Distance and Similarity Measures Effect on the Performance of K-Nearest Neighbor Classifier – A Review

V. B. Surya Prasath[a,*], Haneen Arafat Abu Alfeilat[b], Omar Lasassmeh[b], Ahmad B. A. Hassanat[b]

[a] *Department of Computer Science, University of Missouri-Columbia, MO 65211, USA*
[b] *Department of Information Technology, Mutah University, Karak, Jordan*

Abstract

The K-nearest neighbor (KNN) classifier is one of the simplest and most common classifiers, yet its performance competes with the most complex classifiers in the literature. The core of this classifier depends mainly on measuring the distance or similarity between the tested example and the training examples. This raises a major question about which distance measures to be used for the KNN classifier among a large number of distance and similarity measures? This review attempts to answer the previous question through evaluating the performance (measured by accuracy, precision and recall) of the KNN using a large number of distance measures, tested on a number of real world datasets, with and without adding different levels of noise. The experimental results show that the performance of KNN classifier depends significantly on the distance used, the results showed large gaps between the performances of different distances. We found that a recently proposed non-convex distance performed the best when applied on most datasets comparing to the other tested distances. In addition, the performance of the KNN degraded only about 20% while the noise level reaches 90%, this is true for all the distances used. This means that the KNN classifier using any of the top 10 distances tolerate noise to a certain degree. Moreover, the results show that some distances are less affected by the added noise comparing to other distances.

Keywords: K-nearest neighrbor, classification, distance, similarity, review

1. Introduction

Classification is an important problem in data science in general, and pattern recognition in particular. The K-nearest neighbor (KNN for short) is one

*Corresponding author. Tel.: +1 573 882 8391
Email address: prasaths@missouri.edu (V. B. Surya Prasath)

of the oldest, simplest and accurate algorithms for patterns classification and regression models. KNN was proposed in 1951 by Fix & Hodges (1951), and then modified by Cover & Hart (1967). KNN has been identified as one of the top ten methods in data mining (Wu et al., 2008). Consequently, KNN has been studied over the past few decades and widely applied in many fields (Bhatia & Vandana, 2010). Thus, KNN comprises the baseline classifier in many pattern classification problems such as pattern recognition (Xu & Wu, 2008), text categorization (Manne, Kotha, & Fatima, 2012), ranking models (Xiubo et al., 2008), object recognition (Bajramovic et al., 2006), and event recognition (Yang et al., 2000) applications. KNN is a non-parametric algorithm Kataria & Singh (2013). Non-Parametric means either there are no parameters or fixed number of parameters irrespective of size of data. Instead, parameters would be determined by the size of the training dataset. While there are no assumptions that need to be made to the underlying data distribution. Thus, KNN could be the best choice for any classification study that involves a little or no prior knowledge about the distribution of the data. In addition, KNN is one of the lazy learning methods. This implies storing all training data and waits until having the test data produced, without having to create a learning model Wettschereck, Aha, & John (1997).

1.1. Related works

Several studies have been conducted to analyze the performance of KNN classifier using different distance measures. Each study was applied on various kinds of datasets with different distributions, types of data and using different numbers of distance and similarity measures.

Chomboon and co-workers (Chomboon et al., 2015) analyzed the performance of KNN classifier using 11 distance measures. These include Euclidean, Mahalanobis, Manhattan, Minkowski, Chebychev, Cosine, Correlation, Hamming, Jaccard, Standardized Euclidean and Spearman distances. Their experiment had been applied on eight binary synthetic datasets with various kinds of distributions that were generated using MATLAB. They divided each dataset into 70% for training set and 30% for the testing set. The results showed that the Manhattan, Minkowski, Chebychev, Euclidean, Mahalanobis, and Standardized Euclidean distance measures achieved similar accuracy results and outperformed other tested distances.

Punam & Nitin (2015) evaluated the performance of KNN classifier using Chebychev, Euclidean, Manhattan, distance measures on KDD dataset (Tavallaee et al., 2009). The KDD dataset contains 41 features and two classes which type of data is numeric. The dataset was normalized before conducting the experiment. To evaluate the performance of KNN, accuracy, sensitivity and specificity measures were calculated for each distance. The reported results indicate that the use of Manhattan distance outperform the other tested distances, with 97.8% accuracy rate, 96.76% sensitivity rate and 98.35% Specificity rate.

Hu et al. (2016) analyzed the effect of distance measures on KNN classifier for medical domain datasets. Their experiments were based on three different types of medical datasets containing categorical, numerical, and mixed types of

data, which were chosen from the UCI machine learning repository, and four distance metrics including Euclidean, Cosine, Chi square, and Minkowsky distances. They divided each dataset into 90% of data as training and 10% as testing set, with k values from ranging from 1 to 15. The experimental results showed that Chi square distance function was the best choice for the three different types of datasets. However, using the Cosine, Euclidean and Minkowsky distance metrics performed the 'worst' over the mixed type of datasets. The 'worst' performance means the method with the lowest accuracy.

Todeschini, Ballabio, & Consonni (2015); Todeschini et al. (2016) analyzed the effect of eighteen different distance measures on the performance of KNN classifier using eight benchmark datasets. The investigated distance measures included Manhattan, Euclidean, Soergel, Lance–Williams, contracted Jaccard–Tanimoto, Jaccard–Tanimoto, Bhattacharyya, Lagrange, Mahalanobis, Canberra, Wave-Edge, Clark, Cosine, Correlation and four Locally centered Mahalanobis distances. For evaluating the performance of these distances, the non-error rate and average rank were calculated for each distance. The result indicated that the 'best' performance were the Manhattan, Euclidean, Soergel, Contracted Jaccard–Tanimoto and Lance–Williams distance measures. The 'best' performance means the method with the highest accuracy.

Lopes & Ribeiro (2015) analyzed the impact of five distance metrics, namely Euclidean, Manhattan, Canberra, Chebychev and Minkowsky in instance-based learning algorithms. Particularly, 1-NN Classifier and the Incremental Hypersphere Classifier (IHC) Classifier, they reported the results of their empirical evaluation on fifteen datasets with different sizes showing that the Euclidean and Manhattan metrics significantly yield good results comparing to the other tested distances.

Alkasassbeh, Altarawneh, & Hassanat (2015) investigated the effect of Euclidean, Manhattan and Hassanat (Hassanat, 2014) distance metrics on the performance of the KNN classifier, with K ranging from 1 to the square root of the size of the training set, considering only the odd K's. In addition to experimenting on other classifiers such as the Ensemble Nearest Neighbor classifier (ENN) (Hassanat, 2014), and the Inverted Indexes of Neighbors Classifier (IINC) (Jirina & Jirina, 2010). Their experiments were conducted on 28 datasets taken from the UCI machine learning repository, the reported results show that Hassanat distance outperformed both of Manhattan and Euclidean distances in most of the tested datasets using the three investigated classifiers.

Lindi (2016) investigated three distance metrics to use the best performer among them with the KNN classifier, which was employed as a matcher for their face recognition system that was proposed for the NAO robot. The tested distances were Chi-square, Euclidean and Hassanat distances. Their experiments showed that Hassanat distance outperformed the other two distances in terms of precision, but was slower than both of the other distances.

Table 1 provides a summary of these previous works on evaluating various distances within KNN classifier, along with the best distance assesed by each of them. As can be seen from the above literature review of most related works, that all of the previous works have investigated either a small number of distance

Table 1: Comparison between previous studies for distance measures in KNN classifier along with 'best' performing distance. Comparatively our current work compares the highest number of distance measures on variety of datasets.

Reference	#distances	#datasets	Best distance
Chomboon et al. (2015)	11	8	Manhattan, Minkowski Chebychev Euclidean, Mahalanobis Standardized Euclidean
Punam & Nitin (2015)	3	1	Manhattan
Hu et al. (2016)	4	37	Chi square
Todeschini, Ballabio, & Consonni (2015)	18	8	Manhattan, Euclidean, Soergel Contracted Jaccard–Tanimoto Lance–Williams
Lopes & Ribeiro (2015)	5	15	Euclidean and Manhattan
Alkasassbeh, Altarawneh, & Hassanat (2015)	3	28	Hassanat
Lindi (2016)	3	2	Hassanat
Ours	54	28	Hassanat

and similarity measures (ranging from 3 to 18 distances), a small number of datasets, or both.

1.2. Contributions

In KNN classifier, the distances between the test sample and the training data samples are identified by different measures tools. Therefore, distance measures play a vital role in determining the final classification output (Hu et al., 2016). Euclidean distance is the most widely used distance metric in KNN classifications, however, only few studies examined the effect of different distance metrics on the performance of KNN, these used a small number of distances, a small number of datasets, or both. Such shortage in experiments does not prove which distance is the best to be used with the KNN classifier. Therefore, this review attempts to bridge this gap by testing a large number of distance metrics on a large number of different datasets, in addition to investigate the distance metrics that least affected by added noise.

The KNN classifier can deal with noisy data, therefore, we need to investigate the impact of choosing different distance measures on the KNN performance when classifying a large number of real datasets, in addition to investigate which distance has the lowest noise implications. There are two main research questions addressed in this review:

1. What is be the best distance metric to be implemented with the KNN classifier?
2. What is the best distance metric to be implemented with the KNN classifier in the case of noise existence?

We mean by the 'best distance metric' (in this review) is the one that allows the KNN to classify test examples with the highest precision, recall and accuracy, i.e. the one that gives best performance of the KNN in terms of accuracy.

1.3. Organization

We organized our review as follows. First in Section 2 we provide an introductory overview to KNN classification method and present its history, characteristics, advantages and disadvantages. We review the definitions of various distance measures used in conjunction with KNN. Section 3 explains the datasets that were used in classification experiments, the structure of the experiments model, and the performance evaluations measures. We present and discuss the results produced by the experimental framework. Finally, Section 4 we provide the conclusions and possible future directions.

2. KNN and distance measures

2.1. Brief overview of KNN claasifier

The KNN algorithm classifies an unlabelled test sample based on the majority of similar samples among the k-nearest neighbors that are the closest to test sample. The distances between the test sample and each of the training

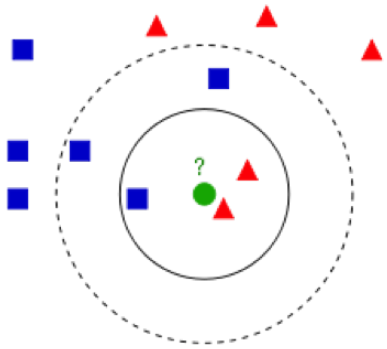

Figure 1: An example of KNN classification with k neighbors $k = 3$ (solid line circle) and $k = 5$ (dashed line circle), distance measure is Euclidean distance.

data samples is determined by a specific distance measure. Figure 1 shows a KNN example, contains training samples with two classes, the first class is 'blue square' and the second class is 'red triangle'. The test sample is represented in green circle. These samples are placed into two dimensional feature spaces with one dimension for each feature. To classify the test sample that belongs to class 'blue square' or to class 'red triangle'; KNN adopts a distance function to find the K nearest neighbors to the test sample. Finding the majority of classes among the k nearest neighbors predicts the class of the test sample. In this case, when $k = 3$ the test sample is classified to the first class 'red triangle' because there are two red triangles and only one blue square inside the inner circle, but when $k = 5$ it is classified to the "blue square" class because there are 2 red triangles and only 3 blue squares.

KNN is simple, but proved to be highly efficient and effective algorithm for solving various real life classification problems. However, KNN has got some disadvantages these include:

1. How to find the optimum K value in KNN Algorithm?
2. High computational time cost as we need to compute the distance between each test sample to all training samples, for each test example we need $O(nm)$ time complexity (number of operations), where n is the number of examples in the training data, and m is the number of features for each example.
3. High Memory requirement as we need to store all training samples O(nm) space complexity.
4. Finally, we need to determine the distance function that is the core of this study.

The first problem was solved either by using all the examples and taking the inverted indexes (Jirina & Jirina, 2008), or using ensemble learning (Hassanat et al., 2014). For the second and third problems, many studies have proposed different solutions depending on reducing the size of the training dataset, those include and not limited to (Hamming, 1968; Gates, 1972; Alpaydin, 1997; Kubat

& , 2000) and Wilson & Martinez (2000), or using approximate KNN classification such as (Arya & Mount, 1993) and (Zheng et al., 2016). Although some previous studies in the literature that investigated the fourth problem (see Section 1.1), here we attempt to investigate the fourth problem on a much larger scale, i.e., investigating a large number of distance metrics tested on a large set of problems. In addition, we investigate the effect of noise on choosing the most suitable distance metric to be used by the KNN classifier.

Algorithm 1 Basic KNN algorithm

Input: Training samples D, Test sample d, K
Output: Class label of test sample

1: Compute the distance between d and every sample in D
2: Choose the K samples in D that are nearest to d; denote the set by P ($\in$D)
3: Assign d the class it that is the most frequent class (or the majority class)

The basic KNN classifier steps can be described as follows:

1. Training phase: The training samples and the class labels of these samples are stored. no missing data allowed, no non-numeric data allowed.
2. Classification phase: Each test sample is classified using majority vote of its neighbors by the following steps:
   a) Distances from the test sample to all stored training sample are calculated using a specific distance function or similarity measure.
   b) The K nearest neighbors of the test sample are selected, where K is a pre-defined small integer.
   c) The most repeated class of these K neighbors is assigned to the test sample. In other words, a test sample is assigned to the class c if it is the most frequent class label among the K nearest training samples. If K = 1, then the class of the nearest neighbor is assigned to the test sample. KNN algorithm is described by Algorithm 1.

We provide a toy example to illustrate how to compute the KNN classifier. Assuming that we have three training examples, having three attributes for each, and one test example as shown in Table 2.

