Effect of Distance Metric and Feature Scaling on KNN Algorithm while Classifying X-rays

Ishan Arora¹, Namit Khanduja¹, Mayank Bansal¹

¹ Gurukul Kangri University, Haridwar, Uttrakhand. India

Abstract

In this world today medical imaging is prominently one in every of the foremost advancing fields. Medical imaging refers to those techniques and processes that are accustomed create images of assorted parts of the soma for diagnostic and treatment purposes [1]. These include various radiological imaging like X-ray radiography, Magnetic resonance imaging (MRI), etc. Many Machine learning and Deep learning algorithms are being employed in medical imaging to spot diseases. There are many algorithms like Convolution Neural Network, Multilayer Perceptron, Logistic Regression, etc. among which one among the algorithms is k-NN (k-Nearest Neighbor) the foremost drawbacks related to the k-NN algorithm are (1) its low efficiency, due to which it takes a lot of time in calculating the distance between query point and the dataset containing many features, (2) its dependency on various distances metrics. During this paper, we propose a k-NN type classifier that is overcoming these shortcomings. Our approach constructs a k-NN model that performs classification by considering various distance metrics and feature scaling so to enhance the classification accuracy. Our Experiments were administrated on some public medical datasets collected from the Kaggle Machine Learning repository to check our approach. Experiment results show that the k-NN model works well with Canberra distance metric and Robust feature scaling, but is more efficient than widely used Euclidean distance and standard feature scaling.

Keywords

Medical Imaging, k-NN, Robust Feature Scaling, Canberra Distance Metric

1. Introduction

The k-Nearest-Neighbors (k-NN) may be a non-parametric classification method, which is straightforward but effective in many cases [2]. It is a Machine learning algorithm that supported the Supervised Learning approach. Its problem domain mainly consists of a classification problem instead of a regression problem. For a Data point P to be classified, Its k-nearest neighbors are retrieved by measuring the distance between a given point and the associated data points, among which the minimum distance is further mapped to classification output. However, to use the k-NN algorithm we have got to settle on an appropriate value of k and distance metric along with appropriate feature scaling so to reduce the computational time required for computing distance between test point and dataset, and also the classification accuracy is extremely much addicted to this value. There are many distance metric and feature scaling which may take classification accuracy far better. There are many ways to settle on distance metrics and feature scaling, but the easy way is to run the algorithm again and again with different distance metrics and feature scaling and choose the one with the simplest performance.

CEUR Workshop Proceedings

RIF'21: The 10th Seminary of Computer Science Research at Feminine, March 08h, 2021, Constantine 2-Abdelhamid Mehri University, Algeria

EMAIL: aroraishan51@gmail.com, (I. Arora); namit.khanduja@gmail.com (N. Khanduja); mayankbansal231@gmail.com, (M. Bansal) © 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

Nowadays, Medical Imaging is one in every of the foremost advancing fields. One requires a classifier that will easily process an X-ray image or another radiological image and gave a decent accuracy.

There are many advanced algorithms like Convolutional Neural Network (CNN), Region proposal Convolutional Network (R-CNN) that may easily provide far better accuracy on the medical domain dataset but with more computational resources. On the other hand Linear classification algorithms such as SVM (Supported Vector Machine) and Logistic Regression still uses parametric classification approach and use some optimization algorithms such as stochastic gradient descent or batch gradient descent to train there parameters before performing any classification on test point. However, K-NN being a non-parametric classifier does not require any high-level computational resources and prior training as linear classifiers, although it might take time complexity of $O(n^2)$ to classify a knowledge point. Deep learning algorithms are data-hungry, k-NN can easily outperform them on a little amount of information. k-NN encompasses a very high cost of classifying new instances this is often thanks to all the computations happen at the time of classification instead of all the training data encountered first. There are techniques [2, 3] that will still reduce the computation required at query time, like indexing training examples, but it is out of our concerns during this paper. Hence we purpose a k-NN type model during this paper to search out whether the performance of k-NN is improved by choosing the suitable feature scaling and Distance metric. Finally, the performance of the model is measured with ROC, AUC, Precision, etc.

