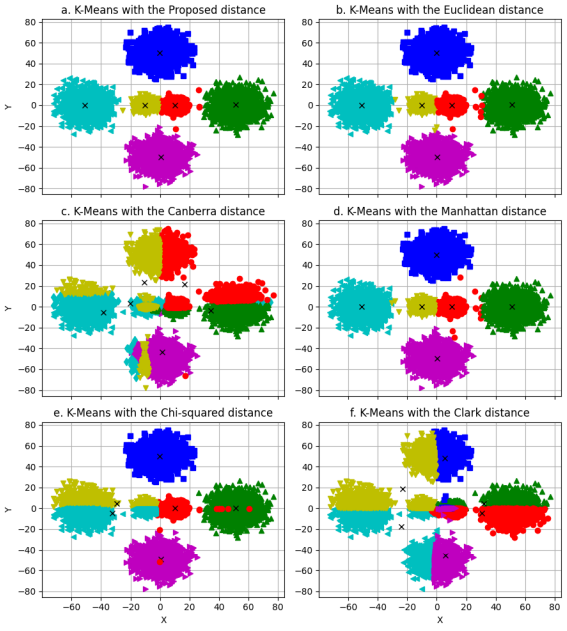

FIGURE 13. Two-dimensional data clustering using K-Means algorithm for six clusters.

for K-Means with the Euclidean and Manhattan distances the maximum happens at both three and seven clusters. The proper number of clusters based on the Silhouette measure for the Canberra and Clark-based clustering is four and for the K-Means with the Chi-squared distance it has a few local maxima but no global maximum.

To evaluate the clustering performance of the proposed metric in a dataset with more than one dimension and compare the results with the other five metrics, we generate two two-dimensional datasets. For the first two-dimensional dataset, we generate 10,000 data points in five equal numbered groups (2000 in each group). One of these five groups is centred on the origin and the other four groups are almost along the X- and Y-axis and nearly 50 units (the blue group is five-unit deviated from the Y-axis) away from the origin (in both directions of positive and negative). For the second dataset, we also generated 10,000 data points in six groups with an equal number of points in each group. Two of these groups are alongside the X-axis and only 10 units away from the origin. The other four groups, similar to the first two-dimensional dataset, are along the X- and Y-axis with distances of 50 units away from the origin. Each of these groups is generated by two independent normal distributions for each dimension and put together to form a two-dimensional space. The standard deviations are 3 and 8 units for the middle and edge groups, respectively. These dimensions (X and Y axis) could represent the velocities and scaled accelerations of wireless users in our considered application in connected autonomous industry. The benchmark for two-dimensional dataset with five clusters is to have one small cluster around the origin and four other larger cluster around the middle cluster as shown in Figure 11-a. For the six-cluster dataset, the goal is to have two small clusters close to the origin with center points in $(-10, 0)$ and $(10, 0)$ and four larger clusters around them, similar to the benchmark diagram in Figure 11-b. The benchmarks in Figure 11 are generated based on the current datasets but if the the dataset generator is run for a significantly large number of times, it will fill the gaps between each two groups.

The clustering results of the two-dimensional datasets for five and six clusters are shown in Figures 12 and 13 respectively. In these figures different colors represent different clusters. For five clusters, the K-Means with the proposed distance metric in Figure 12-a clusters exactly according to the benchmark. The clustering results of the Euclidean distance in Figures 12-b and the Manhattan distance in 12-d have slightly different clustering results compared with the benchmark as there are some red marks close to the side clusters. The Canberra, Chi-Squared and Clark distances in Figures 12-c, 12-e and 12-f have problems for clustering around the origin and along the axis as described before. As a result, the middle group of the data points are clustered in different clusters and have more than one color instead of only red color which is not desirable.

The clustering results for the K-Means algorithm with different distance metrics and six clusters are shown in Figure 13. The proposed metric in Figure 13-a have the closest results to the benchmark and only four points (three red points and one yellow point) are misclustered. The clustering result for the K-Means with the Euclidean and Manhattan distances are shown in Figures 13-b and 13-d respectively. Similar to the clustering results for five clusters, there are some red and yellow color points close to other clusters around the central clusters (green, cyan, and purple clusters) that are not expected according to the benchmark in Figure 11-b and these two metrics have slightly lower

performance compared to the proposed metric. The Canberra, Chi-Squared, and the Clark metric in Figures 13-c, 13-e, and 13-f have very different results compared to the benchmark in Figure 11-b. The issue of Canberra and Clark distances with two points with opposite signs caused two color results for each groups around the origin and mixed colors on the middle clusters. The Chi-Squared distance also has a problem with two points with different signs while their absolute values are very close to each other ($x = -y+\epsilon$) that cause very large distances as explained before. As a result of this problem, the clustering result of the Chi-Squared distance in Figure 13-e is not desirable.