Table 2: Training and testing data examples.

	X1	X2	X3	class
Training sample (1)	5	4	3	1
Training sample (2)	1	2	2	2
Training sample (3)	1	2	3	2
Test sample	4	4	2	?

Step1: Determine the parameter K = number of the nearest neighbors to be considered. for this example we assume K = 1.

Step 2: Calculate the distance between test sample and all training samples using a specific similarity measure, in this example, Euclidean distance is used, see Table 3.

Table 3: Training and testing data examples with distances.

	X1	X2	X3	class	Distance
Training sample (1)	5	4	3	1	$D = \sqrt{(4-5)^2 + (4-4)^2 + (2-3)^2} = 1.4$
Training sample (2)	1	2	2	2	$D = \sqrt{(4-1)^2 + (4-2)^2 + (2-2)^2} = 3.6$
Training sample (3)	1	2	3	2	$D = \sqrt{(4-1)^2 + (4-2)^2 + (2-3)^2} = 3.7$
Test sample	4	4	2	?	

Step 3: Sort all examples based on their similarity or distance to the tested example, and then keep only the K similar (nearest) examples as shown in Table 4:

Table 4: Training and testing data examples with distances.

	X1	X2	X3	class	Distance	Rank minimum distance
Training sample (1)	5	4	3	1	1.4	1
Training sample (2)	1	2	2	2	3.6	2
Training sample (3)	1	2	3	2	3.7	3
Test sample	4	4	2	?		

Step 4: Based on the minimum distance, the class of the test sample is assigned to be 1. However, if $K = 3$ for instance, the class will be 2.

2.2. Noisy data

The existence of noise in data is mainly related to the way that has been applied to acquire and preprocess data from its environment (Nettleton, Orriols-Puig, & Fornells, 2010). Noisy data is a corrupted form of data in some way, which leads to partial alteration of the data values. Two main sources of noise can be identified: First, the implicit errors caused by measurement tools, such as using different types of sensors. Second, the random errors caused by batch processes or experts while collecting data, for example, errors during the process document digitization. Based on these two sources of errors, two types of noise can be classified in a given dataset (Zhu & Wu, 2004):

1. Class noise: occurs when the sample is incorrectly labeled due to several causes such as data entry errors during labeling process, or the inadequacy of information that is being used to label each sample.
2. Attribute noise: refers to the corruptions in values of one or more attributes due to several causes, such as failures in sensor devices, irregularities in sampling or transcription errors (Garcia, Luengo, & Herrera, 2014).

The generation of noise can be classified by three main characteristics (Saez et al., 2013):

1. The place where the noise is introduced: Noise may affect the attributes, class, training data, and test data separately or in combination.

2. The noise distribution: The way in which the noise is introduced, for example, uniform or Gaussian.
3. The magnitude of generated noise values: The extent to which the noise affects the data can be relative to each data value of each attribute, or relative to the standard deviation, minimum, maximum for each attribute.

In this work, we will add different noise levels to the tested datasets, to find the optimal distance metric that is least affected by this added noise with respect to the KNN classifier performance.

2.3. Distance measures review

The first appearance of the word distance can be found in the writings of Aristoteles (384 AC - 322 AC), who argued that the word distance means: "It is between extremities that distance is greatest" or "things which have something between them, that is, a certain distance". In addition, "distance has the sense of dimension [as in space has three dimensions, length, breadth and depth]". Euclid, one of the most important mathematicians of the ancient history, used the word distance only in his third postulate of the Principia (Euclid, 1956): "Every circle can be described by a centre and a distance". The distance is a numerical description of how far apart entities are. In data mining, the distance means a concrete way of describing what it means for elements of some space to be close to or far away from each other. Synonyms for distance include farness, dissimilarity, diversity, and synonyms for similarity include proximity (Cha, 2007), nearness (Todeschini, Ballabio, & Consonni, 2015).

The distance function between two vectors x and y is a function $d(x, y)$ that defines the distance between both vectors as a non-negative real number. This function is considered as a metric if satisfy a certain number of properties (Deza & Deza, 2009) that include the following:

1. **Non-negativity**: The distance between x and y is always a value greater than or equal to zero.
$$d(x, y) \geq 0$$
2. **Identity of indiscernibles**: The distance between x and y is equal to zero if and only if x is equal to y.
$$d(x, y) = 0 \quad \text{iff} \quad x = y$$
3. **Symmetry**: The distance between x and y is equal to the distance between y and x.
$$d(x, y) == d(y, x)$$
4. **Triangle inequality**: Considering the presence of a third point z, the distance between x and y is always less than or equal to the sum of the distance between x and z and the distance between y and z.
$$d(x, y) \leq d(x, z) + d(z, y)$$

When the distance is in the range $[0, 1]$, the calculation of a corresponding similarity measure $s(x, y)$ is as follows:

$$s(x, y) = 1 - d(x, y)$$

We consider the eight major distance families which consist of fifty four total distance measures. We categorized these distance measures following a similar categorization done by Cha (2007). In what follows, we give the mathematical definitions of distances to measure the closeness between two vectors x and y, where $x = (x_1, x_2, ., x_n)$ and $y = (y_1, y_2, ., y_n)$ having numeric attributes. As an example, we show the computed distance value between the vectors $v1 = \{5.1, 3.5, 1.4, 0.3\}$, $v2 = \{5.4, 3.4, 1.7, 0.2\}$. Theoretical analysis of these different distance metrics is beyond the scope of this work.

1. L_p **Minkowski distance measures**: This family of distances includes three distance metrics that are special cases of Minkowski distance, corresponding to different values of p for this power distance. The Minkowski distance, which is also known as L_p norm, is a generalized metric. It is defined as:

$$D_{Mink}(x, y) = \sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p},$$

where p is a positive value. When $p = 2$, the distance becomes the Euclidean distance. When $p = 1$ it becomes Manhattan distance. Chebyshev distance is a variant of Minkowski distance where $p = \infty$. x_i is the i^{th} value in the vector x and y_i is the i^{th} value in the vector y.

   1.1 Manhattan (MD): The Manhattan distance, also known as L_1 norm, Taxicab norm, Rectilinear distance or City block distance, which considered by Hermann Minkowski in 19th-century Germany. This distance represents the sum of the absolute differences between the opposite values in vectors.

$$MD(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

   1.2 Chebyshev (CD): Chebyshev distance is also known as maximum value distance (Grabusts, 2011), Lagrange (Todeschini, Ballabio, & Consonni, 2015) and chessboard distance (Premaratne, 2014). This distance is appropriate in cases when two objects are to be defined as different if they are different in any one dimension (Verma, 2012). It is a metric defined on a vector space where distance between two vectors is the greatest of their difference along any coordinate dimension.

$$CD(x, y) = \max_i |x_i - y_i|$$

   1.3 Euclidean (ED): Also known as L_2 norm or Ruler distance, which is an extension to the Pythagorean Theorem. This distance represents

the root of the sum of the square of differences between the opposite values in vectors.

$$ED(x,y) = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}$$

L_p Minkowski distance measures

Abbrev.	Name	Definition	Result
MD	Manhattan	$\sum_{i=1}^{n} \lvert x_i - y_i \rvert$	0.8
CD	Chebyshev	$\max_i \lvert x_i - y_i \rvert$	0.3
ED	Euclidean	$\sqrt{\sum_{i=1}^{n} \lvert x_i - y_i \rvert^2}$	0.4472

2. L_1 **Distance measures**: This distance family depends mainly finding the absolute difference, the family include Lorentzian, Canberra, Sorensen, Soergel, Kulczynski, Mean Character, Non Intersection distances.

   2.1 Lorentzian distance (LD): Lorentzian distance is represented by the natural log of the absolute difference between two vectors. This distance is sensitive to small changes since the log scale expands the lower range and compresses the higher range.

$$LD(x,y) = \sum_{i=1}^{n} ln(1 + |x_i - y_i|),$$

   where ln is the natural logarithm, and To ensure that the non-negativity property and to avoid log of zero, one is added.

   2.2 Canberra distance (CanD): Canberra distance, which is introduced by Williams & Lance (1966) and modified in Lance & Williams (1967). It is a weighted version of Manhattan distance, where the absolute difference between the attribute values of the vectors x and y is divided by the sum of the absolute attribute values prior to summing (Akila & Chandra, 2013). This distance is mainly used for positive values. It is very sensitive to small changes near zero, where it is more sensitive to proportional than to absolute differences. Therefore, this characteristic becomes more apparent in higher dimensional space, respectively with an increasing number of variables. The Canberra distance is often used for data scattered around an origin.

$$CanD(x,y) = \sum_{i=1}^{n} \frac{|x_i - y_i|}{|x_i| + |y_i|}$$

   2.3 Sorensen distance (SD): The Sorensen distance (Sorensen, 1948), also known as Bray–Curtis is one of the most commonly applied measurements to express relationships in ecology, environmental sciences and related fields. It is a modified Manhattan metric, where the summed differences between the attributes values of the vectors x and y are standardized by their summed attributes values (Szmidt,

2013). When all the vectors values are positive, this measure take value between zero and one.

$$SD(x,y) = \frac{\sum_{i=1}^{n} |x_i - y_i|}{\sum_{i=1}^{n} (x_i + y_i)}$$

2.4 Soergel distance (SoD): Soergel distance is one of the distance measures that is widely used to calculate the evolutionary distance (Chetty, Ngom, & Ahmad, 2008). It is also known as Ruzicka distance. For binary variables only, this distance is identical to the complement of the Tanimoto (or Jaccard) similarity coefficient (Zhou, Chan, & Wang, 2008). This distance obeys all four metric properties provided by all attributes have nonnegative values (Willett, Barnard, & Downs, 1998).

$$SoD(x,y) = \frac{\sum_{i=1}^{n} |x_i - y_i|}{\sum_{i=1}^{n} \max(x_i, y_i)}$$

2.5 Kulczynski Distance (KD): Similar to the Soergel distance, but instead of using the maximum, it uses the minimum function.

$$KD(x,y) = \frac{\sum_{i=1}^{n} |x_i - y_i|}{\sum_{i=1}^{n} \min(x_i, y_i)}$$

2.6 Mean Character Distance (MCD): Also known as Average Manhattan, or Gower distance.

$$MCD(x,y) = \frac{\sum_{i=1}^{n} |x_i - y_i|}{n}$$

2.7 Non Intersection Distance (NID): Non Intersection distance is the complement to the intersection similarity and is obtained by subtracting the intersection similarity from one.

$$NID(x,y) = \frac{1}{2} \sum_{i=1}^{n} |x_i - y_i|.$$

L_1 Distance measures

Abbrev.	Name	Result	
LD	Lorentzian	$\sum_{i=1}^{n} ln(1 + \lvert x_i - y_i \rvert)$	0.7153
CanD	Canberra	$\sum_{i=1}^{n} \frac{\lvert x_i - y_i \rvert}{\lvert x_i \rvert + \lvert y_i \rvert}$	0.0381
SD	Sorensen	$\frac{\sum_{i=1}^{n} \lvert x_i - y_i \rvert}{\sum_{i=1}^{n} (x_i + y_i)}$	0.0381
SoD	Soergel	$\frac{\sum_{i=1}^{n} \lvert x_i - y_i \rvert}{\sum_{i=1}^{n} \max(x_i, y_i)}$	0.0734
KD	Kulczynski	$\frac{\sum_{i=1}^{n} \lvert x_i - y_i \rvert}{\sum_{i=1}^{n} \min(x_i, y_i)}$	0.0792
MCD	Mean Character	$\frac{\sum_{i=1}^{n} \lvert x_i - y_i \rvert}{n}$	0.2
NID	Non Intersection	$\frac{1}{2} \sum_{i=1}^{n} \lvert x_i - y_i \rvert$	0.4

3. **Inner product distance measures**: Distance measures belonging to this family are calculated by some products of pair wise values from both vectors, this type of distances includes: Jaccard, Cosine, Dice, Chord distances.

   3.1 Jaccard distance (JacD): The Jaccard distance measures dissimilarity between sample sets, it is a complementary to the Jaccard similarity coefficient (Jaccard, 1901) and is obtained by subtracting the Jaccard coefficient from one. This distance is a metric (Cesare & Xiang, 2012).

$$JacD(x,y) = \frac{\sum_{i=1}^{n}(x_i - y_i)^2}{\sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} y_i^2 - \sum_{i=1}^{n} x_i y_i}$$

   3.2 Cosine distance (CosD): The Cosine distance, also called angular distance, is derived from the cosine similarity that measures the angle between two vectors, where Cosine distance is obtained by subtracting the cosine similarity from one.

$$CosD(x,y) = 1 - \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

   3.3 Dice distance (DicD): The dice distance is derived from the dice similarity (Dice, 1945), which is a complementary to the dice similarity and is obtained by subtracting the dice similarity from one. It can be sensitive to values near zero. This distance is a not a metric, in particular, the property of triangle inequality does not hold. This distance is widely used in information retrieval in documents and biological taxonomy.

$$DicD(x,y) = 1 - \frac{2\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} y_i^2}$$

   3.4 Chord distance (ChoD): A modification of Euclidean distance (Gan, Ma, & Wu, 2007), which was introduced by Orloci (Orloci, 1967) to be used in analyzing community composition data (Legendre & Legendre, 2012). It was defined as the length of the chord joining two normalized points within a hypersphere of radius one. This distance is one of the distance measures that is commonly used for clustering continuous data (Shirkhorshidi, Aghabozorgi, & Wah, 2015).

$$ChoD(x,y) = \sqrt{2 - 2\frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i^2}}$$

Inner product distance measures family

Abbrev.	Name	Result	
JacD	Jaccard	$\frac{\sum_{i=1}^{n}(x_i-y_i)^2}{\sum_{i=1}^{n}x_i^2+\sum_{i=1}^{n}y_i^2-\sum_{i=1}^{n}x_iy_i}$	0.0048
CosD	Cosine	$1-\frac{\sum_{i=1}^{n}x_iy_i}{\sqrt{\sum_{i=1}^{n}x_i^2}\sqrt{\sum_{i=1}^{n}y_i^2}}$	0.0016
DicD	Dice	$1-\frac{2\sum_{i=1}^{n}x_iy_i}{\sum_{i=1}^{n}x_i^2+\sum_{i=1}^{n}y_i^2}$	0.9524
ChoD	Chord	$\sqrt{2-2\frac{\sum_{i=1}^{n}x_iy_i}{\sum_{i=1}^{n}x_i^2\sum_{i=1}^{n}y_i^2}}$	0.0564

4. **Squared Chord distance measures**: Distances that belong to this family are obtained by calculating the sum of geometrics. The geometric mean of two values is the square root of their product. The distances in this family cannot be used with features vector of negative values, this family includes Bhattachayya, Squared Chord, Matusita, Hellinger distances.

