SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Research Paper Open Access

# Sustainable Implementation of Neuro-Fuzzy Systems Based Clustering Using Hybrid Learning Vector Quantization

Susmita Suresh Pagare[1], Tarun Madan Kanade[2], A. Shamila Ebenezer[3], P. William[4, 5*], Snehal Ashish Patil[6]

[1]Research Scholar, Symbiosis Institute of Operations Management, Symbiosis International (Deemed University), India.
E-mail: susmita.jrf@siom.in
[2]Symbiosis Institute of Operations Management, Symbiosis International (Deemed University), India. E-mail: tarun.kanade@siom.in
[3]School of Computer Science and Technology, Karunya Institute of Technology and Sciences (Deemed University), Coimbatore, India.
E-mail: shamila_cse@karunya.edu
[4]School of Computer Science and Technology, Karunya Institute of Technology and Sciences (Deemed University), Coimbatore, India.
E-mail: william160891@gmail.com
[5]Centre for Research and Consultancy, UNITAR International University, Selangor, Malaysia
[6]Assistant Professor, MIT ADT UNIVERSITY, Pune, India. E-mail: rinkalpatil777@gmail.com
*Corresponding Author: P. William, E-mail: william160891@gmail.com

**Abstract**

In current years, key inventions in Neuro Inference System research have transformed the machine learning background from an engineering perception. Neuro Inference System is hominid ready information handling systems that are grown up extensively in the last thirty years. The proposed work uses the Hybridized Distance function along with different distance functions viz. Euclidean, Canberra and Chebyshev function in Learning Vector Quantization (LVQ) algorithm and observes the clustering results in Breast Cancer dataset. In this work, we use the Supervised Learning Vector Quantization technique on hybrid distance function which work in both supervised as well as unsupervised modes and it is a heuristic search technique based on the patents available. The data sets have been taken form Medical Science for providing learning and examining. It increases the effective finding of diseases by using various data classification methods. Thus, it modifies the Neuro Inference System algorithm to improve its performance. The experimental work is simulated with simulation software MATLAB 7. After a comparative analysis of the results, it observed that the Hybrid LVQ is more accurate as compared to Canberra and conventional Euclidean Distance. The proposed Hybrid LVQ performance is enhanced up to 90.6% from 87.1% of Canberra Distance. It shows that Canberra gives the minimum distance between points from centroid as compared to Euclidean and Chebyshev Distance Function. The proposed Distance Function method gives the minimum distance of points from centroid as compared to Canberra Distance Function.

**Keywords:** Machine Learning, Learning Vector Quantization, Artificial Intelligence, Chebyshey Function

## 1. Introduction

Artificial Neural Network is a type of neural network which challenges to intimate the way hominid brain works [1]. Neural Network aims to get the human ability to adapt the change quickly and behave based on changes of data. The data may be result of researchers' efforts from market analysis and patient analysis. Neural network [2,3] utilizes only raw data to predict the result of the problem like diagnosis report of patient, history of past sales and prices in market. ANNs is made up of interconnections between neurons called network architecture [4]. The training of an architecture aims to adjust the system and respond which is based on input provided to system. The architecture of network describes that how nodes are connected to each other and responsible for communication among nodes. Training is provided to the system for the setting of weight on them. The node with higher weight has more value and node with low weight is less valuable. These weights are responsible for finding the similarity and dissimilarity. The nodes acclimatize the pattern and find a match for it. The activation function is amenable for the output of neuron.

In recent years, key inventions in Neuro Inference System of research have transformed the machine learning background from an engineering perception. Neuro Inference System is hominid ready information handling systems that are grown up extensively in the last thirty years. And it is important to understand the combining results of machine learning algorithms which could improve the training by using various

tools for actions. So, the proposed work uses the Hybridized Distance function along with different distance functions viz. Euclidean, Canberra and Chebyshev function in Learning Vector Quantization (LVQ) algorithm and observes the clustering results in Breast Cancer dataset. In this work, we use the Supervised Learning Vector Quantization technique on hybrid distance function which work in both supervised as well as unsupervised modes and it is a heuristic search technique. The data sets have been taken form Medical Science for providing learning and examining. Thus, it modifies the Neuro Inference System algorithm to improve its performance. The experimental work is simulated with simulation software MATLAB 7. After a comparative analysis of the results, it observed that the Hybrid LVQ is more accurate as compared to Canberra and conventional Euclidean Distance. The proposed Hybrid LVQ performance is enhanced up to 90.6% from 87.1% of Canberra Distance. It shows that Canberra gives the minimum distance between points from centroid as compared to Euclidean and Chebyshev Distance Function.
The further section of the article is categorized as provided: Section 2 discusses the literary works associated with this study. Section 3 explains the flow of the suggested work with the problem statement. Section 4 analyses the behavior of our suggested work. Furthermore, at last section 5 is all about conclusion and future work.

### 1.1 Network Architecture

The network architecture of ANN can be categorized as follows:

**Feed Forward Network:** As the name suggests, this type of network has only forward flow of information in the network layer. The artificial neurons can be arranged in a multi-layered manner having input, hidden and output layers [5]. A weight value is associated between these neurons to interconnections. The neurons layer shown in Figure 1 is responsible for taking the input from predecessor layer and converting it into output for the successor layer is named as Hidden Layer.

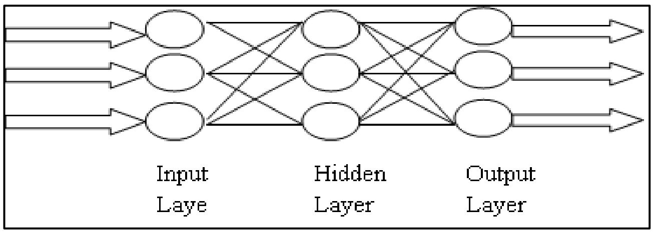


**Figure 1. Structure of Feed Forward Neural Network**

**Competitive Network**: Neurons, which are close to each other, represent similarity in this type of network [6,7]. Output nodes(y) compete with other output nodes and represent input patterns. It is not necessary that each output layer should be completely connected to each other. They may be connected to the nearest node. This type of network is used to match and predict touch, smell, voice, vision etc. Figure 2 shows the architecture of competitive networks.