Simulation results of this section show that the proposed distance metric works well in forming unequal clusters that can be used in autonomous industry and any other industries with similar requirement. In this section, we used both one-dimensional and two-dimensional datasets (in total four datasets) to compare the clustering results of the K-Means algorithm with the proposed metric with the K-Means with five other metrics. In these simulations, one dimension represents the road users velocities and the second dimension is equivalent to the scaled acceleration of these users. Since the proposed metric has special application in forming unequal clusters, it is not expected to outperform other metrics on every multi-features dataset. However, it can be applied to higher dimensions in applications that require unequal clusters specifically with both positive and negative values as some distance metrics have fundamental issues in these cases as discussed in this paper.

V. CONCLUSION

In this paper, we have presented a new distance metric that can be used with the K-means clustering algorithm. Our metric generates unequal cluster sizes with smaller clusters closer to the origin and larger clusters for clusters' centroids farther away from the origin compared to the Euclidean distance. We used both a mathematical prove and exhaustive search to prove the validity of the proposed distance metric. We showed that the Canberra, Clark, and Chi-Squared distances violate some distance metric criterion in case of negative values which makes them invalid distances in this case. We compared the K-means algorithm's results with our proposed metric and five other metrics including the Euclidean, Manhattan, Canberra, Chi-Squared, and Clark distance metrics in both one- and two-dimensional datasets. The proper number of clusters was also investigated with three evaluation measures named Compactness, Sum of squared errors (SSE), and Silhouette measures. Simulation results show the effectiveness of the proposed metric in applications with non-linear distance requirements such as clustering datasets with unequal size cluster in wireless and autonomous networks application.

APPENDIX A
VALID DISTANCE METRIC CRITERIA INVESTIGATION

In this section, we provide the mathematical proof on valid distance criterions (including the function space, non-negative condition, coincidence condition, and symmetry condition) for the proposed distance.

Lemma 1: Function space.

In this lemma, we prove the function space mapping between inputs and an output of the proposed distance. Based on the first assumption in Theorem 1 in Section III, both x and y are real numbers. Therefore, the absolute value of their differences (the numerator) is always a real number. The denominator is also a real number as the sum of the sum of the absolute values of x and y are always greater than zero. Hence, the output of the proposed distance metric is always a real number.

$$\begin{aligned} x, y \in \mathbb{R} \Rightarrow |x-y| \in \mathbb{R},\ \sqrt{|x|+|y|} \in \mathbb{R} \\ \Rightarrow d(x,y) \in \mathbb{R} \end{aligned} \tag{15}$$

Lemma 2: Non-negative condition.

The numerator of the proposed distance metric is always greater or equal to zero and the denominator is always greater than zero. Therefore, the distance can not be negative for any case.

$$\begin{aligned} x, y \in \mathbb{R} \Rightarrow |x-y| \geq 0,\ \sqrt{|x|+|y|} > 0 \\ \Rightarrow d(x,y) \geq 0 \end{aligned} \tag{16}$$

Lemma 3: Coincidence condition.

A valid distance metric can be zero only and only if two points are at the same location and have equal values. In our distance metric, the result is zero only and only if the numerator is zero and that only happens if x is equal to y which proves the third condition of a valid distance metric.

$$\begin{aligned} d(x,y) = 0 \Leftrightarrow |x-y| = 0 \\ \Rightarrow x - y = 0 \Rightarrow x = y \end{aligned} \tag{17}$$

Lemma 4: Symmetry condition.