   4.1 Bhattacharyya distance (BD): The Bhattacharyya distance measures the similarity of two probability distributions (Bhattachayya, 1943).

$$BD(x,y) = -ln\sum_{i=1}^{n}\sqrt{x_iy_i}$$

   4.2 Squared chord distance (SCD): Squared chord distance is mostly used with paleontologists and in studies on pollen. In this distance, the sum of square of square root difference at each point is taken along both vectors, which increases the difference for more dissimilar feature.

$$SCD(x,y) = \sum_{i=1}^{n}(\sqrt{x_i}-\sqrt{y_i})^2$$

   4.3 Matusita distance (MatD): Matusita distance is the square root of the squared chord distance.

$$MatD(x,y) = \sqrt{\sum_{i=1}^{n}(\sqrt{x_i}-\sqrt{y_i})^2}$$

   4.4 Hellinger distance (HeD): Hellinger distance also called Jeffries - Matusita distance (Abbad & Tairi, 2016) was introduced in 1909 by Hellinger (Hellinger, 1909), it is a metric used to measure the similarity between two probability distributions. This distance is closely related to Bhattacharyya distance.

$$HeD(x,y) = \sqrt{2\sum_{i=1}^{n}(\sqrt{x_i}-\sqrt{y_i})^2}$$

Squared Chord distance measures family

Abbrev.	Name	Result	
BD	Bhattacharyya	$-ln \sum_{i=1}^{n} \sqrt{x_i y_i}$	-2.34996
SCD	Squared Chord	$\sum_{i=1}^{n} (\sqrt{x_i} - \sqrt{y_i})^2$	0.0297
MatD	Matusita	$\sqrt{\sum_{i=1}^{n} (\sqrt{x_i} - \sqrt{y_i})^2}$	0.1722
HeD	Hellinger	$\sqrt{2 \sum_{i=1}^{n} (\sqrt{x_i} - \sqrt{y_i})^2}$	0.2436

5. **Squared L_2 distance measures**: In L_2 distance measure family, the square of difference at each point a long both vectors is considered for the total distance, this family includes Squared Euclidean, Clark, Neyman χ^2, Pearson χ^2, Squared χ^2, Probabilistic Symmetric χ^2, Divergence, Additive Symmetric χ^2, Average, Mean Censored Euclidean and Squared Chi-Squared distances.

5.1 Squared Euclidean distance (SED): Squared Euclidean distance is the sum of the squared differences without taking the square root.

$$SED(x, y) = \sum_{i=1}^{n} (x_i - y_i)^2$$

5.2 Clark distance (ClaD): The Clark distance also called coefficient of divergence was introduced by Clark (Clark, 2014). It is the squared root of half of the divergence distance.

$$ClaD(x, y) = \sqrt{\sum_{i=1}^{n} \left(\frac{|x_i - y_i|}{x_i + y_i} \right)^2}$$

5.3 Neyman χ^2 distance (NCSD): The Neyman χ^2 (Neyman & John, 1949) is called a quasi-distance.

$$NCSD(x, y) = \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{x_i}$$

5.4 Pearson χ^2 distance (PCSD): Pearson χ^2 distance (Pearson, 1900), also called χ^2 distance.

$$PCSD(x, y) = \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{y_i}$$

5.5 Squared χ^2 distance (SquD): Also called triangular discrimination distance. This distance is a quasi-distance.

$$SquD(x, y) = \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{x_i + y_i}$$

5.6 Probabilistic Symmetric χ^2 distance (PSCSD): This distance is equivalent to Sangvi χ^2 distance.

$$PSCSD(x, y) = 2 \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{x_i + y_i}$$

5.7 Divergence distance (DivD):

$$DivD(x,y) = 2\sum_{i=1}^{n}\frac{(x_i - y_i)^2}{(x_i + y_i)^2}$$

5.8 Additive Symmetric χ^2 (ASCSD): Also known as symmetric χ^2 divergence.

$$ASCSD(x,y) = 2\sum_{i=1}^{n}\frac{(x_i - y_i)^2(x_i + y_i)}{x_i y_i}$$

5.9 Average distance (AD): The average distance, also known as average Euclidean is a modified version of the Euclidean distance (Shirkhorshidi, Aghabozorgi, & Wah, 2015). Where the Euclidean distance has the following drawback, "if two data vectors have no attribute values in common, they may have a smaller distance than the other pair of data vectors containing the same attribute values" (Gan, Ma, & Wu, 2007), so that, this distance was adopted.

$$AD(x,y) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2}$$

5.10 Mean Censored Euclidean Distance (MCED): In this distance, the sum of squared differences between values is calculated and, to get the mean value, the summed value is divided by the total number of values where the pairs values do not equal to zero. After that, the square root of the mean should be computed to get the final distance.

$$MCED(x,y) = \sqrt{\frac{\sum_{i=1}^{n}(x_i - y_i)^2}{\sum_{i=1}^{n} 1_{x_i^2 + y_i^2 \neq 0}}}$$

5.11 Squared Chi-Squared (SCSD):

$$SCSD(x,y) = \sum_{i=1}^{n}\frac{(x_i - y_i)^2}{|x_i + y_i|}$$

Squared L_2 distance measures family

SED	Squared Euclidean	$\sum_{i=1}^{n}(x_i - y_i)^2$	0.2
ClaD	Clark	$\sqrt{\sum_{i=1}^{n}\left(\frac{\lvert x_i - y_i\rvert}{x_i + y_i}\right)^2}$	0.2245
NCSD	Neyman χ^2	$\sum_{i=1}^{n}\frac{(x_i - y_i)^2}{x_i}$	0.1181
PCSD	Pearson χ^2	$\sum_{i=1}^{n}\frac{(x_i - y_i)^2}{y_i}$	0.1225
SquD	Squared χ^2	$\sum_{i=1}^{n}\frac{(x_i - y_i)^2}{x_i + y_i}$	0.0591
PSCSD	Probabilistic Symmetric χ^2	$2\sum_{i=1}^{n}\frac{(x_i - y_i)^2}{x_i + y_i}$	0.1182
DivD	Divergence	$2\sum_{i=1}^{n}\frac{(x_i - y_i)^2}{(x_i + y_i)^2}$	0.1008
ASCSD	Additive Symmetric χ^2	$2\sum_{i=1}^{n}\frac{(x_i - y_i)^2(x_i + y_i)}{x_i y_i}$	0.8054
AD	Average	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2}$	0.2236
MCED	Mean Censored Euclidean	$\sqrt{\frac{\sum_{i=1}^{n}(x_i - y_i)^2}{\sum_{i=1}^{n} 1_{x_i^2 + y_i^2 \neq 0}}}$	0.2236
SCSD	Squared Chi-Squared	$\sum_{i=1}^{n}\frac{(x_i - y_i)^2}{\lvert x_i + y_i\rvert}$	0.0591

6. **Shannon entropy distance measures**: The distance measures belonging to this family are related to the Shannon entropy (Shannon, 2001). These distances include Kullback-Leibler, Jeffreys, K divergence, Topsoe, Jensen-Shannon, Jensen difference distances.

   6.1 Kullback-Leibler distance (KLD): Kullback-Leibler distance was introduced by Kullback & Leibler (1951), it is also known as KL divergence, relative entropy, or information deviation, which measures the difference between two probability distributions. This distance is not a metric measure, because it is not symmetric. Furthermore, it does not satisfy triangular inequality property, therefore it is called quasi-distance. Kullback-Leibler divergence has been used in several natural language applications such as for query expansion, language models, and categorization (Pinto, Benedi, & Rosso, 2007).

$$KLD(x, y) = \sum_{i=1}^{n} x_i \, ln\frac{x_i}{y_i},$$

   where ln is the natural logarithm.

   6.2 Jeffreys Distance (JefD): Jeffreys distance (Jeffreys, 1946), also called J-divergence or KL2- distance, is a symmetric version of the Kullback-Leibler distance.

$$JefD(x, y) = \sum_{i=1}^{n}(x_i - y_i) \, ln\frac{x_i}{y_i}$$

   6.3 K divergence Distance (KDD):

$$KDD(x, y) = \sum_{i=1}^{n} x_i \, ln\frac{2x_i}{x_i + y_i}$$

6.4 Topsoe Distance (TopD): The Topsoe distance (Topsoe, 2000), also called information statistics, is a symmetric version of the Kullback-Leibler distance. The Topsoe distance is twice the Jensen-Shannon divergence. This distance is not a metric, but its square root is a metric.

$$TopD(x,y) = \sum_{i=1}^{n} x_i \, ln\left(\frac{2x_i}{x_i + y_i}\right) + \sum_{i=1}^{n} y_i \, ln\left(\frac{2y_i}{x_i + y_i}\right)$$

6.5 Jensen-Shannon Distance (JSD): Jensen-Shannon distance is the square root of the Jensen Shannon divergence. It is the half of the Topsoe distance which uses the average method to make the K divergence symmetric.

$$JSD(x,y) = \frac{1}{2}\left[\sum_{i=1}^{n} x_i \, ln\left(\frac{2x_i}{x_i + y_i}\right) + \sum_{i=1}^{n} y_i \, ln\left(\frac{2y_i}{x_i + y_i}\right)\right]$$

6.6 Jensen difference distance (JDD): Jensen difference distance was introduced by Sibson (1969).

$$JDD(x,y) = \frac{1}{2}\left[\sum_{i=1}^{n} \frac{x_i \, lnx_i + y_i lny_i}{2} - \left(\frac{x_i + y_i}{2}\right) \, ln\left(\frac{x_i + y_i}{2}\right)\right]$$

Shannon entropy distance measures family

Abbrev.	Name	Result	
KLD	Kullback-Leibler	$\sum_{i=1}^{n} x_i \, ln\frac{x_i}{y_i}$	-0.3402
JefD	Jeffreys	$\sum_{i=1}^{n} (x_i - y_i) \, ln\frac{x_i}{y_i}$	0.1184
KDD	K divergence	$\sum_{i=1}^{n} x_i \, ln\frac{2x_i}{x_i+y_i}$	-0.1853
TopD	Topsoe	$\sum_{i=1}^{n} x_i \, ln\left(\frac{2x_i}{x_i+y_i}\right) + \sum_{i=1}^{n} y_i \, ln\left(\frac{2y_i}{x_i+y_i}\right)$	0.0323
JSD	Jensen-Shannon	$\frac{1}{2}\left[\sum_{i=1}^{n} x_i \, ln\frac{2x_i}{x_i+y_i} + \sum_{i=1}^{n} y_i \, ln\frac{2y_i}{x_i+y_i}\right]$	0.014809
JDD	Jensen difference	$\frac{1}{2}\left[\sum_{i=1}^{n} \frac{x_i \, lnx_i + y_i lny_i}{2} - \left(\frac{x_i+y_i}{2}\right) \, ln\left(\frac{x_i+y_i}{2}\right)\right]$	0.0074

7. **Vicissitude distance measures**: Vicissitude distance family consists of four distances, Vicis-Wave Hedges, Vicis Symmetric, Max Symmetric χ^2 and Min Symmetric χ^2 distances. These distances were generated from syntactic relationship for the aforementioned distance measures.

7.1 Vicis-Wave Hedges distance (VWHD): The so-called "Wave-Hedges distance" has been applied to compressed image retrieval (Hatzigiorgaki & Skodras, 2003), content based video retrieval (Patel & Meshram, 2012), time series classification (Giusti & Batista, 2013),image fidelity (Macklem, 2002), finger print recognition (Bharkad & Kokare, 2011), etc.. Interestingly, the source of the "Wave-Hedges" metric has not been correctly cited, and some of the previously mentioned resources allude to it incorrectly as Hedges (1976). The source

of this metric eludes the authors, despite best efforts otherwise. Even the name of the distance "Wave-Hedges" is questioned (Hassanat, 2014).

$$VWHD(x,y) = \sum_{i=1}^{n} \frac{|x_i - y_i|}{\min(x_i, y_i)}$$

7.2 Vicis symmetric distance (VSD): Vicis Symmetric distance is defined by three formulas, VSDF1, VSDF2, VSDF3 as the following

$$VSDF1(x,y) = \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{\min(x_i, y_i)^2},$$

$$VSDF2(x,y) = \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{\min(x_i, y_i)},$$

$$VSDF3(x,y) = \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{\max(x_i, y_i)}$$

7.3 Max symmetric χ^2 distance (MSCD):

$$MSCD(x,y) = \max\left(\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{x_i}, \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{y_i}\right)$$

7.4 Min symmetric χ^2 distance (MiSCSD):

$$MiSCSD(x,y) = \min\left(\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{x_i}, \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{y_i}\right)$$

Vicissitude distance measures family

Abbrev.	Name	Result	
VWHD	Vicis-Wave Hedges	$\sum_{i=1}^{n} \frac{\lvert x_i - y_i \rvert}{\min(x_i, y_i)}$	0.8025
VSDF1	Vicis Symmetric1	$\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{\min(x_i, y_i)^2}$	0.3002
VSDF2	Vicis Symmetric2	$\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{\min(x_i, y_i)}$	0.1349
VSDF3	Vicis Symmetric3	$\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{\max(x_i, y_i)}$	0.1058
MSCD	Max Symmetric χ^2	$\max\left(\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{x_i}, \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{y_i}\right)$	0.1225
MiSCSD	Min Symmetric χ^2	$\min\left(\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{x_i}, \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{y_i}\right)$	0.1181

8. **Other distance measures**: These metrics exhibits distance measures utilizing multiple ideas or measures from previous distance measures, these include and not limited to Average (L_1,L_∞), Kumar-Johnson, Taneja, Pearson, Correlation, Squared Pearson, Hamming, Hausdorff, χ^2 statistic, Whittaker's index of association, Meehl, Motyka and Hassanat distances.