2. Related Works

Several studies have been conducted to analyze the performance of KNN classifier using different distance measures and feature scaling. 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 et al [3] analyzed the performance of KNN classifier using 11 distance measures and standard feature scaling. These include Euclidean, Mahalanobis, Manhattan, Minkowski, Chebyshev, 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, Chebyshev, Euclidean, Mahalanobis, and Standardized Euclidean distance measures achieved similar accuracy results and outperformed other tested distances.

Hu et al [4] 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 Minkowski 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 Minkowski distance metrics performed the 'worst' over the mixed type of datasets. The 'worst' performance means the method with the lowest accuracy.

Todeschini et al [5, 6] 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 and Ribeiro [7] analyzed the impact of five distance metrics, namely Euclidean, Manhattan, Canberra, Chebyshev and Minkowski in instance-based learning algorithms. Particularly, 1-NN

Classifier and the Incremental Hyper sphere 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 et al [8] investigated the effect of Euclidean, Manhattan and a non-convex distance due to Hassanat [9] 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) [10], and the Inverted Indexes of Neighbors Classifier (IINC) [11]. Their experiments were conducted on 28 datasets taken from the UCI machine learning repository, the reported results show that Hassanat distance [12] outperformed both of Manhattan and Euclidean distances in most of the tested datasets using the three investigated classifiers.

Table 1

Reference	Distances Used	Datasets	Best Distance	Feature Scaling	Best Feature Scaling
[3]	11	8	Manhattan, Chebyshev,	Standard	None
			Euclidean, Minkowski		
[4]	4	37	Chi-square	Standard	Standard Scaling
[7]	5	15	Euclidean, Manhattan	Standard	None
[5]	18	8	Manhattan,	Standard	None
			Contracted	:1	
			Jaccard–		
			Tanimoto		
			Lance–Williams		
[8]	3	28	Hassanat	Standard	None
[13]	3	2	Hassanat	Standard	Standard Scaling
Ours	8	3	Canberra	Standard	, Robust
				Robust,	
				Min-Max	

Comparison between previous studies for distance metric and Feature scaling for k-NN along with 'best' performing metric and Feature scaling. Comparatively our work compares the use different variety of feature scaling and distance measures on variety of datasets.

Lindi [13] 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. Above Table provides a summary of these previous works on evaluating various distances and feature scaling within *k*-NN classifier, along with the best distance assessed 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 feature scaler or different distance measures or both.

3. k-NN with Distance metric and Feature Scaling

3.1 Basic Overview of k-NN algorithm

The k-NN classifier works by computing the distance between the unlabeled datum point and also the majority of comparable data points among k-nearest neighbors that are closed thereto query point. The distance between the test sample and the training data samples is calculated by a specific distance metric.



Figure 1: AN example of *k*-NN type classifier classifying the query point (red dot) between two classes

Since the above figure contains two features they can be represented in two-dimensional space with one dimension for every feature. To classify the test sample that belongs to the class 'yellow dot' or class 'purple dot' *k*-NN adopts a distance function to seek out the *k* nearest neighbors to the test sample as shown in fig. Here when k = 3 test sample is classed to the class 'purple dot' because there are two 'purple dot' and one 'yellow dot' inside the inner circle, but when k = 6 test sample is assessed to the class 'yellow dot' because there are three 'yellow dot' and no 'purple dot' inside the outer circle.

The Basic *k*-NN classifier steps may be described as follows: Algorithm :

INPUT: Training Samples X, Test Sample P, Value of K, Distance metric

- OUTPUT: Class Label of Test Sample
 - Compute the Distance between the Test Sample P and Training samples X.
 - Choose the value of k (nearest neighbors for consideration).
 - Assign the Test Label P the bulk class.