According to the symmetry condition, there should not be any priority and order in metric input points. In other words, the distance between x and y must be equal to the distance between y and x. In (18), we prove the symmetry condition of the proposed distance metric.

$$\begin{aligned} d(x,y) &= \frac{|x-y|}{\sqrt{|x|+|y|}} = \frac{|-(y-x)|}{\sqrt{|y|+|x|}} \\ d(y,x) &= \frac{|y-x|}{\sqrt{|y|+|x|}} = \frac{|-(y-x)|}{\sqrt{|y|+|x|}} \\ &\Rightarrow \quad d(x,y) = d(y,x) \end{aligned} \tag{18}$$

APPENDIX B
TRIANGLE INEQUALITY SIMULATION SCALING

To show that scaling does not affect the triangle inequality and that Figure 2 can be generalized to any arbitrary large area, we use a scaling factor A. Assume A is a positive arbitrary real number and used as scaler for x, y, and z points. Based on (20), although our simulations are limited to the range of -1000 to 1000, it can be scaled to the entire space ($A \to \infty$) and still stay valid.

$$d(Ax, Ay) = \frac{|Ax - Ay|}{\sqrt{|Ax| + |Ay|}}$$

$$= \frac{|A(x-y)|}{\sqrt{A(|x|+|y|)}}$$
$$= \frac{A|x-y|}{\sqrt{A} \times \sqrt{|x|+|y|}}$$
$$d(Ax, Ay) = \sqrt{A}d(x, y) \tag{19}$$

According to (19), the distance between scaled x (Ax) and scaled y (Ay) is linearly related to the distance between x and y with scale factor of $\sqrt{A}$. Therefore, scaling has no impact on triangle inequality.

$$d(Ax, Ay) + d(Ay, Az) \geq d(Ax, Az)$$
$$\sqrt{A}d(x, y) + \sqrt{A}d(y, z) \geq \sqrt{A}d(x, z)$$
$$d(x, y) + d(y, z) \geq d(x, z) \tag{20}$$

Simulations in Figures 2 and 3 plus implementation of the Algorithm 1 in Section III, covered millions of random combinations of x, y, and z in range of -1000 to 1000 and we proved that it can be scaled to any desired large region. It should be noted that in case of arbitrary scales for each point such as A_1x, A_2y, and A_3z, we can use $A = \max(A_1, A_2, A_3)$ and it covers different scales on each dimension.

REFERENCES

[1] B. Nguyen and B. De Baets, "Kernel-based distance metric learning for supervised k-means clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 10, pp. 3084–3095, Oct. 2019.

[2] Y. K. Rupesh, P. Behnam, G. R. Pandla, M. Miryala, and M. N. Bojnordi, "Accelerating k-medians clustering using a novel 4T-4R RRAM cell," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 12, pp. 2709–2722, Dec. 2018.

[3] S. Zhang, Y. Wang, P. Wan, J. Zhuang, Y. Zhang, and Y. Li, "Clustering algorithm-based data fusion scheme for robust cooperative spectrum sensing," *IEEE Access*, vol. 8, pp. 5777–5786, 2020.

[4] A. Bryant and K. Cios, "RNN-DBSCAN: A density-based clustering algorithm using reverse nearest neighbor density estimates," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 6, pp. 1109–1121, Jun. 2018.

[5] A. Singh, A. Yadav, and A. Rana, "K-means with three different distance metrics," *Int. J. Comput. Appl.*, vol. 67, no. 10, pp. 13–17, Apr. 2013.

[6] H. Jia, Y.-M. Cheung, and J. Liu, "A new distance metric for unsupervised learning of categorical data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 5, pp. 1065–1079, May 2016.

[7] Y. Zhang, Y.-M. Cheung, and K. C. Tan, "A unified entropy-based distance metric for ordinal-and-nominal-attribute data clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 1, pp. 39–52, Jan. 2020.

[8] X. Feng, J. Zhang, C. Ren, and T. Guan, "An unequal clustering algorithm concerned with time-delay for Internet of Things," *IEEE Access*, vol. 6, pp. 33895–33909, 2018.

[9] G. Wang and L. Jia, "Short-term wind power forecasting based on BOMLS K-means similar hours clustering method," in *Proc. IEEE PES Asia–Pacific Power Energy Eng. Conf. (APPEEC)*, Dec. 2019, pp. 1–5.

[10] G. N. Lance and W. T. Williams, "Mixed-data classificatory programs I. Agglomerative systems," *Austral. Comput. J.*, vol. 1, no. 1, pp. 15–20, 1967.

[11] F. A. Sebayang, M. S. Lydia, and B. B. Nasution, "Optimization on purity K-means using variant distance measure," in *Proc. 3rd Int. Conf. Mech., Electron., Comput., Ind. Technol. (MECnIT)*, Jun. 2020, pp. 143–147.

[12] E. Xing, A. Y. Ng, M. Jordan, S. J. Russell, S. Oyama, and K. Tanaka, "Distance metric learning with application to clustering with side-information," in *Proc. 15th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2002, pp. 521–528.