8.1 Average (L_1, L_∞) distance (AvgD): Average (L_1, L_∞) distance is the average of Manhattan and Chebyshev distances.

$$AvgD(x,y) = \frac{\sum_{i=1}^{n} |x_i - y_i| + \max_i |x_i - y_i|}{2}$$

8.2 Kumar- Johnson Distance (KJD):

$$KJD(x,y) = \sum_{i=1}^{n} \left(\frac{(x_i^2 + y_i^2)^2}{2(x_i y_i)^{3/2}} \right)$$

8.3 Taneja Distance (TanD): (Taneja, 1995)

$$TJD(x,y) = \sum_{i=1}^{n} \left(\frac{x_i + y_i}{2} \right) ln \left(\frac{x_i + y_i}{2\sqrt{x_i y_i}} \right)$$

8.4 Pearson Distance (PeaD): The Pearson distance is derived from the Pearsons correlation coefficient, which measures the linear relationship between two vectors (Fulekar, 2009). This distance is obtained by subtracting the Pearsons correlation coefficient from one.

$$PeaD(x,y) = 1 - \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2} \star \sum_{i=1}^{n}(y_i - \bar{y})^2}$$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$.

8.5 Correlation Distance (CorD): Correlation distance is a version of the Pearson distance, where the Pearson distance is scaled in order to obtain a distance measure in the range between zero and one.

$$CorD(x,y) = \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}\right)$$

8.6 Squared Pearson Distance (SPeaD):

$$SPeaD(x,y) = 1 - \left(\frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}\right)^2$$

8.7 Hamming Distance (HamD): Hamming distance (Hamming, 1958) is a distance metric that measures the number of mismatches between two vectors. It is mostly used for nominal data, string and bitwise analyses, and also can be useful for numerical data.

$$HamD(x,y) = \sum_{i=1}^{n} 1_{x_i \neq y_i}$$

8.8 Hausdorff Distance (HauD):

$$HauD(x,y) = \max(h(x,y), h(y,x))$$

where $h(x,y) = \max_{x_i \in x} \min_{y_i \in y} ||x_i - y_i||$, and $||\cdot||$ is the vector norm (e.g. L_2 norm). The function $h(x,y)$ is called the directed Hausdorff distance from x to y. The Hausdorff distance HauD(x, y) measures the degree of mismatch between the sets x and y by measuring the remoteness between each point x_i and y_i and vice versa.

8.9 χ^2 statistic Distance (CSSD): The χ^2 statistic distance was used for image retrieval (Kadir et al., 2012), histogram (Rubner & Tomasi, 2013), etc.

$$CSSD(x,y) = \sum_{i=1}^{n} \frac{x_i - m_i}{m_i}$$

where $m_i = \frac{x_i + y_i}{2}$.

8.10 Whittaker's index of association Distance (WIAD): Whittaker's index of association distance was designed for species abundance data (Whittaker, 1952).

$$WIAD(x,y) = \frac{1}{2} \sum_{i=1}^{n} \left| \frac{x_i}{\sum_{i=1}^{n} x_i} - \frac{y_i}{\sum_{i=1}^{n} y_i} \right|$$

8.11 Meehl Distance (MeeD): Meehl distance depends on one consecutive point in each vector.

$$MeeD(x,y) = \sum_{i=1}^{n-1} (x_i - y_i - x_{i+1} + y_{i+1})^2$$

8.12 Motyka Distance (MotD):

$$MotD(x,y) = \frac{\sum_{i=1}^{n} \max(x_i, y_i)}{\sum_{i=1}^{n} (x_i + y_i)}$$

8.13 Hassanat Distance (HasD): Hassanat Distance introduced by Hassanat (2014).

$$HasD(x,y) = \sum_{i=1}^{n} D(x_i, y_i)$$

where

$$D(x,y) = \begin{cases} 1 - \frac{1 + \min(x_i, y_i)}{\max(x_i, y_i)}, & \min(x_i, y_i) \geq 0 \\ 1 - \frac{1 + \min(x_i, y_i) + |\min(x_i, y_i)|}{\max(x_i, y_i) + |\min(x_i, y_i)|}, & \min(x_i, y_i) < 0 \end{cases}$$

As can be seen, Hassanat distance is bounded by $[0, 1[$. It reaches 1 when the maximum value approaches infinity assuming the minimum is finite, or when the minimum value approaches minus infinity

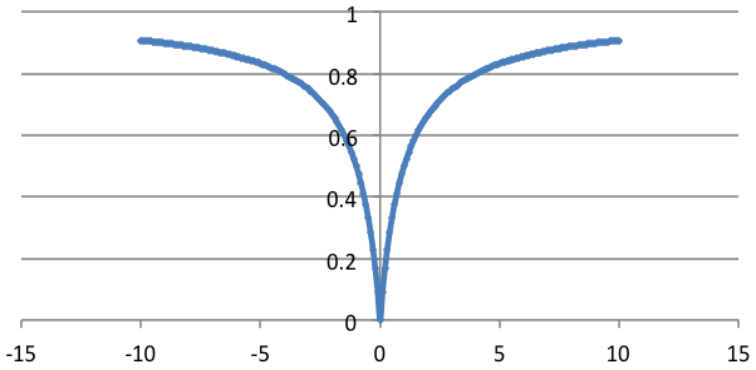

Figure 2: Representation of Hassanat distance between the point 0 and n, where n belongs to $[-10, 10]$.

assuming the maximum is finite. This is shown by Figure 2 and the following equations.

$$\lim_{\max(A_i,B_i)\to\infty} D(A_i, B_i) = \lim_{\max(A_i,B_i)\to-\infty} D(A_i, B_i) = 1,$$

By satisfying all the metric properties this distance was proved to be a metric by Hassanat (2014). In this metric no matter what the difference between two values is, the distance will be in the range of 0 to 1. so the maximum distance approaches to the dimension of the tested vectors, therefore the increases in dimensions increases the distance linearly in the worst case.

Other distance measures family

Abbrev.	Name	Result	
AvgD	Average (L_1, L_∞)	$\frac{\sum_{i=1}^{n} \lvert x_i-y_i\rvert+\max_i \lvert x_i-y_i\rvert}{2}$	0.55
KJD	Kumar-Johnson	$\sum_{i=1}^{n} \left(\frac{(x_i^2+y_i^2)^2}{2(x_i y_i)^{3/2}}\right)$	21.2138
TanD	Taneja	$\sum_{i=1}^{n} \left(\frac{x_i+y_i}{2}\right) ln\left(\frac{x_i+y_i}{2\sqrt{x_i y_i}}\right)$	0.0149
PeaD	Pearson	$1-\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2 \star \sum_{i=1}^{n}(y_i-\bar{y})^2}}$	
		$\bar{x}=\frac{1}{n}\sum_{i=1}^{n} x_i$	0.9684
CorD	Correlation	$\frac{1}{2}\left(1-\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}\right)$	0.4842
SPeaD	Squared Pearson	$1-\left(\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}\right)^2$	0.999
HamD	Hamming	$\sum_{i=1}^{n} 1_{x_i \neq y_i}$	4
HauD	Hausdorff	$\max(h(x,y),h(y,x))$	
		$h(x,y)=\max_{x_i \in x} \min_{y_i \in y} \lvert\lvert x_i-y_i\rvert\rvert$	0.3
CSSD	χ^2 statistic	$\sum_{i=1}^{n} \frac{x_i-m_i}{m_i},\ m_i=\frac{x_i+y_i}{2}$	0.0894
WIAD	Whittaker's index of assoc.	$\frac{1}{2}\sum_{i=1}^{n} \lvert\frac{x_i}{\sum_{i=1}^{n} x_i}-\frac{y_i}{\sum_{i=1}^{n} y_i}\rvert$	1.9377
MeeD	Meehl	$\sum_{i=1}^{n-1}(x_i-y_i-x_{i+1}+y_{i+1})^2$	0.48
MotD	Motyka	$\frac{\sum_{i=1}^{n}\max(x_i,y_i)}{\sum_{i=1}^{n}(x_i+y_i)}$	0.5190
HasD	Hassanat	$\sum_{i=1}^{n} D(x_i,y_i)$	
		$=\begin{cases}1-\frac{1+\min(x_i,y_i)}{\max(x_i,y_i)}, & \min(x_i,y_i)\geq 0\\ 1-\frac{1+\min(x_i,y_i)+\lvert\min(x_i,y_i)\rvert}{\max(x_i,y_i)+\lvert\min(x_i,y_i)\rvert}, & \min(x_i,y_i)<0\end{cases}$	0.2571

3. Experimental framework

3.1. Datasets used for experiments

The experiments were done on twenty eight datasets which represent real life classification problems, obtained from the UCI Machine Learning Repository (Lichman, 2013). The UCI Machine Learning Repository is a collection of databases, domain theories, and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms. The database was created in 1987 by David Aha and fellow graduate students at UC Irvine. Since that time, it has been widely used by students, educators, and researchers all over the world as a primary source of machine learning data sets.

Each dataset consist of a set of examples. Each example is defined by a number of attributes and all the examples inside the data are represented by the same number of attributes. One of these attributes is called the class attribute, which contains the class value (label) of the data, whose values are predicted for test the examples. Short description of all the datasets used is provided in Table 5.

Table 5: Description of the real world datasets used (from the UCI Machine Learning Repository), where #E means number of examples, and #F means number of features and #C means number of classes.

Name	#E	#F	#C	data type	Min	Max
Heart	270	25	2	real& integer	0	564
Balance	625	4	3	positive integer	1	5
Cancer	683	9	2	positive integer	0	9
German	1000	24	2	positive integer	0	184
Liver	345	6	2	real& integer	0	297
Vehicle	846	18	4	positive integer	0	1018
Vote	399	10	2	positive integer	0	2
BCW	699	10	2	positive integer	1	13454352
Haberman	306	3	2	positive integer	0	83
Letter rec.	20000	16	26	positive integer	0	15
Wholesale	440	7	2	positive integer	1	112151
Australian	690	42	2	positive real	0	100001
Glass	214	9	6	positive real	0	75.41
Sonar	208	60	2	positive real	0	1
Wine	178	13	3	positive real	0.13	1680
EEG	14980	14	2	positive real	86.67	715897
Parkinson	1040	27	2	positive real	0	1490
Iris	150	4	3	positive real	0.1	7.9
Diabetes	768	8	2	real& integer	0	846
Monkey1	556	17	2	binary	0	1
Ionosphere	351	34	2	real	-1	1
Phoneme	5404	5	2	real	-1.82	4.38
Segmen	2310	19	7	real	-50	1386.33
Vowel	528	10	11	real	-5.21	5.07
Wave21	5000	21	3	real	-4.2	9.06
Wave40	5000	40	3	real	-3.97	8.82
Banknote	1372	4	2	real	-13.77	17.93
QSAR	1055	41	2	real	-5.256	147

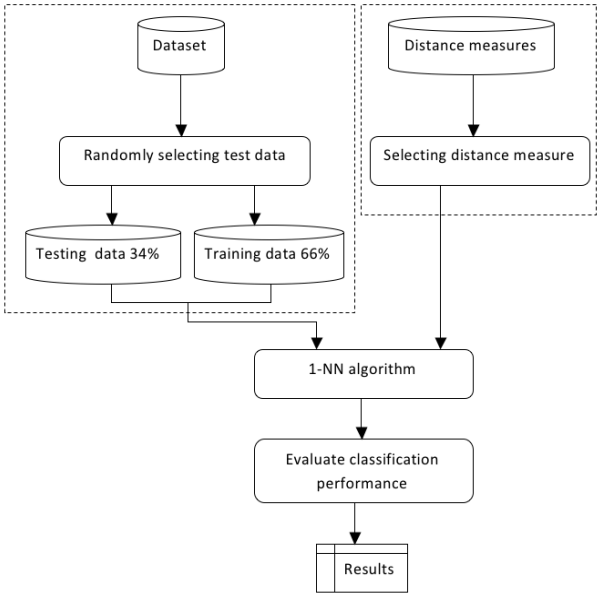

Figure 3: The framework of our experiments for discerning the effect of various distances on the performance of KNN classifier.

Algorithm 2 Create noisy dataset

Input: Original dataset D, level of noise $x\%$ [10%-90%]
Output: Noisy dataset

Number of noisy examples : $N = x\% *$ number of examples in D
Array NoisyExample $[N]$
for $K = 1$ to N **do**
    Randomly choose an example number as E from D
    if E **then** is chosen previously
        Go to Step 4
    else
        NoisyExample $[k] = E$
for each attribute A_i **do** do
    for each NoisyExample NE_j **do** do
        RV = Random value between Min(A_i) and Max(A_i)
        $NE_j A_j = RV$.