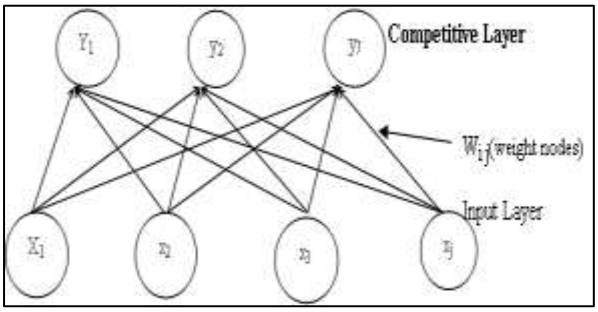


**Figure 2. Competitive Networks**

**Recurrent Network:** Each input and output unit are connected to all other units. Each unit works as input and output and the flow of information is sequential [8]. Connections between units form a directed cycle. Figure 3 shows the architecture of recurrent networks.

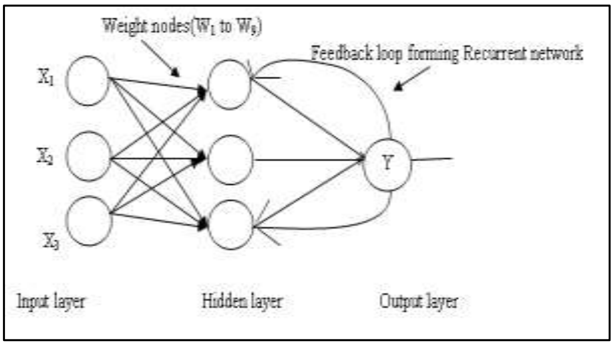


**Figure 3. Recurrent Networks**

**Activation Function:** It defines the output of the input provided to that node [9]. It is applied with input signal to obtain response.
**Weight:** It refers to the strength or amplitude of a connection between two nodes. It is denoted by w.
**Sigmoidal Function:** Sigmoid function is also called squashing function. It is used in neural network to give real-valued output based on logistic neuron which is a smooth and bounded function of their total output. This function is formulated as below equation.

$$\frac{1}{1+e^{-x}} \tag{1}$$

Its' output boundaries lie between (0, 1).
**Calculation of net input** using Matrix Multiplication Method.
**Threshold:** The beginning level afterward neuron is made to fires in network structure.

## 2. Literature Review

In this Section, various topics and research papers have been reviewed which are related to this work. First, descriptive learning models for neural networks (NN), historical developments in neural network (NN) learning, neural network learning algorithms and their tasks, data mining techniques using ANNs used to improve accuracy, and data relevance.
In addition, we have reviewed the work implemented in the algorithm for research purposes. Next, we explored the use of distance in distance searches. Finally, the ANN algorithm for pattern classification was checked.

### 2.1. Fundamental Models of ANN

The development of neural networks began with the development of the most famous model, the McCulloch-Pitts model [10]. Learning in network systems was first introduced by Hebb, known as the "Hebbian Learning Rule", and these models were considered the most dominant models.

**Mc-Cullous-Pitts Neuron Model:**

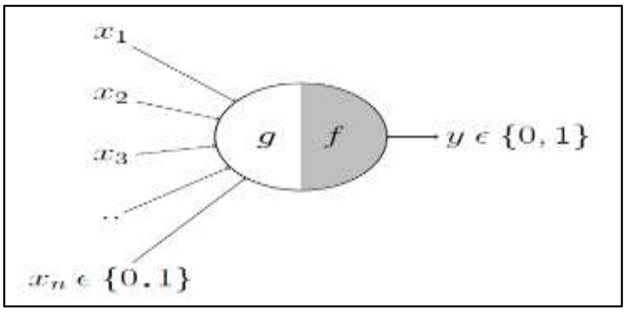


**Figure 4. Diagram showing architecture of McCulloch-Pits Neuron**

The most popular and widely used neuron model as per the work of McCulloch and Pitts, shown in Figure 4. Binary 0 and 1 states are possible in this model. Neurons are assigned by weighted path. This pathway can be excitatory or inhibitory. Excitatory coupling pathways are those with positive values and inhibitory pathways are those with negative values. The neurons in this model are linked to thresholds. In the diagram above, Y is a McCulloch-Pitts neuron that accepts signals from other neurons. A neuron with connection weight W is excitatory, and the connection weights of neurons from $x_n$ to $x_{n+m}$ are inhibitory, denoted by w. In this case Y has an activation function

$$\mathbf{F(y)}= \begin{cases} \mathbf{1\ if\ y\ \geq\ \Omega} \\ \mathbf{0\ if\ y \leq \Omega} \end{cases} \tag{2}$$

Where Ω is threshold value .

2.2. **Transfer Function**

The output is decided by the internal activation of neuron. Transfer function expresses the relationship in order to satisfy some problem specialization which is meant for the neuron to solve. Following Table 1 shows the most commonly transfer functions [13].