3.2 Distance Metrics

In math, the term distance is defined as "how far the two objects are from one another". Euclid, one among the foremost important mathematicians of ancient history, used the word distance only in his third postulate of Principia [14]: "Every circle can be described by a center ad distance". In data mining, distance means how elements in some space are close or isolated from each other. The distance function between vectors x and y is function d(x, y) that defines the distance between two vectors as a non-negative real number. This function is said to be a metric if it satisfies some properties [15] that include the following:

1. Non-negativity: It means the distance function between the vectors is always greater than zero.

 $d(x, y) \ge 0$

2. Identity of indiscernible: The distance between x and y is zero if and only if x *is equal to* y.

d(x, y) = 0 if 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 a third point p, the distance between x and y is always less than the sum of the distance between x and p and the distance between y and p.

 $d(x, y) \le d(x, p) + d(y, p)$

Here in our experiment, we are considering the eight types of distance metrics. We gave the mathematical definition of distances to measure the closeness between two vectors x and y, where x = $(x_1, x_2, x_3, - - - - x_n)$ and y = $(y_1, y_2, y_3, - - - y_n)$ having numeric attributes. Theoretical analysis of these distances is beyond the scope of this work.

5. Euclidean (ED): Also known as 12 norm or Euler 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 value in vectors.

Euclidean distance (x, y) = $\sqrt{\sum_{i=1}^{m} (x_i - y_i)^2}$

6. Manhattan (MD): It is also known as 11 norm or City block distance. This distance is considered by Hermann Minkowski which represents the sum of absolute differences between the opposite values in vectors. Manhattan distance $(x, y) = \sum^{m} |x_i - y_i|$

$$|x_i - y_i|_{i=1}$$

7. Chebyshev (CD): Also known as maximum value distance [5]. It is a metric defined on vector space where the distance between the two vectors is the greatest of their difference in any coordinate dimension.

Chebyshev distance $(x, y) = \max(|x_i - y_i|)$

8. Canberra (CnD): Canberra distance, which is introduced by [6] and modified in [7]. It is an expanded version of the Manhattan distance, where the absolute difference between the feature values of vectors is divided by the sum of absolute feature values before the summing [8]

Canberra Distance = $\frac{\sum |x_i - y_i|}{\sum |x_i + y_i|}$

9. Cosine (CosD): Also known as angular distance, measures the angle between two vectors.

Cosine distance (x, y) = $\frac{\sum x_i y_i}{\sqrt{x_i^2 y_i^2}}$

10. BrayCurtis (BRD): It is the distance between the range {0,1} and undefined if the vector length is zero

BrayCurtis distance (x, y) = $\frac{\sum_{i}^{m} |x_i - y_i|}{\sum_{i}^{m} x_i + y_i}$

11. 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.

Correlation distance (x, y) = $\frac{1}{2} \left(1 - \frac{\sum_{i=1}^{m} (x_i - x')(y_i - y')}{\sqrt{\sum_{i=1}^{m} (x_i - x')^2} \sqrt{\sum_{i=1}^{m} (y_i - y')^2}} \right)$

12. Minkowski Distance (MD): The Minkowski distance, which is also known as L_p norm, is a generalized metric. It is defined as:

Euclidean	$d(x, y) = \sqrt{\sum (x_i - y_i)^2}$
Squared Euclidean	$d(x, y) = \sum (x_i - y_i)^2$
Manhattan	$d(x, y) = \sum_{i} (x_i - y_i)$
Canberra	$d(x, y) = \sum \frac{ x_i - y_i }{ x_i + y_i }$
Chebychev	$d(\mathbf{x}, \mathbf{y}) = \max(\mathbf{x}_i - \mathbf{y}_i)$
Bray Curtis	$d(x, y) = \frac{\sum x_i - y_i }{\sum x_i + y_i}$
Cosine Correlation	$d(x, y) = \frac{\sum (x_i y_i)}{\sqrt{\sum (x_i)^2 \sum (y_i)^2}}$
Pearson Correlation	$d(x, y) = \frac{\sum (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum (y_i - \overline{y})^2} \sqrt{\sum (y_i - \overline{y})^2}}$
Uncentered Peason Correlation	$d(x, y) = \frac{\sum x_i y_i}{\sqrt{\sum (y_i - \overline{y})^2} \sqrt{\sum (y_i - \overline{y})^2}}$
Euclidean Nullweighted	Same as Euclidean, but only the indexes where both x and y have a value (not NULL) are used, and the result is weighted by the number of values calculated. Nulls must be replaced by the missing value calculator (in dataloader).