3.2. Experimental setup

Each dataset is divided into two data sets, one for training, and the other for testing. For this purpose, 34% of the data set is used for testing, and 66% of the data is dedicated for training. The value of K is set to 1 for simplicity. The 34% of the data, which were used as a test sample, were chosen randomly, and each experiment on each data set was repeated 10 times to obtain random examples for testing and training. The overall experimental framework is shown in Figure 3. Our experiments are divided into two major parts:

1. The first part of experiments aims to find the best distance measures to be used by KNN classifier without any noise in the datasets. We used all the 54 distances which were reviewed in Section 2.3.
2. The second part of experiments aims to find the best distance measure to be used by KNN classifier in the case of noisy data. In this work, we define the 'best' method as the method that performs with the highest accuracy. We added noise into each dataset at various levels of noise. The experiments in the second part were conducted using the top 10 distances, those which achieved the best results in the first part of experiments. Therefore, in order to create a noisy dataset from the original one, a level of noise $x\%$ is selected in the range of (10% to 90%), the level of noise means the number of examples that need to be noisy, the amount of noise is selected randomly between the minimum and maximum values of each attribute, all attributes for each examples are corrupted by a random noise, the number of noisy examples are selected randomly. Algorithm 2 describes the process of corrupting data with random noise to be used for further experiments for the purposes of this work.

3.3. Performance evaluation measures

Different measures are available for evaluating the performance of classifiers. In this study, three measures were used, accuracy, precision, and recall. Accuracy is calculated to evaluate the overall classifier performance. It is defined as the ratio of the test samples that are correctly classified to the number of tested examples,

$$Accuracy = \frac{\text{Number of correct classifications}}{\text{Total number of test samples}}. \tag{1}$$

In order to assess the performance with respect to every class in a dataset, We compute precision and recall measures. Precision (or positive predictive value) is the fraction of retrieved instances that are relevant, while recall (or sensitivity) is the fraction of relevant instances that are retrieved. These measures can be constructed by computing the following:

1. True positive (TP): The number of correctly classified examples of a specific class (as we calculate these measures for each class)
2. True negative (TN):The number of correctly classified examples that were not belonging to the specific class

3. False positive (FP):The number of examples that incorrectly assigned to the specific class
4. False negative (FN): The number of examples that incorrectly assigned to another class

The precision and recall of a multi-class classification system are defined by,

$$\text{Average Precision} = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + FP_i}, \tag{2}$$

$$\text{Average Recall} = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + FN_i}, \tag{3}$$

where N is the number of classes, TP_i is the number of true positive for class i, FN_i is the number of false negative for class i and FP_i is the number of false positive for class i.

These performance measures can be derived from the confusion matrix. The confusion matrix is represented by a matrix that shows the predicted and actual classification. The matrix is $n \times n$, where n is the number of classes. The structure of confusion matrix for multi-class classification is given by,

	Predicted Class			
	Classified as c_1	Classified as c_{12}	$\cdots$	Classified as c_{1n}
Actual Class c_1	c_{11}	c_{12}	$\cdots$	c_{1n}
Actual Class c_2	c_{21}	c_{22}	$\cdots$	c_{2n}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Actual Class c_n	c_{n1}	c_{n2}	$\cdots$	c_{nn}

(4)

This matrix reports the number of false positives, false negatives, true positives, and true negatives which are defined through elements of the confusion matrix as follows,

$$TP_i = c_{ii} \tag{5}$$

$$FP_i = \sum_{k=1}^{N} c_{ki} - TP_i \tag{6}$$

$$FN_i = \sum_{k=1}^{N} c_{ik} - TP_i \tag{7}$$

$$TN_i = \sum_{k=1}^{N}\sum_{f=1}^{N} c_{kf} - TP_i - FP_i - FN_i \tag{8}$$

Accuracy, precision and recall will be calculated for the KNN classifier using all the similarity measures and distance metrics discussed in Section 2.3, on all the datasets described in Table 5, this is to compare and asses the performance of the KNN classifier using different distance metrics and similarity measures.

3.4. Experimental results and discussion

For the purposes of this review, two sets of experiments have been conducted. The aim of the first set is to compare the performance of the KNN classifiers when used with each of the 54 distances and similarity measures reviewed in Section 2.3 without any noise. The second set of experiments is designed to find the most robust distance that affected the least with different noise levels.

3.5. Without noise

A number of different predefined distance families were used in this set of experiments. The accuracy of each distance on each dataset is averaged over 10 runs. The same technique is followed for all other distance families to report accuracy, recall, and precision of the KNN classifier for each distance on each dataset. The average values for each of 54 distances considered in the paper is summarized in Table 6, where HasD obtained the highest overall average.

Table 7, show the highest accuracies on each of the datasets obtained by which of the distances. Based on these results we summarize the following observations.

- The distance measures in L_1 family outperformed the other distance families in 5 datasets. LD achieved the highest accuracy in two datasets, namely on Vehicle and Vowel with an average accuracies of 69.13%, 97.71% respectively. On the other hand, CanD achieved the highest accuracy in two datasets, Australian and Wine datasets with an average accuracies of 82.09%, 98.5% respectively. SD and SoD achieved the highest accuracy on Segmen dataset with an average accuracy of 96.76%. Among the L_p Minkowski and L_1 distance families, the MD, NID and MCD achieved similar performance with overall accuracies on all datasets; this is due to the similarity between these distances.

- In Inner product family, JacD and DicD outperform all other tested distances on Letter rec. dataset with an average accuracy of 95.16%. Among the L_p Minkowski and L_1 distance families, the CD, JacD and DicD outperform the other tested distances on the Banknote dataset with an average accuracy of 100%.

- In Squared Chord family, MatD, SCD, and HeD achieved similar performance with overall accuracies on all datasets, this is expected because these distances are very similar.

- In Squared L_2 distance measures family, the SquD and PSCSD achieved similar performance with overall accuracy in all datasets, this is due to the similarity between these two distances. The distance measures in this family outperform the other distance families on two datasets, namely, the ASCSD achieved the highest accuracy on the German dataset with an average accuracy of 71%. ClaD and DivD achieved the highest accuracy on the Vote dataset with an average accuracy of 91.87%. Among the L_p Minkowski and Squared L_2 distance measures family, the ED, SED

Table 6: Average accuracies, recalls, precisions over all datasets for each distance. HasD obtained the highest overall average.

Distance	Accuracy	Recall	Precision	Distance	Accuracy	Recall	Precision
ED	0.8001	0.6749	0.6724	PSCSD	0.6821	0.5528	0.5504
MD	0.8113	0.6831	0.681	DivD	0.812	0.678	0.6768
CD	0.7708	0.656	0.6467	ClaD	0.8227	0.6892	0.6871
LD	0.8316	0.6964	0.6934	ASCSD	0.6259	0.4814	0.4861
CanD	0.8282	0.6932	0.6916	SED	0.8001	0.6749	0.6724
SD	0.7407	0.6141	0.6152	AD	0.8001	0.6749	0.6724
SoD	0.7881	0.6651	0.6587	SCSD	0.8275	0.693	0.6909
KD	0.6657	0.5369	0.5325	MCED	0.7973	0.6735	0.6704
MCD	0.8113	0.6831	0.681	TopD	0.6793	0.461	0.4879
NID	0.8113	0.6831	0.681	JSD	0.6793	0.461	0.4879
CosD	0.803	0.6735	0.6693	JDD	0.7482	0.5543	0.5676
ChoD	0.7984	0.662	0.6647	JefD	0.7951	0.6404	0.6251
JacD	0.8024	0.6756	0.6739	KLD	0.513	0.3456	0.3808
DicD	0.8024	0.6756	0.6739	KDD	0.5375	0.3863	0.4182
SCD	0.8164	0.65	0.4813	KJD	0.6501	0.4984	0.5222
HeD	0.8164	0.65	0.6143	TanD	0.7496	0.5553	0.5718
BD	0.4875	0.3283	0.4855	AvgD	0.8084	0.6811	0.6794
MatD	0.8164	0.65	0.5799	HamD	0.6413	0.5407	0.5348
VWHD	0.6174	0.4772	0.5871	MeeD	0.415	0.1605	0.3333
VSDF1	0.7514	0.6043	0.5125	WIAD	0.812	0.6815	0.6804
VSDF2	0.6226	0.4828	0.5621	HauD	0.5967	0.4793	0.4871
VSDF3	0.7084	0.5791	0.5621	CSSD	0.4397	0.2538	0.332
MSCSD	0.7224	0.5876	0.3769	SPeaD	0.8023	0.6711	0.6685
MiSCSD	0.6475	0.5137	0.5621	CorD	0.803	0.6717	0.6692
PCSD	0.6946	0.5709	0.5696	PeaD	0.759	0.6546	0.6395
NCSD	0.6536	0.5144	0.5148	MotD	0.7407	0.6141	0.6152
SquD	0.6821	0.5528	0.5504	HasD	**0.8394**	**0.7018**	**0.701**

Table 7: The highest accuracy in each dataset.

Dataset	Distance	Accuracy
Australian	CanD	0.8209
Balance	ChoD,SPeaD,CorD,CosD	0.933
Banknote	CD,DicD,JacD	1
BCW	HasD	0.9624
Cancer	HasD	0.9616
Diabetes	MSCSD	0.6897
Glass	LD,MCED	0.7111
Haberman	KLD	0.7327
Heart	HamD	0.7714
Ionosphere	HasD	0.9025
Liver	VSDF1	0.6581
Monkey1	WIAD	0.9497
Parkinson	VSDF1	0.9997
Phoneme	VSDF1	0.898
QSAR	HasD	0.8257
Segmen	SoD,SD	0.9676
Sonar	MiSCSD	0.8771
Vehicle	LD	0.6913
Vote	ClaD,DivD	0.9178
Vowel	LD	0.9771
Wholesale	AvgD	0.8866
Wine	CanD	0.985
German	ASCSD	0.71
Iris	ChoD,SPeaD,CorD,CosD	0.9588
Wave 21	ED,AD,MCED,SED	0.7774
Egg	ChoD,SPeaD,CorD,CosD	0.9725
Wave 40	ED,AD,MCED,SED	0.7587
Letter rec.	JacD,DicD	0.9516

and AD achieved similar performance in all datasets; this is due to the similarity between these three distances. Also, these distances and MCED outperform the other tested distances in two datasets, Wave21, Wave40 with an average accuracies of 77.74%, 75.87% respectively. Among the L_1 distance and Squared L_2 families, the MCED and LD achieved the highest accuracy on the Glass dataset with an average accuracy of 71.11%.

- In Shannon entropy distance measures family, JSD and TopD achieved similar performance with overall accuracies on all datasets, this is due to similarity between both of the distances, as the TopD is twice the JSD. KLD outperforms all the tested distances on Haberman dataset with an average accuracy of 73.27%.

- The Vicissitude distance measures family outperform the other distance families on 5 datasets, namely, VSDF1 achieved the highest accuracy in three datasets, Liver, Parkinson and Phoneme with accuracies of 65.81%, 99.97%, and 89.8% respectively. MSCSD achieved the highest accuracy on the Diabetes dataset with an average accuracy of 68.79%. MiSCSD also achieved the highest accuracy on Sonar dataset with an average accuracy of 87.71%.

- The other distance measures family outperforms all other distance families in 7 datasets. The WIAD achieved the highest accuracy on Monkey1 dataset with an average accuracy of 94.97%. The AvgD also achieved the highest accuracy on the Wholesale dataset with an average accuracy of 88.66%. HasD also achieved the highest accuracy in four datasets, namely, Cancer, BCW, Ionosphere and QSAR with an average accuracies of 96.16%, 96.24%, 90.25%, and 82.57% respectively. Finally, HamD achieved the highest accuracy on the Heart dataset with an average accuracy of 77.14%. Among the inner product and other distance measures families, the SPeaD, CorD, ChoD and CosD outperform other tested distances in three datasets, namely, Balance, Iris and Egg with an average accuracies of 94.3%, 95.88%, and 97.25% respectively.

Table 8 show the highest recalls on each of the datasets obtained by which of the distances. Based on these results we summarize the following observations.

- The L_1 distance measures family outperform the other distance families in 7 datasets, for example, CanD achieved the highest recalls in two datasets, Australian and Wine with 81.83% and 73.94% average recalls respectively. LD also achieved the highest recalls on four datasets, Glass, Ionosphere, Vehicle and Vowel with 51.15%, 61.52%, 54.85% and 97.68% average recalls respectively. SD and SoD achieved the highest recall on Segmen dataset with 84.67% average recall. Among the L_p Minkowski and L_1 distance families, the MD, NID and MCD achieved similar performance as expected, due to their similarity.

- In Inner Product distance measures family, JacD and DicD outperform all other tested distances in Heberman dataset with 38.53% average recall.

Table 8: The highest recall in each dataset.

Dataset	Distance	Recall
Australian	CanD	0.8183
Balance	ChoD,SPeaD,CorD,CosD	0.6437
Banknote	CD,DicD,JacD	1
BCW	HasD	0.3833
Cancer	HasD	0.9608
Diabetes	MSCSD	0.4371
Glass	LD	0.5115
Haberman	DicD,JacD	0.3853
Heart	HamD	0.5122
Ionosphere	LD	0.6152
Liver	VSDF1	0.4365
Monkey1	WIAD	0.9498
Parkinson	VSDF1	0.9997
Phoneme	VSDF1	0.8813
QSAR	HasD	0.8041
Segmen	SoD,SD	0.8467
Sonar	MiSCSD	0.5888
Vehicle	LD	0.5485
Vote	ClaD,DivD	0.9103
Vowel	LD	0.9768
Wholesale	PCSD	0.5816
Wine	CanD	0.7394
German	ASCSD	0.4392
Iris	ChoD,SPeaD,CorD,CosD	0.9592
Wave 21	ED,AD,MCED,SED	0.7771
Egg	ChoD,SPeaD,CorD,CosD	0.9772
Wave 40	ED,AD,MCED,SED	0.7588
Letter rec.	VSDF2	0.9514

Among the L_p Minkowski and Inner Product distance measures families, the CD, JacD and DicD also outperform the other tested distances in the Banknote dataset with 100% average recall.