**Table 1. Summary of Different Versions of LVQ**

| Implementations of the LVQ | Modification |
|---|---|
| LVQ 1 | A single BMV (best matching vector) is selected and relocated nearer or more away from each data vector if class labels of selected vector matches. |
| OLVQ1 (Optimized LVQ1) | This version is identical to LVQ1, in addition with every codebook vector has its own learning rate |
| LVQ 2.1 | Two BMV's are selected arbitrarily and only updated if one belongs to the looked-for class and the distance fraction is inside a defined window |
| LVQ 3 | LVQ 3 is extension to LVQ 2.1 but matched vectors are adjusted by an epsilon value (adjusted learning rate as an alternative of the global learning rate) |
| OLVQ (Optimized LVQ3 | This version of algorithm is combination of LVQ 3 and OLVQ1 as every codebook vector has its learning rate |
| Multi-Pass LVQ | This model is proposed by using all above versions of the algorithm (Kohonen) where first pass is made on the model using OLVQ1, then a long fine tune pass is made with any of LVQ1, LVQ2.1 or LVQ3 |
| Hierarchal LVQ | This is concept where basic LVQ model is built and each codebook vector is considered as a centroid of group. All codebook vectors are estimated and out of them few are selected as candidates for sub-models. |

### 2.3. Performance Evaluation

The performance can be assessed by estimating its accurateness and efficiency for future prediction. The simplest method in the series is holdout, which aims to make a training set and test set (also known as holdout set) [14]. It aims to minimize the bias linked with training and holdout data. Here, we primarily discuss key Performance Evaluation Techniques.

### K-Fold Cross Validation

Whole data operations are done k times. The overall accuracy of a system is calculated by averaging the k individual measures.

$$\textbf{Accuracy} = \frac{1}{k}\sum_{i=1}^{k} A_i \qquad (3)$$

Where k is a number of splits or folds used and A is accuracy.

**Confusion Matrix**

The concept of a Confusion matrix was given by Kohavi and Provost in 1998, which describes the knowledge regarding actual and predicted values of classifications and evaluation of systems. This is done by using the data exist in the matrix. Table 2 Represent the Confusion matrix as follow.

**Table 2. Confusion Matrix**

| Predicted/Actual | Negative | Positive |
|---|---|---|
| Negative | P(TP) | Q(FN) |
| Positive | R(FP) | S(TN) |

Terminologies for the two-class matrix:

**Accuracy (AC):** Accuracy can be calculated as total number of cases that prediction is correct.

$$\textbf{AC} = \frac{p+s}{p+q+r+s} \qquad (4)$$

**True Positive (TP): The** cases from input data in which prediction is positive and actual result is also positive.

$$\textbf{TP} = \frac{s}{r+s} \qquad (5)$$

**False Positive (FP): The** cases in which prediction is positive but does not belong to positive case

$$\textbf{FP} = \frac{q}{p+q} \qquad (6)$$

**True Negative (TN):** The cases from input data in which prediction is negative and actual outcome is also negative.

$$\textbf{TN} = \frac{p}{p+q} \qquad (7)$$

**False Negative (FN):** Cases in which prediction is negative but they actually belong to positive case.

$$\textbf{FN} = \frac{r}{r+s} \qquad (8)$$

**Precision (P)**: The Positive **Predictive Value** or Precision is the proportion of the positive cases predicted which were correct.

$$\textbf{P} = \frac{s}{q+s} \qquad (9)$$

At initial level, the arrangement is made by the predefined topology function in physical positions. Calculations are done based on the distance from their positions with the predefined distance measuring. This map detects a wining neuron i* and weight [15]. Updating is performed by using recognizing neighbor neurons within a definite neighborhood sector Ni*(d). All such neurons I Ni*(d), are specifically adjusted as follows:

$$\mathbf{w(q) = w\,(q\text{-}1) + \alpha\,[p(q) -_{I} w\,(q\text{-}1)\quad ]} \qquad (10)$$

$$\mathbf{w(q) = (1\text{-}\ \alpha_{I}\, w\,(q\text{-}1) + \alpha\, p(q))} \qquad (11)$$

The Neighbourhood neuron $N_{i^*}$ (d) possesses the indices for all of the neurons which lie within a radius d of the winning neuron i*

$$N_i(d) = \{j, d_{ij} \leq d\} \quad (12)$$

Therefore, when a vector p is offered, the weights are updated for winning neuron and it deviates towards p. After much repetitive iteration, the neighbouring neurons contain learned vectors. Data to neural network is fed by Batch algorithm prior to any update occurs to the weights. Algorithm determines the winning neuron of iteration. Now, weight vector of winning neuron moves towards the average location of input vectors' set.

### 2.4. Self Organizing Map Architecture

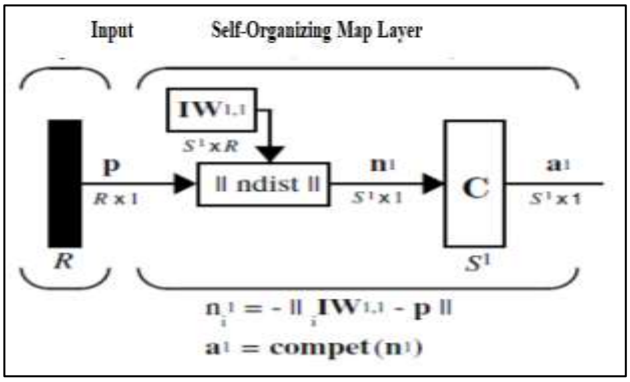


**Figure 5. Self-Organizing Map Architecture**

Here, the architecture in Figure 5 resembles unbiased competitive learning. Competing transfer functions produce the value 1 for each output element ai 1 for the winning neuron i*. Any remaining output elements present in a1 are now evaluated at 0. As the neurons mentioned above are close to the winner neuron which is modified together with the winner neuron [16]. Different topologies the of neuron can be selected and apply different distance formulas to identify which neurons are closer to the winning neuron.

### 2.5. Learning Vector Quantization (LVQ)

LVQ is a technique for statistical data clustering that can highlight the quality of a classifier's decision region by slightly shifting the class information towards a "Voronoi vector". LVQ transforms a large set of input vectors. This is a modified version of Self-Organizing Maps (SOM). This method by which LVQ works is described below. First, an input vector x is randomly selected from the input space [17,18]. If the class label belongs to the input vector x and the Voronoi vector w matches, the Voronoi vector w is shifted towards the input vector of x, and if the class label does not belong to the input vector, the Voronoi vector leaves. From what is shown in Figure 6, the x-shifted input vector is shown.