Minkowski distance (x, y) = $(\sum_{i=1}^{m} (x_i - y_i))^p$

Figure 2: shows the various distance metrics

3.3 Feature Scaling

Data preprocessing is the technique in data mining to transform raw data into an understandable format. One of the technique comes under Data preprocessing is Feature scaling. Feature scaling usually refers to the Normalization or Standardization of data around the certain mean value or standard deviation value so to increase the rate of convergence of some optimization algorithms like Gradient Descent, Adam optimizer, etc. required by Linear classifiers and also to reduce the computational time required for computing distance between the points as in case of distance measure algorithms like *k*-NN, *k*-Means etc. There are many types of feature scaling methods available but our research is centered around the most commonly used three types of feature scaling discussed below:

13. Min max Scaler: It scales the data based on the minimum and the maximum of features within a range. Let us say after scaling our x feature vector becomes x'.





Figure3 : shows the feature before and after Min max scaling

14. Standard Scaler: Another rescaling method compared to the Min max scaler is Standard Scaler. It transforms feature vector x to the standard normally distributed, with mean *zero* and standard deviation *one*.





Figure 4: shows the feature before and after standard scaling

15. Robust Scaler: This type of feature scaling is commonly used to overcome the presence of *outliers* in our data. It rescales features using the interquartile range. Where X' is the new feature vector, Q1 is the first quartile and Q3 is the third quartile. All the above- mentioned feature scaling can easily be done using *python* library *scikit-learn*.

$$x_i' = x_i - Q_1 / Q_3 - Q_1$$



Figure 5: shows the feature before and after Robust scaling

4. Experimental Framework

4.1 Datasets used for experiments

The experiments were done on three different datasets which represent real-life classification problems, collected from the Kaggle Machine Learning repository. X-ray images are collected from Covid-19 related papers from medRxiv, bioRxiv, JAMA, Lancet, etc. Kaggle is an open-source online community where data scientists and machine learning practitioners, can easily access and publish their datasets. Each dataset consists of a set of examples, Each example is defined by the number of attributes and, all the examples in data are represented by the same number of attributes. One of these attributes is known as the class attribute or the class (label) which has to be predicted for the test sample.



Figure 6: Framework of our experimental setup

4.2 Experiment Setup

From above figure details of each step is as follows : Each dataset is divided into three data sets, one for training, the other for validation, and the rest for testing. For, this purpose 60% of the dataset is used for training, 20% for cross-validation, and the rest 20% for testing. We explicitly set the value of k = 1 and the 20% of data which were used as a test sample is randomly shuffled and each experiment is repeated for eight times for each value of distance metric and after each experiment value of k is incremented and execution goes on till we reach the value of k = 15.



Figure 7. shows X-ray image feeding into a *k*-NN classifier.

The Experiment also aims to find the best feature scaling used by the *k*-NN classifier. Hence all the three feature scaling as discussed above is used and the above-discussed algorithm is repeated three times each time for each feature scaling.

4.3 Performance Evaluation measures

Different measures were available for evaluating the performance of classifiers. In this paper, two measures were used accuracy and ROC-AUC curve. Accuracy is calculated to evaluate the overall performance of the classifier. It is defined as the ratio of test samples correctly classified to the total number of samples.

$$Accuracy = \frac{Number of correctly classified sample}{Number of Total samples}$$

We have plotted the graph of Accuracy v/s Value of K for best feature scaling i.e. Robust Feature scaling of all three datasets along with each distance metric.