- In Squared chord distance measures family, MatD, SCD, and HeD achieved similar performance; this is due to their equations similarity as clarified previously.
- The Squared L_2 distance measures family outperform the other distance families on two datasets, namely, ClaD and DivD outperform the other tested distances on Vote dataset with 91.03% average recall. The PCSD outperforms the other tested distances on Wholesale dataset with 58.16% average recall. ASCSD also outperforms the other tested distances on German dataset with 43.92% average recall. Among the L_p Minkowski and Squared L_2 distance measures families, the ED, SED and AD achieved similar performance in all datasets; this is due to their equations similarity as clarified previously. These distances and MCED distance outperform the other tested distances in two datasets, namely, the Wave21, Wave40 with 77.71% and 75.88% average recalls respectively.
- In Shannon entropy distance measures family, JSD and TopD distances achieved similar performance as expected, due to their similarity.
- The Vicissitude distance measures family outperform the other distance families in six datasets. The VSDF1 achieved the highest recall on three datasets, Liver, Parkinson and Phoneme datasets with 43.65%, 99.97%, 88.13% average recalls respectively. MSCSD achieved the highest recall on Diabetes dataset with 43.71% average recall. MiSCSD also achieved the highest recall on Sonar dataset with 58.88% average recall. The VSDF2 achieved the highest recall on Letter rec. dataset with 95.14% average recall.
- The other distance measures family outperforms the all other distance families in 5 datasets. Particularly, HamD achieved the highest recall on the Heart dataset with 51.22% average recall. The WIAD also achieved the highest average recall on the Monkey1 dataset with 94.98% average recall. HasD also has achieved the highest average recall on three datasets, namely, Cancer, BCW and QSAR with 96.08%, 38.33%, and 80.41% average recalls respectively. Among the inner product and other distance measures families, the SPeaD, CorD, ChoD and CosD outperform the other tested distances in three datasets, namely, Balance, Iris and Egg with 64.37%, 95.92%, and 97.72% average recalls respectively.

Table 9 show the highest precisions on each of the datasets obtained by which of the distances. Based on these results we summarize the following observations.

- The distance measures in L_1 family outperformed the other distance families in 5 datasets. CanD achieved the highest precision on two datasets,

Table 9: The highest precision in each dataset.

Dataset	Distance	Precision
Australian	CanD	0.8188
Balance	ChoD,SPeaD,CorD,CosD	0.6895
Banknote	CD,DicD,JacD	1
BCW	HasD	0.3835
Cancer	HasD	0.9562
Diabetes	ASCSD	0.4401
Glass	LD	0.5115
Haberman	SPeaD,CorD,CosD	0.3887
Heart	HamD	0.5112
Ionosphere	HasD	0.5812
Liver	VSDF1	0.4324
Monkey1	WIAD	0.95
Parkinson	VSDF1	0.9997
Phoneme	VSDF1	0.8723
QSAR	HasD	0.8154
Segmen	SoD,SD	0.8466
Sonar	MiSCSD	0.5799
Vehicle	LD	0.5537
Vote	ClaD,DivD	0.9211
Vowel	LD	0.9787
Wholesale	DicD,JacD	0.5853
Wine	CanD	0.7408
German	ASCSD	0.4343
Iris	ChoD,SPeaD,CorD,CosD	0.9585
Wave 21	ED,AD,MCED,SED	0.7775
Egg	ChoD,SPeaD,CorD,CosD	0.9722
Wave 40	ED,AD,MCED,SED	0.759
Letter rec.	ED,AD,SED	0.9557

namely, Australian and Wine with 81.88%, 74.08% average precisions respectively. SD and SoD achieved the highest precision on the Segmen dataset with 84.66% average precision. In addition, LD achieved the highest precision on three datasets, namely, Glass, Vehicle and Vowel, with 51.15%, 55.37%, and 97.87% average precisions respectively. Among the L_p Minkowski and L_1 distance families, the MD, NID and MCD achieved similar performance in all datasets; this is due to their equations similarity as clarified previously.

- Inner Product family outperform other distance families in two datasets. Also, JacD and DicD outperform the other tested measures on Wholesale dataset with 58.53% average precision. Among the L_p Minkowski and L_1 distance families, that the CD, JacD and DicD on the other tested distances on the Banknote dataset with 100% average precision.

- In Squared chord distance measures family, MatD, SCD, and HeD achieved similar performance with overall precision results in all datasets; this is due to their equations similarity as clarified previously.

- In Squared L_2 distance measures family, SCSD and PSCSD achieved similar performance; this is due to their equations similarity as clarified previously. The distance measures in this family outperform the other distance families on three datasets, namely, ASCSD achieved the highest average precisions on two datasets, Diabetes and German with 44.01%, and 43.43% average precisions respectively. ClaD and DicD also achieved the highest precision on the Vote dataset with 92.11% average precision. Among the L_p Minkowski and Squared L_2 distance measures families, the ED, SED and AD achieved similar performance as expected, due to their similarity. These distances and MCED outperform the other tested measures in two datasets, namely, Wave 21, Wave 40 with 77.75%, 75.9% average precisions respectively. Also, ED, SED and AD outperform the other tested measures on the Letter rec. with 95.57% average precision.

- In Shannon entropy distance measures family, JSD and TopD achieved similar performance with overall precision in all datasets, due to their equations similarity as clarified earlier.

- The Vicissitude distance measures family outperform other distance families on four datasets. The VSDF1 achieved the highest average precisions on three datasets, Liver, Parkinson and Phoneme with 43.24%, 99.97%, 87.23% average precisions respectively. MiSCSD also achieved the highest precision on the Sonar dataset with 57.99% average precision.

- The other distance measures family outperforms all the other distance families in 6 datasets. In particular, HamD achieved the highest precision on the Heart dataset with 51.12% average precision. Also, WIAD achieved the highest precision on the Monkey1 dataset with 95% average precision. Moreover, HasD yield the highest precision in four datasets, namely,

Table 10: The 10 top distances in terms of average accuracy, recall, and precision based performance on noise-free datasets.

	Accuracy			Recall			Precision	
No.	Distance	Average	No.	Distance	Average	No.	Distance	Average
1	HasD	0.8394	1	HasD	0.7018	1	HasD	0.701
2	LD	0.8316	2	LD	0.6964	2	LD	0.6934
3	CanD	0.8282	3	CanD	0.6932	3	CanD	0.6916
4	SCSD	0.8275	4	SCSD	0.693	4	SCSD	0.6909
5	ClaD	0.8227	5	ClaD	0.6892	5	ClaD	0.6871
6	DivD	0.812	6	MD	0.6831	6	MD	0.681
6	WIAD	0.812	7	WIAD	0.6815	7	WIAD	0.6804
7	MD	0.8113	8	AvgD	0.6811	8	DivD	0.6768
8	AvgD	0.8084	9	DivD	0.678	9	DicD	0.6739
9	CosD	0.803	10	DicD	0.6756	10	ED	0.6724
9	CorD	0.803						
10	DicD	0.8024						

Cancer, BCW, Ionosphere and QSR, with 38.35%, 95.62%, 58.12%, and 81.54% average precisions respectively. Among the inner product and other distance measures families, the SPeaD, CorD, ChoD and CosD outperform the other tested distances in three datasets, namely, Balance, Iris and Egg, with 68.95%, 95.85%, and 97.22% average precisions respectively. Also, CosD, SPeaD, CorD achieved the highest precision on Heberman dataset with 38.87% average precision.

Table 10 shows the top 10 distances in respect to the overall average accuracy, recall and precision over all datasets. HasD outperforms all other tested distances in all performance measures, followed by LD, CanD and SCSD. Moreover, a closer look at the data of the average as well as highest accuracies, precisions, recalls, we find that the HasD outperform all distance measures on 4 datasets, namely, Cancer, BCW, Ionosphere and QSAR, this is true for accuracy, precision and recall, and it is the only distance metric that won at least 4 datasets in this noise-free experiment set. Note that the performance of the following five group members, (1) MCD, MD, NID (2) AD, ED SED, (3) TopD, JSD, (4) SquD, PSCSD, and (5) MatD, SCD, and HeD are the same within themselves due to their close similarity in defining the corresponding distances.

We attribute the success of Hassanat distance in this experimental part to its characteristics discussed in Section 2.3 (See distance equation in 8.13, Figure 2), where each dimension in the tested vectors contributes maximally 1 to the final distance, this lowers and neutralizes the effects of outliers in different datasets. To further analyze the performance of Hassanat distance comparing with other top distances we used the Wilcoxon's rank-sum test (Wilcoxon, 1945). This is a non-parametric pairwise test that aims to detect significant differences between two sample means, to judge if the null hypothesis is true or not. Null hypothesis is a hypothesis used in statistics that assumes there is no significant difference

Table 11: The P-values of the Wilcoxon test for the results of Hassanat distance with each of other top distances over the datasets used. The P-values that were less than the significance level (0.05) are highlighted in **boldface**.

Distance	Accuracy	Recall	Precision
ED	**0.0418**	0.0582	**0.0446**
MD	0.1469	0.1492	0.1446
CanD	**0.0017**	**0.008**	**0.0034**
LD	0.0594	0.121	**0.0427**
CosD	**0.0048**	**0.0066**	**0.0064**
DicD	0.0901	0.0934	0.0778
ClaD	**0.0089**	**0.0197**	**0.0129**
SCSD	**0.0329**	0.0735	0.0708
WIAD	**0.0183**	**0.0281**	**0.0207**
CorD	**0.0048**	**0.0066**	**0.0064**
AvgD	0.1357	0.1314	0.102
DivD	**0.0084**	**0.0188**	**0.017**

between different results or observations. This test was conducted between Hassanat distance and with each of the other top distances (see Table 10) over the tested datasets. Therefore, our null hypothesis is: "there is no significant difference between the performance of Hassanat distance and the compared distance over all the datasets used". According to the Wilcoxon test, if the result of the test showed that the P-value is less than the significance level (0.05) then we reject the null hypothesis, and conclude that there is a significant difference between the tested samples; otherwise we cannot conclude anything about the significant difference (Derrac et al., 2011).

The accuracies, recalls and precisions of Hassanat distance over all the datasets used in this experiment set were compared to those of each of the top 10 distance measures, with the corresponding P-values are given in Table 11. The P-values that were less than the significance level (0.05) are highlighted in bold. As can be seen from Table 11, the P-values of accuracy results is less than the significance level (0.05) eight times, here we can reject the null hypothesis and conclude that there is a significant difference in the performance of Hassanat distance compared to ED, CanD, CosD, ClaD, SCSD, WIAD, CorD and DivD, and since the average performance of Hassanat distance was better than all of these distance measures from the previous tables, we can conclude that the accuracy yielded by Hassanat distance is better than that of most of the distance measures tested. Similar analysis applies for the recall, and precision columns comparing Hassanat results to the other distances.

3.6. With noise

These next experiments aim to identify the impact of noisy data on the performance of KNN classifier regarding accuracy, recall and precision using different distance measures. Accordingly, nine different levels of noise were added into each dataset using Algorithm 2. For simplicity, this set of experiments

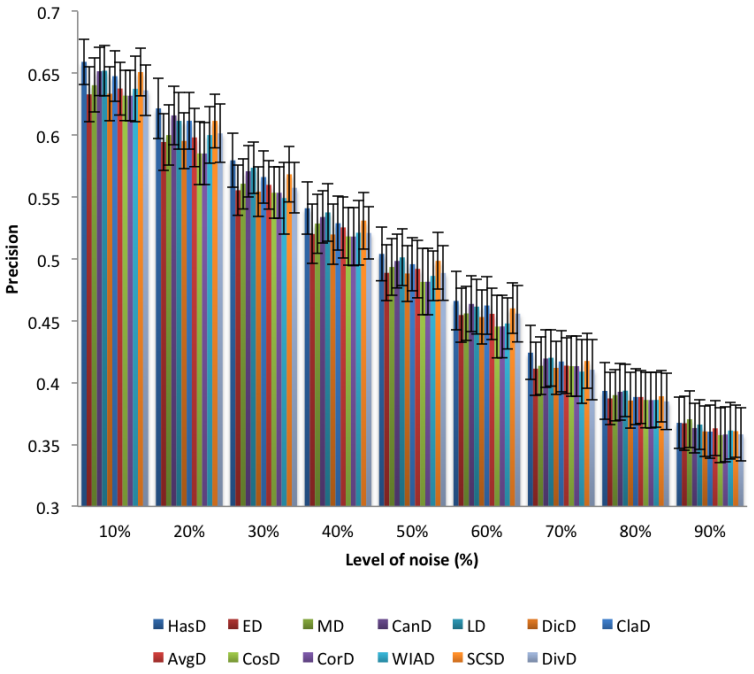

Figure 4: The overall average accuracies and standard deviations of KNN classifier using top 10 distance measures with different levels of noise.

conducted using only the top 10 distances shown in Table 10 that are obtained based on the noise-free datasets.