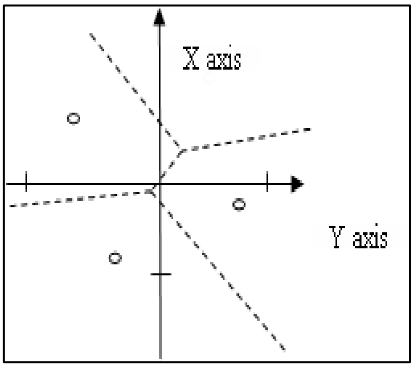


**Figure 6. Voronoi Space Involving Four Cells**

It provides a good approximation method for computing the Voronoi vectors by the synaptic weight vectors which are approximated shown in Figure 7. The first stage is the computation of the feature map for adaptive pattern classification and the second stage for fine tuning of the class boundaries [19].

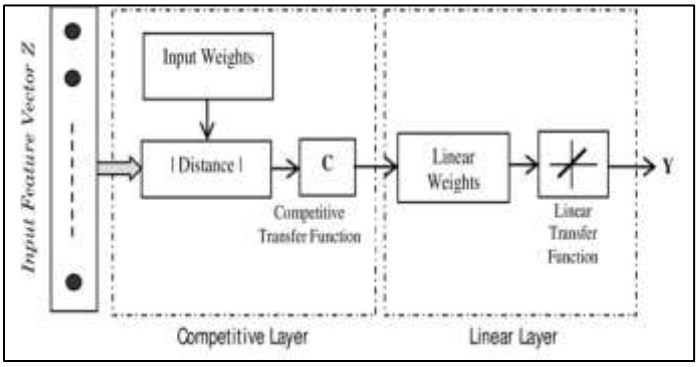


**Figure 7. Adaptive Pattern Classification using SOM and LVQ**

### 2.6. Architecture and Operations

The LVQ architecture networks is related to Kohonen Self-Organizing Neural Networks and Competitive Learning Networks. Topological order is not preserved. In LVQ networks, known class is there for each output unit [20]. Training begins with the design of codebook vectors. Figure 8 below shows the architecture of a two-layer LVQ neural net. The first layer, called the competition layer, allows us to extract the classifier and classify the input vector. The second layer, called the linear layer, maps classes to their respective user-defined target classifications.

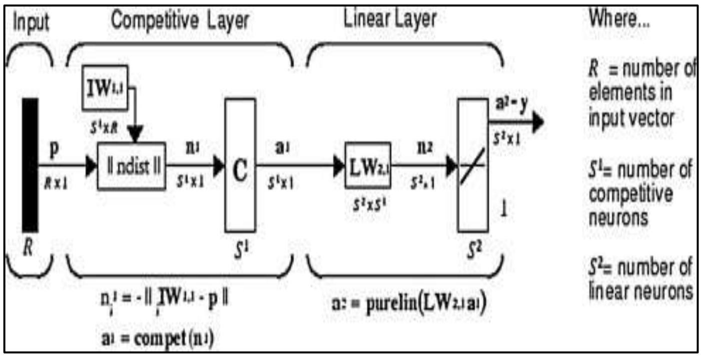


**Figure 8. Architecture of Learning Vector Quantization Networks**

**The steps for initialization of reference vectors are:**

Select first 'm' training vectors and use them as weight vectors. The rest of the vectors are used for the purpose of training. Now, weight and random initial are assigned to the reference vectors and class respectively.

### 2.7. Supplemental LVQ

**LVQ 2 Learning Rule:** This variant of LVQ is based on heuristic training scheme which goals to increase efficiency of an algorithm in respect to convergence rate and better approximation of Bayesian decision boundaries [21]. The basic idea behind Bayesian decision boundaries is to choose the class which has the smallest expected loss. The classification is used in this is identical to LVQ1. At first, in the Learning phase of an algorithm $m_p$ and $m_q$ are two codebook vectors and close neighbors of x and update their weight simultaneously. Out of them $m_p$ must be in the same class and the other $m_q$ to a different class, respectively, where x resides around the midplane of $m_i$ and $m_j$. Distance of x from codebook vectors $m_p$ and $m_q$ is calculated and analyzed by the distance formula name Euclidean Distance [22]. The midplane, where x resides, known as 'Window' of relative width w if

$$\min\left(\frac{d_i}{d_j}, \frac{d_j}{d_i}\right) > s, \text{ where } s = \frac{(1-w)}{(1+w)} \qquad (13)$$

In general, the relative width of midplane calls 'window' w varies from 0.2 to 0.3. Then, equation of the algorithm:

$$m_p\ (t+1) = m_p\ (t) - \alpha(t)\ [x(t) - m_i\ (t)] \qquad (14)$$

$$\mathbf{m_q\ (t + 1) = mq(t) + \alpha(t)\ [x(t) - m_j\ (t)]} \quad (15)$$

Where $m_p$ and $m_q$ are the two closest codebook vectors to x, where $m_q$ and x are of the same class, while $m_p$ and x are of different classes respectively.

**LVQ 3 Learning Rule:** The LVQ3 algorithm was introduced to ensure the shifting of vector to some large distance that the $m_i$ continues to approximate the class distributions.

$$\mathbf{m_k\ (t + 1) = m_k\ (t) + epsilon\ (t)\ [x(t) - m_k\ (t)]} \quad (16)$$

For k in {i,j}, if x, mi, and mj belong to the same class. Epsilon range may vary between 0.1 and 0.5. The adjusted learning rate is the Epsilon range. The version of algorithm has a feature to stabilize itself in continual learning.

As various topics have been reviewed based on research which are related to this work. After reviewing, it is clear that the performance of Canberra distance function is limited to a dataset which is closer to zero. Hence, there are limitations observed in Canberra distance function which is further discussed in problem statement. The framework of future networks must be dependent on hybrid techniques, which integrate various networks with the internet backbone. The existing works indicate that a huge number of retransmission of data happens. The data is retransferred over a similar path that causes receiver buffer blocking.