[13] X. Jing, Z. Yan, Y. Shen, W. Pedrycz, and J. Yang, "A group-based distance learning method for semisupervised fuzzy clustering," *IEEE Trans. Cybern.*, vol. 52, no. 5, pp. 3083–3096, May 2022.

[14] J. Erman, M. Arlitt, and A. Mahanti, "Traffic classification using clustering algorithms," in *Proc. SIGCOMM Workshop Mining Netw. Data (MineNet)*, vol. 1, 2006, pp. 281–286.

[15] T. V. Le, R. Oentaryo, S. Liu, and H. C. Lau, "Local Gaussian processes for efficient fine-grained traffic speed prediction," *IEEE Trans. Big Data*, vol. 3, no. 2, pp. 194–207, Jun. 2017.

[16] M. Dong, Y. Wang, X. Yang, and J.-H. Xue, "Learning local metrics and influential regions for classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 6, pp. 1522–1529, Jun. 2020.

[17] H. A. A. Alfeilat, A. B. Hassanat, O. Lasassmeh, A. S. Tarawneh, M. B. Alhasanat, H. S. E. Salman, and V. S. Prasath, "Effects of distance measure choice on K-nearest neighbor classifier performance: A review," *Big Data*, vol. 7, no. 4, pp. 221–248, Dec. 2019.

[18] S. Chandra and A. K. Bharti, "Speed distribution curves for pedestrians during walking and crossing," *Proc. Social Behav. Sci.*, vol. 104, pp. 660–667, Dec. 2013.

[19] J. Dill and J. Gliebe, "Understanding and measuring bicycling behavior: A focus on travel time and route choice," Oregon Transp. Res. Educ. Consortium (OTREC), Urban Stud. Planning Fac. Publications Presentations, Portland, OR, USA, Final Rep. OTREC-RR-08-03, Dec. 2008. [Online]. Available: https://pdxscholar.library.pdx.edu/usp_fac/28/, doi: 10.15760/trec.151.

[20] S. Bernardi and F. Rupi, "An analysis of bicycle travel speed and disturbances on off-street and on-street facilities," *Transp. Res. Proc.*, vol. 5, pp. 82–94, Jan. 2015.

[21] M. Hou, K. Mahadevan, S. Somanath, E. Sharlin, and L. Oehlberg, "Autonomous vehicle-cyclist interaction: Peril and promise," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, Apr. 2020, pp. 1–12.

MOSTAFA RAEISI received the B.S. degree in electrical engineering from KIAU, Karaj, Iran, in 2009, and the M.S. degree in electrical engineering from the Iran University of Science and Technology (IUST), Tehran, Iran, in 2013. He is currently pursuing the Ph.D. degree in electrical and software engineering with the University of Calgary, Calgary, AB, Canada.

From 2009 to 2018, he was working in telecommunication and electrical engineering, Tehran. His research interests include the wireless communications, autonomous vehicles, machine learning, big data, data science, and cloud computing.

ABU B. SESAY (Life Senior Member, IEEE) received the Ph.D. degree in electrical engineering from McMaster University, Hamilton, ON, Canada, in 1988. From 1979 to 1984, he worked on various International Telecommunications Union projects. From 1986 to 1989, he was a Research Associate with McMaster University. In 1989, he joined the University of Calgary, Calgary, AB, Canada, where he is currently a Full Professor with the Department of Electrical and Computer Engineering. He was the Head of Department, from 2005 to 2011, and served as the Acting Associate Dean for Graduate Studies. From 1989 to 2005, he was a TRLaboratories Adjunct Scientist, where he conducted wireless research for various sponsors, including Nortel and Lucent. He spent sabbatical visits at Nortel Networks in Ottawa and Calgary. He has received numerous best paper awards with his students most notably the 1996 Neal Shepherd Memorial Best Propagation Paper Award for the paper "Effects of Antenna Height, Antenna Gain, and Pattern Downtilting for Cellular Mobile Radio," published in the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, Vol. 45, No. 2, May 1996. His current research interests include cooperative cellular wireless networks, orthogonal frequency-division multiple-access and code-division multiple-access systems, multiple-input-multiple-output systems, equalization, adaptive signal processing, and heterogeneous wireless network resource and mobility management, advanced signal processing (including machine learning) for unmanned aerial vehicles (UAV) using GNSS and 5G assisted autonomous vehicles, and relay networks for LTE and 5G.