Figure 4 shows the experimental results of KNN classifier that clarify the impact of noise on the accuracy performance measure using the top 10 distances. X-axis denotes the noise level and Y-axis represents the classification accuracy. Each column at each noise level represents the overall average accuracy for each distance on all datasets used. Error bars represent the average of standard deviation values for each distance on all datasets. Figure 5 shows the recall results of KNN classifier that clarify the impact of noise on the performance using the top 10 distance measures. Figure 6 shows the precision results of KNN classifier that clarify the impact of noise on the performance using the top 10 distance measures. As can be seen from Figures 4, 5 and 6 the performance (measured by accuracy, recall, and precision respectively) of the KNN degraded only about 20% while the noise level reaches 90%, this is true for all the distances used. This means that the KNN classifier using any of the top 10 distances tolerate noise to a certain degree. Moreover, some distances are less affected by the added noise comparing to other distances. Therefore, we ordered the distances according to their overall average accuracy, recall and precision results for each level of noise. The distance with highest performance is ranked in the first position, while the distance with the lowest performance is ranked in the last position of the order. Tables 12, 13 and 14 show this ranking structure

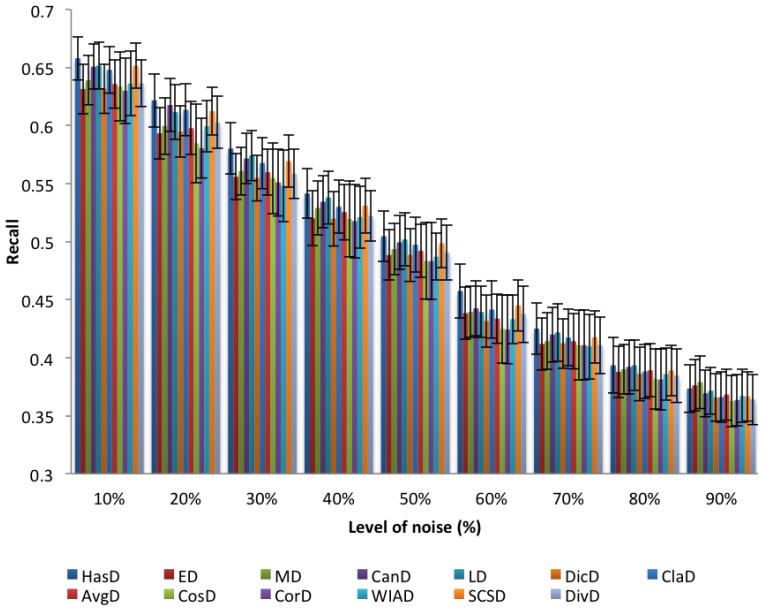

Figure 5: The overall average recalls and standard deviations of KNN classifier using top 10 distance measures with different levels of noise.

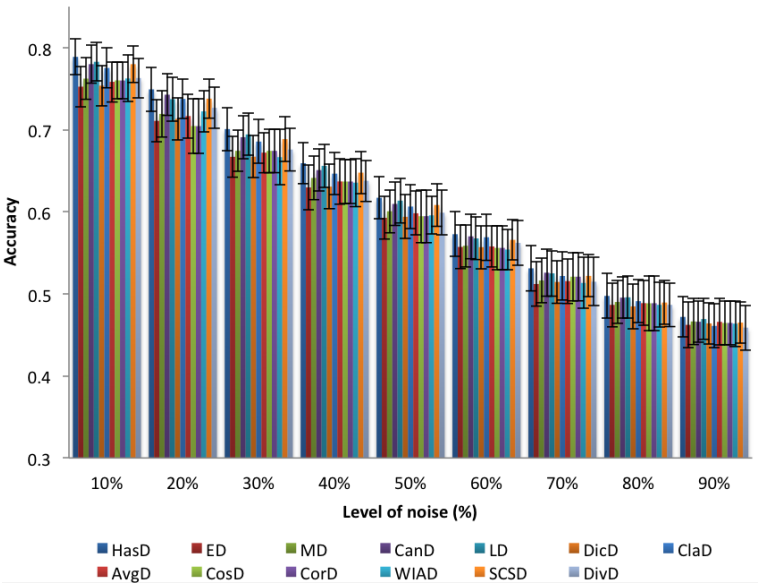

Figure 6: The overall average precisions and standard deviations of KNN classifier using top 10 distance measures with different levels of noise.

Table 12: Ranking of distances in descending order based on the accuracy results at each noise level.

Rank	10%	20%	30%	40%	50%	60%	70%	80%	90%
1	HasD	HasD	HasD	HasD	HasD	HasD	HasD	HasD	HasD
2	LD	CanD	LD	LD	LD	CanD	CanD	LD	LD
3	CanD SCSD	SCSD	CanD	CanD	CanD	ClaD	LD	CanD	MD
4	ClaD	ClaD	SCSD	SCSD	SCSD	LD	SCSD	ClaD	CanD
5	DivD	LD	ClaD	ClaD	ClaD	SCSD	ClaD	MD	AvgD
6	WIAD	DivD	DivD	MD	MD	DivD	CosD CorD	SCSD	SCSD
7	MD	WIAD	MD CosD CorD	DivD	DivD	MD	MD	AvgD	CorD
8	CD	MD	AvgD	AvgD	AvgD	AvgD	AvgD	CorD	CosD
9	CorD	AvgD	DicD	CosD CorD	WIAD	ED	DivD	CosD	DicD
10	AvgD	DicD	ED	WIAD	CoD	DicD	DicD	DivD WIAD ED	WIAD
11	DicD	ED	WIAD	DicD	CorD	CorD CosD	WIAD	DicD	ED
12	ED	CosD	-	ED	DicD	WIAD	ED	-	ClaD
13	-	CorD	-	-	ED	-	-	-	DivD

Table 13: Ranking of distances in descending order based on the recall results at each noise level.

Rank	10%	20%	30%	40%	50%	60%	70%	80%	90%
1	HasD	HasD	HasD	HasD	HasD	HasD	HasD	LD HasD	MD
2	LD	CanD	LD	LD	LD	SCSD	LD	CanD	ED
3	SCSD	ClaD	CanD	CanD	CanD	CanD	CanD	MD	HasD
4	CanD SCSD	SCSD	SCSD	SCSD	ClaD	SCSD	AvgD	LD	
5	ClaD	LD	ClaD	ClaD	ClaD	MD	ClaD	SCSD	CanD
6	MD	DivD	MD	MD	MD	LD	MD	ClaD	AvgD
7	DivD	MD	AvgD	AvgD	AvgD	ED	AvgD	ED	WIAD
8	WIOD	WIOD	DivD	DivD	DivD	DivD	DicD	DicD	SCSD
9	AvgD	AvgD	ED	WIAD	DicD	AvgD	ED	WIAD	ClaD
10	CosD	DicD	DicD	ED	ED	WIAD	CosD CorD DivD	DivD	DicD
11	DicD	ED	CosD	DicD	WIAD	DicD	WIAD	CosD	DivD
12	ED	CosD CorD	CosD	CoD	CosD	-	CorD	CorD	
13	CorD	CorD	WIOD	CorD	CorD	CorD	-	-	CosD

Table 14: Ranking of distances in descending order based on the precision results at each noise level.

Rank	10%	20%	30%	40%	50%	60%	70%	80%	90%
1	HasD	HasD	HasD	HasD	HasD	HasD	HasD	LD	MD
2	LD	CanD	LD	LD	LD	CanD	LD	HasD	HasD
3	CanD	ClaD	CanD	CanD	SCSD	ClaD	CanD	CanD	ED
4	SCSD	LD SCSD	SCSD	SCSD	CanD	LD	SCSD	MD	LD
5	ClaD	DivD	ClaD	ClaD	ClaD	SCSD	ClaD	SCSD	CanD
6	MD	MD	MD WIAD	MD	MD	MD	AvgD	ClaD	AvgD
7	AvgD	AvgD	AvgD	AvgD	AvgD	DivD	MD	AvgD	WIAD
8	WIAD	DicD	DivD	WIAD	ED	AvgD	CosD CorD	ED	SCSD
9	DivD	ED	ED	DivD	DivD	ED	DicD	CosD WIAD	DicD
10	DicD	CosD	DicD	ED	DicD	DicD	ED	CorD	ClaD
11	ED	CorD	CosD CorD	DicD	WIAD	WIAD	DivD	DicD	DivD
12	CosD	-	WIAD	CosD	CorD	CorD	WIAD	DivD	CorD
13	CorD	-	-	CorD	CosD	CosD	-	-	CosD

in terms of accuracy, precision, and recall under each noise level from low 10% to high 90%. The empty cells occur because of sharing same rank by more than one distance. The following points summarize the observations in terms of accuracy, precision, and recall values:

- According to the average precision results, the highest precision was obtained by HasD which achieved the first rank in the majority of noise levels. This distance succeeds to be in the first rank at noise levels 10% up to 70%. However, at a level 80%, LD outperformed HasD. Also, MD outperformed on the HasD at a noise level 90%.

- LD achieved the second rank at noise levels 10%, 30%, 40%, 50%, and 70%.The CanD achieved the second rank at noise levels 20% and 60%. Moreover, this distance achieved the third rank in the rest noise levels except at noise levels 50% and 90%. The SCSD achieved the fourth rank at noise levels 10%, 30%, 40%, and 70% and the third rank at a level of noise 50%. This distance was equal with the LD at a noise level 20%. The ClarD achieved the third rank at noise levels 20%, and 60%.

- The rest of distances achieved the middle and the last ranks in different orders at each level of noise. The cosine distance at level 80% was equal to the WIAD in the result. This distance was also equal with the CorD at levels 30% and 70%. These two distances performed the worst (lowest precision) in most noise levels.

Based on results in Tables 12, 13 and 14, we observe that the ranking of distances in terms of accuracy, recall and precision without the presence of noise is different with their ranking when adding the first level of noise 10% and it become variants significantly when we increased the level of noise progressively. This means that the distances are affected by noise. However, the crucial question is: which one is the distances is least affected by noise? From the above results we conclude that HasD is the least affected one, followed by LD, CanD and SCSD.

3.7. Precise evaluation of the effects of noise

In order to justify why some distances are affected either less or more by noise, the following toy Examples 3.1 and 3.2 are designed. These illustrate the effect of noise on the final decision of the KNN classifier using Hassanat (HasD) and the standard Euclidean (ED) distances. In both examples, we assume that we have two training vectors (v1 and v2) having three attributes for each, in addition to one test vector (v3). As usual, we calculate the distances between v3 and both v1 and v2 using both of Euclidean and Hassanat distances.

Example 3.1. *This example shows the KNN classification using two different distances on clean data (without noise). We find the Distance to test vector (v3) according to ED and HasD.*

	X1	*X2*	*X3*	*X4*	*Class*	*Dist(·, V3)*	
						ED	*HasD*
V1	*3*	*4*	*5*	*3*	*2*	*2*	*0.87*
V2	*1*	*3*	*4*	*2*	*1*	*1*	*0.33*
V3	*2*	*3*	*4*	*2*	*?*		

As can be seen, assuming that we use k = 1, based on the 1-NN approach, and using both distances, the test vector is assigned to class 1, both results are reasonable, because V3 is almost the same as V2 (class =1) except for the 1st feature, which differs only by 1.

Example 3.2. *This example shows the same feature vectors as in Example 3.1, but after corrupting one of the features with an added noise. That is, we make the same previous calculations using noisy data instead of the clean data; the first attribute in V2 is corrupted by an added noise of (4, i.e. X1 = 5).*

	X1	*X2*	*X3*	*X4*	*Class*	*Dist(·, V3)*	
						ED	*HasD*
V1	*3*	*4*	*5*	*3*	*2*	*2*	*0.87*
V2	*5*	*3*	*4*	*2*	*1*	*3*	*0.5*
V3	*2*	*3*	*4*	*2*	*?*		

Based on the minimum distance approach, using Euclidian distance, the test vector is assigned to class 2 instead of 1. However, the test vector is assigned to class 1 using the Hassanat distance, this makes the distance more accurate with the existence of noise. Although simple, these examples showed that the Euclidean distance was affected by noise and consequently affected the

KNN classification ability. Although the performance of the KNN classifier is decreased as the noise increased (as shown by the extensive experiments with various datasets), we find that some distances are less affected by noise than other distances. For example, when using ED any change in any attribute contributes highly to the final distance, even if both vectors were similar but in one feature there was noise, the distance (in such a case) becomes unpredictable. In contrast, with the Hassanat distance we found that the distance between both consecutive attributes are bounded in the range $[0, 1]$, thus, regardless of the value of the added noise, each feature will contributes up to 1 maximally to the final distance, and not proportional to the value of the added noise. Therefore the impact of noise on the final classification is mitigated.

4. Conclusions

In this review, the performance (accuracy, precision and recall) of the KNN classifier has been evaluated using a large number of distance measures, on clean and noisy datasets, attempting to find the most appropriate distance measure that can be used with the KNN in general. In addition we tried finding the most appropriate and robust distance that can be used in the case of noisy data. A large number of experiments conducted for the purposes of this review, and the results and analysis of these experiments show the following:

1. The performance of KNN classifier depends significantly on the distance used, the results showed large gaps between the performances of different distances. For example we found that Hassanat distance performed the best when applied on most datasets comparing to the other tested distances.
2. We get similar classification results when we use distances from the same family having almost the same equation, some distances are very similar, for example, one is twice the other, or one is the square of another. In these cases, and since the KNN compares examples using the same distance, the nearest neighbors will be the same if all distances were multiplied or divided by the same constant.
3. There was no optimal distance metric that can be used for all types of datasets, as the results show that each dataset favors a specific distance metric, and this result complies with the no-free-lunch theorem.
4. The performance (measured by accuracy, precision and recall) of the KNN degraded only about 20% while the noise level reaches 90%, this is true for all the distances used. This means that the KNN classifier using any of the top 10 distances tolerate noise to a certain degree.
5. Some distances are less affected by the added noise comparing to other distances, for example we found that Hassanat distance performed the best when applied on most datasets under different levels of heavy noise.