## 3. Problem Formulation and Methodology

### 3.1. Problem Statement

The performance of Canberra distance function is limited to a dataset which is closer to zero. Hence, there are two limitations observed in Canberra distance function.

1. Use of Canberra distance is limited to datasets having the characteristic similar to features of Canberra distance measure.
2. It uses the only categorical data and there can be continuous data in some scenarios.
3. The simulation can be implemented by other data mining techniques in order to obtain better diagnostic outcomes.

### 3.2. Methodology

In this dissertation, work on LVQ algorithm by applying different distance functions and analyses of their accuracy has been focused. The second part of this thesis is the application of Hybridization with the most widely used classification techniques used and that is the k-means algorithm. The first part of the proposed work uses medical dataset from UC Irvine Machine Learning Repository (UCI). The second part of the proposed work employs random data generated for pattern classification. One of the most important issues in using the conventional.

Euclidean Distance is below described.

a) Euclidean may not perform well if data is at origin.
b) Canberra distance works only if data is at origin.

Thus, we propose Hybrid methods as possible substitutes for the conventional Euclidean distance function in pattern classification with LVQ.

The main objective of this methodology is to analyses the accuracy of Hybrid method with LVQ and k-means in Clustering problems, and evaluate them under specific conditions.

### 3.3. Distance Function and its Uses

The selction of distance function plays an significant role in Categorization. The motive of any Categorization function is to contain integration of similar object from a group of homogeneous objects. The Categorization finds out objects which have the shortest distance from each other. [23]. The choice of distance function is very difficult as there are many cases, where other distance function may give better outcomes of distance. So, it cannot be mentioned that distance function X is better than Y and distance function Z is more effective than There is another point to remember that while opting for distance function is there are continues and categorical both types of data. Distance measures are also depending on type of data. In

Learning Vector Quantization, To measure the distance between the weighted vector and input vector, Euclidean distance function is used.

### 3.4. Design of Hybrid Distance Function

In this work, we have proposed a hybrid function that is used in LVQ algorithm. This function is developed by combining various other functions. In the literature, it is found that the existing LVQ algorithm obtain the distance between input and weight vector by using below mentioned functions.

**Canberra Distance Function:**

This function is used whenever the data is present near to origin and we want to find the distance between two points. When the canberra distance function has rectangular function then it returns true. Otherwise it returns false as output [24,25].

$$\mathbf{D}_{\text{Canberra}}\,(\mathbf{p}, \mathbf{q}) = \sum_{i=0}^{n} \frac{|p_i - q_i|}{|p_i| + |q_i|} \tag{17}$$

Where, p and q are points. The Canberra distance itself not lies between 0 and 1. The term become unity if one of the coordinates is 0 and distance will not be affected. When both coordinates are 0, we need to be states as 0. The Canberra distance is extremely sensitive [26] to a minor modification when both of the coordinate is close to 0.

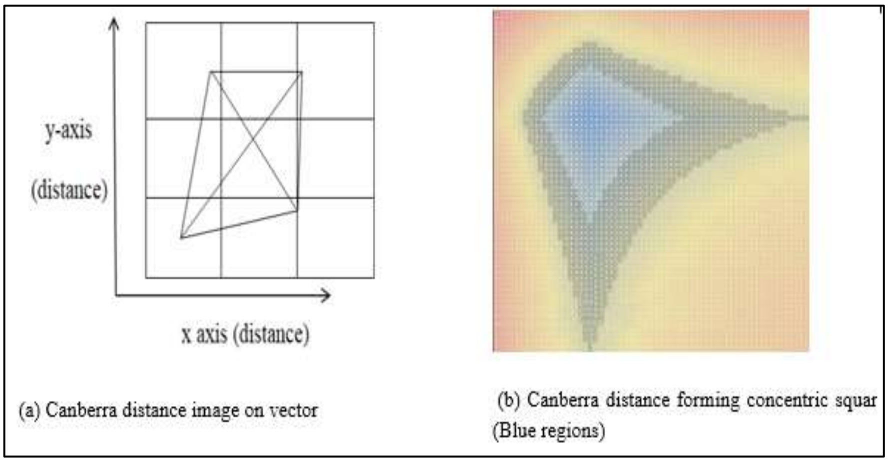


**Figure 9. Canberra Distance Function**

In this figure 9 (a), the pattern on the vector space is almost tetra gram with lines nearer to each other at two ends. These lines never cross to each other and their coordinates are zero. These lines are not parallel to each other. This figure is drawn with sink at middle point. The core of blue section in this figure denotes the quadrilateral pattern of similar data. The figure shows that the Canberra distance makes a quadrilateral shape for similar data on vector whereas Euclidean distance [27] makes concentric circles around the center.

**Euclidean Distance Function:**

The function used for identifying distance between two points which are lie on a straight line [28]. The formula for Euclidean distance shown in figure 10 is as follows:

$$\mathbf{E}_{\text{euclidean}}\,(\mathbf{p},\mathbf{q}) = \mathbf{d}(\mathbf{q},\mathbf{p}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \tag{18}$$

Where, p and q are points in Euclidean space.

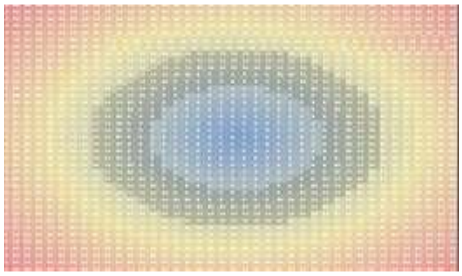

**Figure 10. Euclidean Distance Function**

**Chebyshev Distance Function:**
This function calculates the distance by considering the absolute difference between coordinates of two points. The distance is in concentric squares around the center only when the sink is on the center. Figure 11(a) shows square pattern on vector whereas figure 11 (b) showing square patterns in blue regions plotted in 2-dimensional region[29].

$$\mathbf{D}_{\text{Chebyshev}}\,(\mathbf{p},\mathbf{q}) = \max_{i}(|\mathbf{p_i} - \mathbf{q_i}|) \qquad (19)$$

Where, p and q are vector space or called as points.