Graph 1: showing the accuracy v/s K for each distance metric with Robust Feature scaling for dataset 1

Graph 1 shows the classification accuracy of k-NN over dataset 1. For the distance function, it is clear that Canberra distance metric is winner of all with 85.56% accuracy. However, overall speaking, using the Euclidean distance function is not the best metric for k-NN for dataset 1. The classification accuracy by Euclidean and Minkowski distance functions are almost the same, which means that using euclidean or minkowski does not affect the k-NN performance. On the other hand, k-NN by the Bray Curtis distance function gave a tough competition to best distance function i.e. Canberra distance function wit classification accuracy of 83.95%.



Graph2: showing the accuracy v/s K for each distance metric with Robust Feature scaling for dataset 2.

Graph 2 shows the classification accuracy of k-NN over dataset 2. For the distance function, it is clear that Canberra distance metric is winner of all with 77.09% accuracy. However, overall speaking, using the Euclidean distance function is not the best metric for k-NN for dataset 2. The classification accuracy by Euclidean and Minkowski distance functions are almost the same, which means that using euclidean or minkowski does not affect the k-NN performance. On the other hand, k-NN by the Bray Curtis distance function gave a tough competition to best distance function i.e. Canberra distance function wit classification accuracy of 77.08%.



Graph3: showing the accuracy v/s K for each distance metric with Robust Feature scaling for dataset 3

For the distance function, it is clear that Canberra distance metric is winner of all with 84.61% accuracy. However, overall speaking, using the Euclidean distance function is not the best metric for *k*-NN for dataset 2. The classification accuracy by Euclidean and Minkowski distance functions are almost the same, which means that using Euclidean or Minkowski does not affect the *k*-NN performance. On the other hand, *k*-NN by the Bray Curtis, Correlation, Cosine distance function gave a tough competition to best distance function i.e. Canberra distance function with their average classification accuracy of 83.4%.

AUC – ROC curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree of measure of separability. It tells us how much the model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s. By analogy, higher the AUC better the model is at distinguishing between patients with disease and no disease.



Figure 8: shows the ROC-AUC evaluation metric

To evaluate performance with every class in a dataset. We compute the ROC-AUC curve for each dataset for the best distance metric and comparable distance metrics with the best values of k associated with them. All three curves have best feature scaling i.e. Robust feature scaling

Graph 4 shows the ROC-AUC curve of three comparable distance metric over dataset 3. For the distance function, it is clear from top right ROC-AUC curve that Canberra distance metric is winner of all with maximum AUC score of 0.9178. However, overall speaking, using the bray Curtis and Manhattan distance function gave comparable results with AUC score of 0.9068, 0.8787 for dataset 1 having value of k as 4.

Graph 5 shows the ROC-AUC curve of three comparable distance metric over dataset 2. For the distance function, it is clear from top right ROC-AUC curve that Canberra distance metric is winner of all with maximum AUC score of 0.8047. However, overall speaking, using the bray Curtis and Cosine distance function gave comparable results with AUC score of 0.78, for dataset 2 having value of k as 13.

Graph 6 shows the ROC-AUC curve of three comparable distance metric over dataset 3. For the distance function, it is clear from top left ROC-AUC curve that Canberra distance metric is winner of all with maximum AUC score of 0.907774. However, overall speaking, using the Euclidean and Manhattan distance function gave comparable results with AUC score of 0.8864, 0.910 for dataset 3 having value of k as 15.







Graph 5: showing the ROC-AUC curve for Dataset 2



Graph 6: showing the ROC-AUC curve for Dataset 3

5. Conclusion and Future Prespectives

In this review, the performance (accuracy, ROC-AUC) of the k-NN classifier has been evaluated using various distance measures and Feature scaling, attempting to search out the most appropriate distance metric and therefore the feature scaling which will be used with k-NN normally. The results of those experiments show the following:

- The performance of the *k*-NN algorithm depends significantly on feature scaling and distance metric used, the results show the massive gap between the performance of far metrics. For instance, we have found that the Canberra distance metric is performing best when applied on most datasets instead of mostly used Euclidean distance metric.
- 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 neighbours will be the same if all distances were multiplied or divided by the same constant.
- We get the same performance of Robust feature scaling on every dataset instead of a Min-Max or Standard Scaler. Thus it is considered a decent choice over traditionally used standard Scaler.
- There was no optimal distance metric, that may be used for all kinds of datasets because the results show that each of the dataset favors a particular distance metric. However one can start classification with the Canberra distance metric because it outperforms the commonly used Euclidean distance.