Our work has the following limitations, and future works will focus on studying, evaluating and investigating these.

1. Although, we have tested a large number of distance measures, there are still many other distances and similarity measures that are available in the machine learning area that are need to be tested and evaluated for performance.
2. The 28 datasets though higher than previously tested, still might not be enough to draw significant conclusions in terms of the effectiveness of certain distance measures, and therefore, there is a need to use larger number of datasets with varied data types.
3. The added noise used in this review might not simulate the other types of noise that occur in the real world, so other types of noise need to be used to corrupt data, so as to evaluate the distance measures more robustly.
4. Only KNN classifier was reviewed in this work, other variant of KNN such as the approximate KNN need to be investigated.
5. Distance measures are not used only with the KNN, but also with other machine learning algorithms, such as different types of clustering, those need to be evaluated under different distance measures.

References

Abbad, A., & Tairi, H. (2016). Combining Jaccard and Mahalanobis Cosine distance to enhance the face recognition rate. WSEAS Transactions on Signal Processing, 16, 171–178.

Akila, A., & Chandra, E. (2013). Slope finder – A distance measure for DTW based isolated word speech recognition. International Journal of Engineering And Computer Science, 2 (12), 3411–3417.

Alkasassbeh, M., Altarawneh, G. A., & Hassanat, A. B. (2015). On enhancing the performance of nearest neighbour classifiers using Hassanat distance metric. Canadian Journal of Pure and Applied Sciences, 9 (1), 3291–3298.

Alpaydin, E. (1997). Voting over multiple condensed nearest neighbors. Artificial Intelligence Review, 11, 115–132.

Arya, S., & Mount, D. M. (1993). Approximate nearest neighbor queries in fixed dimensions. 4th annual ACM/SIGACT-SIAM Symposium on Discrete Algorithms.

Bajramovic, F., Mattern, F., Butko, N., & Denzler, J. (2006). A comparison of nearest neighbor search algorithms for generic object recognition. In Advanced Concepts for Intelligent Vision Systems (pp. 1186–1197). Springer.

Bharkad, S. D., & Kokare, M. (2011). Performance evaluation of distance metrics: application to fingerprint recognition. International Journal of Pattern Recognition and Artificial Intelligence, 25 (6), 777–806.

Bhatia, N., & Vandana. (2010). Survey of Nearest Neighbor Techniques. International Journal of Computer Science and Information Security, 8 (2), 302–305.

Bhattachayya, A. (1943). On a measure of divergence between two statistical population defined by their population distributions. Bulletin Calcutta Mathematical Society, 35, 99–109.

Cesare, S., and Xiang, Y. (2012). Software Similarity and Classification. Springer.

Cha, S.-H. (2007). Comprehensive survey on distance/similarity measures between probability density functions. International Journal of Mathematical Models and Methods in Applied Sciences, 1(4), 300–307.

Chetty, M., Ngom, A., & Ahmad, S. (2008). Pattern Recognition in Bioinformatics. Springer.

Chomboon, K., Pasapichi, C., Pongsakorn, T., Kerdprasop, K., & Kerdprasop, N. (2015). An empirical study of distance metrics for k-nearest neighbor algorithm. In The 3rd International Conference on Industrial Application Engineering 2015 (pp. 280–285).

Clark, P. J. (1952). An extension of the coefficient of divergence for use with multiple characters. Copeia, 1952 (2), 61–64.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13 (1), 21–27.

Derrac, J., Garcia, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation, 1 (1), 3–18.

Deza, E., and Deza, M. M. (2009). Encyclopedia of distances. Springer.

Dice, L. R. (1945). Measures of the amount of ecologic association between species. Ecology, 26 (3), 297–302.

Euclid. (1956). The Thirteen Books of Euclid's Elements. Courier Corporation.

Fix, E. and Hodges, J.L. (1951). Discriminatory analysis. Nonparametric discrimination; consistency properties. Technical Report 4, USAF School of Aviation Medicine, Randolph Field, TX, USA, 1951.

Fulekar, M. H. (2009). Bioinformatics: Applications in life and environmental sciences. Springer.

Gan, G., Ma, C., & Wu, J. (2007). Data clustering: Theory, algorithms, and applications. SIAM.

Garcia, S., Luengo, J., & Herrera, F. (2014). Data preprocessing in data mining. Springer.

Gates, G. (1972). The reduced nearest neighbour rule. IEEE Transactions on Information Theory, 18, 431–433.

Giusti, R., & Batista, G. (2013). An empirical comparison of dissimilarity measures for time series classification. Brazilian Conference on Intelligent Systems (BRACIS) (pp. 82–88). Fortaleza: IEEE.

Grabusts, P. (2011). The choice of metrics for clustering algorithms. Environment. Technology. Resources , 70–76.

Hamming, R. W. (1958). Error detecting and error correcting codes. Bell System technical journal, 131 (1), 147–160.

Hart, P. (1968). The condensed nearest neighbour rule. IEEE Transactions on Information Theory, 14, 515–516.

Hassanat, A. B. (2014). Dimensionality invariant similarity measure. Journal of American Science, 10 (8), 221–26.

Hassanat, A. B. (2014). Solving the problem of the k parameter in the KNN classifier using an ensemble learning approach. International Journal of Computer Science and Information Security, 12 (8), 33–39.

Hassanat, A. B., Abbadi, M. A., Altarawneh, G. A., & Alhasanat, A. A. (2014). Solving the problem of the k parameter in the KNN classifier using an ensemble learning approach. International Journal of Computer Science and Information Security, 12 (8), 33–39.

Hatzigiorgaki, M., & Skodras, A. (2003). Compressed domain image retrieval: a comparative study of similarity metrics. Proceedings of SPIE 5150 , 439-448.

Hedges, T. (1976). An empirical modification to linear wave theory. Proc. Inst. Civ. Eng., 61, 575–579.

Hellinger, E. E. (1909). Neue Begründung der Theorie quadratischer Formen von unendlichvielen Veränderlichen. für die reine und angewandte Mathematik, 136, 210–271.

Hu, L.-Y., Huang, M.-W., Ke, S.-W., & Tsai, C.-F. (2016). The distance function effect on k-nearest neighbor classification for medical datasets. SpringerPlus, 5 (1), 1304.

Jaccard, P. (1901). Etude comparative de la distribution florale dans une portion des Alpes et du Jura.

Jeffreys, H. (1946). An invariant form for the prior probability in estimation problems. In Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences (Vol. 186, pp. 453–461).

Jirina, M., & Jirina, M. J. (2008). Classifier based on inverted indexes of neighbors. Institute of Computer Science. Academy of Sciences of the Czech Republic.

Jirina, M., & Jirina, M. (2010). Using singularity exponent in distance based classifier. Proceedings of the 10th International Conference on Intelligent Systems Design and Applications (ISDA2010). Cairo.

Kadir, A., Nugroho, L. E., Susanto, A., & Insap, P. S. (2012). Experiments of distance measurements in a foliage plant retrieval system. International Journal of Signal Processing, Image Processing and Pattern Recognition, 5 (2), 47–60.

Kataria, A., & Singh, M. D. (2013). A Review of data classification Using K-nearest neighbour Algorithm. International Journal of Emerging Technology and Advanced Engineering, 3 (6), 354–360.

Kubat, M., & Cooperson, Jr., M. (2000). Voting nearest-neighbour subclassifiers. Proceedings of the 17th International Conference on Machine Learning (ICML), (pp. 503–510). Stanford, CA, USA.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. The annals of mathematical statistics, 22 (1), 79–86.

Lance, G. N., & Williams, W. T. (1967). Mixed-data classificatory programs I - Agglomerative systems. Australian Computer Journal, 1 (1), 15–20.

Legendre, P., & Legendre, L. (2012). Numerical ecology. Elsevier.

Lichman, M. (2013). Retrieved from UC Irvine Machine Learning Repository: http://archive.ics.uci.edu/ml

Lindi, G. A. (2016). Development of face recognition system for use on the NAO robot. Stavanger University, Norway.

Lopes, N., & Ribeiro, B. (2015). On the Impact of Distance Metrics in Instance-Based Learning Algorithms. Iberian Conference on Pattern Recognition and Image Analysis (pp. 48–56). Springer.

Macklem, M. (2002). Multidimensional Modelling of Image Fidelity Measures. Burnaby, BC, Canada: Simon Fraser University.

Manne, S., Kotha, S., & Fatima, S. S. (2012). Text categorization with K-nearest neighbor approach. In Proceedings of the International Conference on Information Systems Design and Intelligent Applications 2012 (INDIA 2012) held in Visakhapatnam, India, January 2012 (Vol. 132, pp. 413–420). Springer.

Nettleton, D. F., Orriols-Puig, A., & Fornells, A. (2010). A study of the effect of different types of noise on the precision of supervised learning techniques. Artificial intelligence review , 33 (4), 275–306.

Neyman, J. (1949). Contributions to the theory of the $\chi 2$ test. in proceedings of the first Berkeley symposium on mathematical statistics and probability.

Orloci, L. (1967). An agglomerative method for classification of plant communities. Journal of Ecology , 55(1), 193–206.

Patel, B., & Meshram, B. (2012). Content based video retrieval systems. International Journal of UbiComp, 3(2).

Pearson, K. (1900). On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 50 (302), 157–175.

Pinto, D., Benedi, J.-M., & Rosso, P. (2007). Clustering narrow-domain short texts by using the Kullback-Leibler distance. International Conference on Intelligent Text Processing and Computational Linguistics, 611–622.

Premaratne, P. (2014). Human computer interaction using hand gestures. Springer.

Punam, M., & Nitin, T. (2015). Analysis of distance measures using k-nearest. International Journal of Science and Research, 7 (4), 2101–2104.

Rubner, Y., & Tomasi, C. (2013). Perceptual metrics for image database navigation. Springer.

Saez, J. A., Galar, M., Luengo, J., & Herrera, F. (2013). Tackling the problem of classification with noisy data using multiple classifier Systems: Analysis of the performance and robustness. Information Sciences, 247, 1–20.

Shannon, C. E. (2001). A mathematical theory of communication. ACM SIGMOBILE Mobile Computing and Communications Review, 3–55.

Shirkhorshidi, A. S., Aghabozorgi, S., & Wah, T. Y. (2015). A Comparison Study on Similarity and Dissimilarity Measures in Clustering Continuous Data. PloS one, 10 (12), e0144059.

Sibson, R. (1969). Information radius. Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete, 14 (2), 149–160.

Sorensen, T. (1948). A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. Biol. Skr., 5, 1–34.

Szmidt, E. (2013). Distances and similarities in intuitionistic fuzzy sets. Springer.

Taneja, I. J. (1995). New developments in generalized information measures. Advances in Imaging and Electron Physics, 91, 37–135.

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. In IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA 2009) (pp. 1–6).

Todeschini, R., Ballabio, D., & Consonni, V. (2015). Distances and other dissimilarity measures in chemometrics. Encyclopedia of Analytical Chemistry.

Todeschini, R., Consonni, V., Grisoni, F. G., & Ballabio, D. (2016). A new concept of higher-order similarity and the role of distance/similarity measures in local classification methods. Chemometrics and Intelligent Laboratory Systems, 157, 50–57.

Topsoe, F. (2000). Some inequalities for information divergence and related measures of discrimination. IEEE Transactions on information theory, 46 (4), 1602–1609.

Verma, J. P. (2012). Data Analysis in Management with SPSS Software. Springer.

Wettschereck, D., Aha, D. W., & Mohri, T. (1997). A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. Artificial Intelligence Review, 11 (1-5), 273–314.

Whittaker, R. H. (1952). A study of summer foliage insect communities in the Great Smoky Mountains. Ecological monographs, 22 (1), 1–44.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. Biometrics Bulletin, 1 (6), 80–83.

Willett, P., Barnard, J. M., & Downs, G. M. (1998). Chemical similarity searching. Journal of chemical information and computer sciences , 38 (6), 983–996.

Williams, W. T., & Lance, G. N. (1966). Computer programs for hierarchical polythetic classification ("similarity analyses"). The Computer, 9 (1), 60–64.

Wilson, D. R., & Martinez, T. R. (2000). Reduction techniques for exemplar-based learning algorithms. Machine learning, 38 (3), 257–286.

Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., et al. (2008). Top 10 algorithms in data mining. Knowledge and information systems, 14 (1), 1–37.

Xiubo, G., Tie-Yan, L., Qin, T., Andrew, A., Li, H., & Shum, H.-Y. (2008). Query dependent ranking using k-nearest neighbor. In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval (pp. 115–122).

Xu, S., & Wu, Y. (2008). An algorithm for remote sensing image classification based on artificial immune B-cell network. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 37, 107–112.

Yang, Y., Ault, T., Pierce, T., & Lattimer, C. W. (2000). Improving text categorization methods for event tracking. In Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval (pp. 65–72).

Zheng, Y., Guo, Q., Tung, A. K., & Wu, S. (2016). LazyLSH: Approximate nearest neighbor search for multiple distance functions with a single index. International Conference on Management of Data (pp. 2023–2037). ACM.

Zhou, T., Chan, K. C., & Wang, Z. (2008). TopEVM: using co-occurrence and topology patterns of enzymes in metabolic networks to construct phylogenetic trees. In IAPR International Conference on Pattern Recognition in Bioinformatics (pp. 225–236). Springer.

Zhu, X., & Wu, X. (2004). Class noise vs. attribute noise: A quantitative study. Artificial Intelligence Review, 22 (3), 177–210.