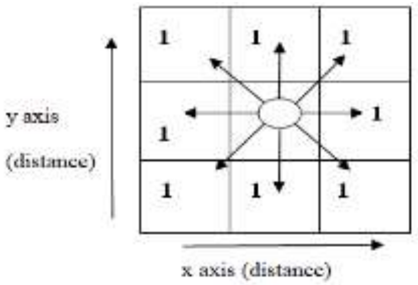


**Figure 11. (a) Image on Vector**

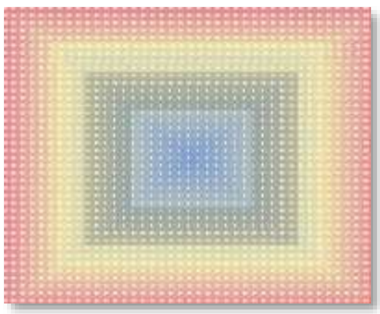

**Figure 11. (b) Chebyshev Distance Forming Concentric Squares (Blue Regions)**

We have a coordinate value of two objects denoted with O say A and B and features denoted with F1 to F5 as shown in table 3 below the object that doesn't possess features is dictated as (Not Available) NA.

**Table 3. Objects with Features**

| F/O | F1 | F2 | F3 | F4 | F5 |
|---|---|---|---|---|---|
| A | 0 | 3 | 4 | 5 | NA |
| B | 7 | 6 | 3 | 1 | NA |

Both objects A and B have 4 features.

Now, the Euclidean distance ($D_{ab}$) between object A and object B is

$D_{ab} = \sqrt{(0-7)2 + (3-6)2 + (4-3)2 + (5+1)2}$

$= \sqrt{49 + 9 + 1 + 36} = 9.747$

Chebyshev Distance ($D_{ab}$) between object A and object B is $D_{ab}$ =max {|0-7|, |3-6|, |4-3|, |5+1|} =max {7, 3, 1, 6} = 7

Canberra Distance ($D_{ab}$) between object A and object B is

$D_{ab} = \frac{0-7}{0+7} + \frac{3-6}{3+6} + \frac{4-3}{4+3} + \frac{5+1}{5+1} = 2.467$

### 3.5. Proposed Hybrid Model

In the hybrid model, we have proposed a distance function by combining the Canberra[30], Chebyshev and Euclidean distance function. This function finds distance as follows:

$$\mathbf{D_{max1} = max\ (d_{Canberra},\ d_{Chebyshev})} \quad (20)$$

$$\mathbf{D_{max2} = max\ (d_{max1},\ d_{Euclidean})} \quad (21)$$

$$\mathbf{D_{Hybrid} = max\ (d_{max2})} \quad (22)$$

The above equation gives the Hybrid Distance formula.

$D_{max1}$= max (9.747, 7) = 7

$D_{max2}$= max (7, 2.467) = 2.467

$D_{min2}$= max (7, 2.647) = 2.46

## 4. Experimental Analysis

As discussed in Literature Survey, MATLAB's Neural network Clustering Tool has been used to apply the Batch Unsupervised weights SOM algorithm [31] (trainbuwb) over the Cancer dataset (stored in file cancer.txt), which contains 768 samples with 9 columns. The network was trained with 200 epochs / iterations. Various plottings using MATLAB-NCTOOL for cancer Dataset are shown in figure 12, figure 13, figure 14 and figure 15 respectively.