5.1 Our work has subsequent limitations

- Although we have tested a large number of distance metrics still there are many distance metrics available within the machine learning area that require to be tested and evaluated further for optimal performance.
- The 3 datasets we have used belong to medical domain which 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 a larger number of datasets with varied data types and should be from different domains.
- We have got reviewed only one style of *k*-NN during this paper, other variants of *k*-NN such as [23,24,25] need to be investigated.
- 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.

5. References

- [1] Innovatemedtec, accessed 9 September 2020, <u>https://innovatemedtec.com</u>.
- [2] D.Hand, H. Manilla, P. Symth.: Principles of Data Mining. The MIT Press. (2001).
- [3] 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).
- [4] Hu, L.-Y., Huang, M.-W., Ke, S.-W., & Tsai, C.-F. (2016). The distance function effect on knearest neighbor classification for medical datasets. SpringerPlus, 5 (1), 1304.
- [5] Todeschini, R., Ballabio, D., & Consonni, V. (2015). Distances and other dissimilarity measures in chemometrics. Encyclopedia of Analytical Chemistry.
- [6] Todeschini, R., Consonni, V., Grisoni, F. G., & Ballabio, D. (2016). A new concept of higherorder similarity and the role of distance/similarity measures in local classification methods. Chemometrics and Intelligent Laboratory Systems, 157, 50–57.
- [7] 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.

- [8] 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.
- [9] Hassanat, A. B. (2014). Dimensionality invariant similarity measure. Journal of American Science, 10 (8), 221–26.
- [10] 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.
- [11] 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.
- [12] Hassanat, A. B. (2014). Dimensionality invariant similarity measure. Journal of American Science, 10 (8), 221–26.
- [13] Lindi, G. A. (2016). Development of face recognition system for use on the NAO robot. Stavanger University, Norway.
- [14] Euclid. (1956). The Thirteen Books of Euclid's Elements. Courier Corporation
- [15] Deza, E., and Deza, M. M. (2009). Encyclopedia of distances. Springer.
- [16] Grabusts, P. (2011). The choice of metrics for clustering algorithms. Environment. Technology. Resources, 70-76.
- [17] Williams, W. T., & Lance, G. N. (1966). Computer programs for hierarchical polythetic classification (" similarity analyses "). The Computer, 9 (1), 60-64.
- [18] Lance, G. N., & Williams, W. T. (1967). Mixed-data classificatory programs 1 Agglomerative systems. Australian Computer Journal, 1 (1), 15-20
- [19] Akila, A., & Tairi, H. (2016). Combining Jaccard and Mahalanobis Cosine distance to enhance the face recognition rate. WSEAS Transactions on Signal Processing, 16, 171-178.
- [20] Wilcoxon, F. (1945). Individual comparisons by ranking methods. Biometrics Bulletin, 1 (6), 80-83.
- [21] Hassanat, A. B. (2018). Norm-Based Binary Search Trees for Speeding Up KNN Big Data Classification. Computers, 7(4), 54.
- [22] Hassanat, A. B. (2018). Two-point-based binary search trees for accelerating big data classification using KNN. PloS one, 13(11), e0207772.
- [23] Hassanat, A. B. (2018). Furthest-Pair-Based Decision Trees: Experimental Results on Big Data Classification. Information, 9(11), 284.
- [24] Hassanat, A. B. (2018). Furthest-pair-based binary search tree for speeding big data classification using k nearest neighbors. Big Data, 6(3), 225-235.