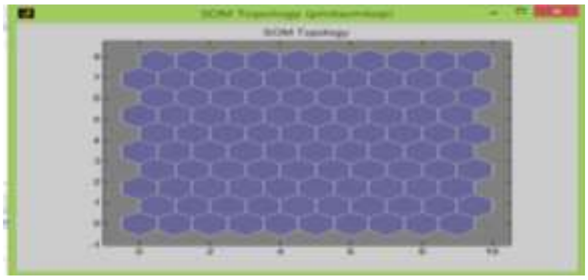


**Figure 12. SOM Topology in MATLAB-NCTOOL for cancer Dataset**

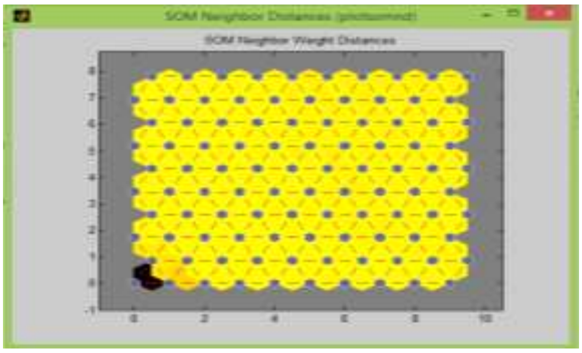


**Figure 13. SOM Weight Connections in NCTOOL for cancer Dataset**

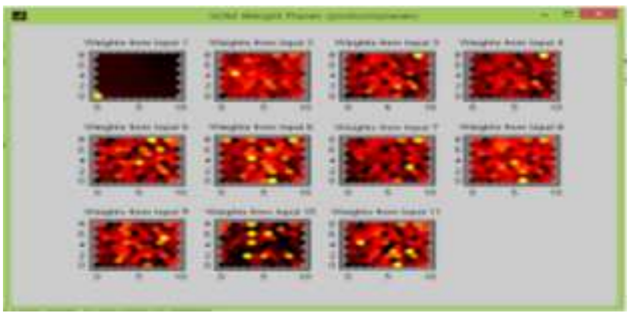

**Figure 14. SOM Weight Planes in MATLAB-NCTOOL for cancer Dataset**

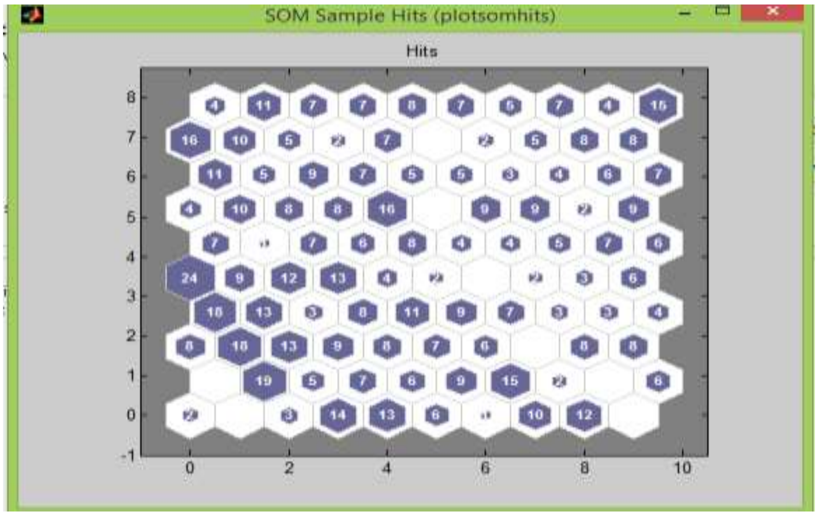


**Figure 15. SOM Sample Hits in MATLAB-NCTOOL for cancer Dataset**

### 4.1. Simulation of Work with Cancer Dataset

Here, we simulate and analyze the performance of algorithm with conventional Euclidean, Canberra, Chebyshev [32] and Hybrid method as proposed method. A Confusion matrix and ROC curve have been plotted for algorithm with each distance function. Various ROC Curve plotting with different distance functions are shown in figure 16, figure 17, figure 18 and figure 19 respectively. Confusion Matrix for aapplication of LVQ with Euclidean Distance computed in Table 4.

**Table 4. Confusion Matrix for Euclidean Distance with Cancer Dataset**

| PC/AC | Generated confusion matrix with cancer dataset for Canberra distance | | |
|---|---|---|---|
| | Negative | Positive | Rates |
| Negative | True Negative (TN) 225(35.8%) | False Positive (FP) 0(0.0%) | 100%(TNR) 0.0%(FPR) |
| Positive | False Negative (FN) 2 (0.3%) | True Positive (TP) ,402(63.9%) | 99.5%, 0.5% |
| Rates | Negative Predictive Value 99.1%,0.9% | Precision 100%,0.0% | Accuracy 99.68% ,0.32% |

As per Confusion Matrix plot retrieved for Euclidean Distance function is observed in the below mentioned data:

Negative Cases: 212 (212+0) out of 629 (33.7%)

a) TNR = TN / TN + FP = 212/212which is100%.

b) FPR = FP / TN + FP = 0/212 = 0.0%.

Positive Cases: 614(212+402) cases of 629 (97. 3%)

a) FNR = FN / FN + TP = 15/417 = 3.59%

b) TPR = TP / TP + FN = 402/417= 96.40% ~ 96 %

Precision (P) =TP / FP + TP = 402/ (0+402) = 100%

Accuracy (AC) = TN+TP/TN+FP+FN+TP

= (212+402) / (212+0+15+402)

= 614/629= 97.6% ~ 97.6%.

Here TPR rise up to 0.96.

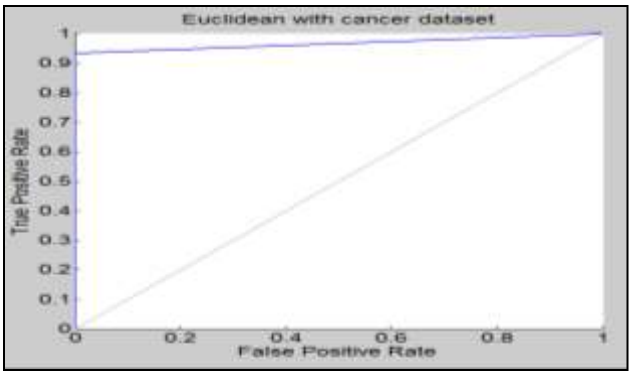


**Figure 16. Euclidean Distance ROC curve on cancer Dataset Confusion Matrix for application of LVQ with Canbera Distance on Cancer Dataset are show in Table 5.**

**Table 5. Confusion Matrix for Canberra Distance with Cancer Dataset**

| PC/AC | Generated confusion matrix with cancer dataset for Canberra distance | | |
|---|---|---|---|
| | Negative | Positive | Rates |
| Nega-tive | True Negative (TN) 225(35.8%) | False Positive (FP) 0(0.0%) | 100%(TNR) 0.0%(FPR) |
| Posi-tive | False Negative (FN) 2 (0.3%) | True Positive (TP) 402(63.9%) | 99.5% 0.5% |
| Rates | Negative Predictive Value 99.1%,0.9% | Precision 100%,0.0% | Accuracy 99.68% ,0.32% |

As per Confusion Matrix plot retrieved for Canberra Distance function is observed in the below mentioned data:
1. Negative Cases: 225 (225+0) out of 629 (35.8%)
a) TNR = TN / TN + FP
= 225/225 which is 100%.
FPR = FP / TN + FP= 0/225 = 0.0%
Positive Cases: 627(225+402) cases of 629 (99.68%)
FNR = FN / FN + TP =2/404 = 0.4%
TPR = TP / FN + TP = 402/404= 99.50 ~ 99.50 %
Precision (P) = TP/TP+FP = 402/ (402+0) = 100%
Accuracy (AC) = TN+TP/TN+FP+FN+TP
= (225+402) / (225+0+2+402) = 627/629
= 99.68% ~ 99.7%

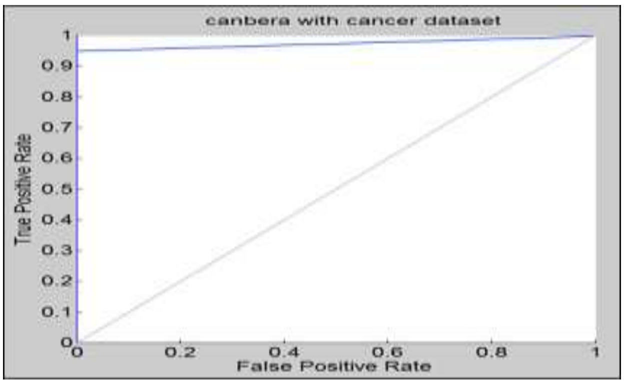


**Figure 17. Canberra Distance ROC curve on cancer Dataset Confusion Matrix for application of LVQ with Hybrid Distance (Proposed Model) as shown in Table 6.**

**Table 6. Confusion Matrix for Hybrid Distance with Cancer Dataset**

| PC/AC | Generated confusion matrix with cancer dataset for Hybrid distance | | |
|---|---|---|---|
| | Negative | Positive | Rates |
| **Negative** | True Negative (TN) (226)35.9% | False Positive (FP) 0(0.0%) | 100 % (TNR) 0.0%(FPR) |
| **Positive** | False Negative (FN) 1(0.2%) | True Positive (TP) (402) 64.1 % | 99.75%(TPR) 0.25 (FNR) |
| **Rates** | Negative Predictive Value 99.3%, 0.7% | Precision 100%,0.0% | Accuracy 99.84%,0.16 % |

As per Confusion Matrix plot retrieved for Hybrid Distance function, below mentioned parameter values are observed: First Observation for the function resultant confusion matrix[33] gives result equivalent to Canberra Distance Function. From second observation onwards obtained confusion matrix as shown above. Calculation is descripted based on above matrix below

Negative Cases: 226 out of 629 = 35.9

TNR = TN/TN+FP= 100%.

FPR = FP/FP+TN = 0.0%

Positive Cases: 628(402+226) out of 629 cases (99.84%)

FNR = FN/FN+TP =1/403 = 0. 2%

TPR = TP / FN + TP = 402/1+402 = 99.75%

Precision (P) = TP/TP+FP= 100%

Accuracy (AC) = TN+FP/TN+FP+FN+TP

= (628) / (629) =99.84%

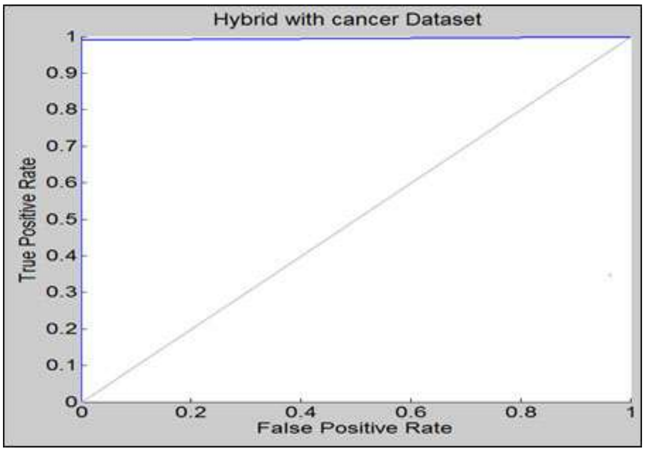


**Figure 18. Hybrid Distance ROC curve on cancer Dataset**

**Table 7. Confusion Matrix for Hybrid Distance with Cancer Dataset**

| PC/AC | Confusion matrix plot for Chebyshev distance on cancer dataset | | |
|---|---|---|---|
| | **Negative** | **Positive** | **Rates** |
| **Negative** | True Negative (TN) | False Positive (FP) | 100%(TNR) |
| **Positive** | False Negative (FN) 24 (3.8%) | True Positive (TP) 402(63.9%) | 94.6% 5.6% |

| | | | |
|---|---|---|---|
| **Rates** | Negative Predictive Value 89.4.0%,10.6% | Precision 100%,0.0% | Accuracy 96.2%, 3.8% |

As per Confusion Matrix plot retrieved for Chebyshev Distance function [34] is shown in table 7 as per below mentioned data:
Negative Cases: 203 (203+0) out of 629 (32.27%)
True Positive Rate (TPR) = TN/TN+FP
= 203/203 which is 100%.
False Positive Rate (FPR) = FP/FP+TN= 0/203 = 0.0 %.
Positive Cases: 605(203+402) cases of 629 (96.18%)
FNR =FN/FN+TP = 24/426 = 5.63%
TPR = TP/ FN+TP= 402/426= 94.36 ~ 94 %
Precision (P) = TP/ FP+TP = 402/ (0+402) = 100%
Accuracy (AC) = TN+TP/TN+FP+FN+TP
= (203+402) / (203+0+24+402) = 605/629
= 96.18% ~ 96%

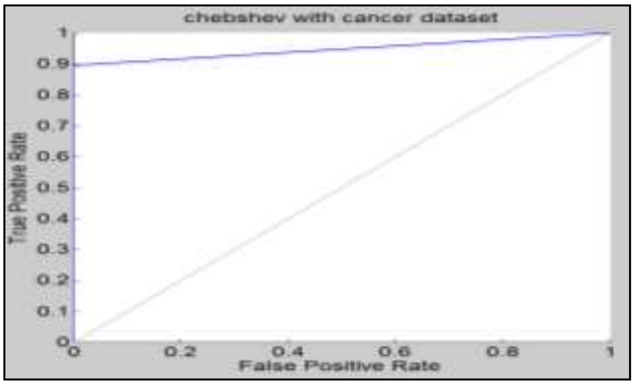


**Figure 19. Chebyshev Distance ROC curve between TPR and FPR on cancer Dataset**

### 4.2. Hybridization over k-means algorithm:

Here, we apply hybrid on various other different metrics and calculate centroid between each applied distance function and hybrid distance function, which replaces the total squared distance function [35]. The outcome distance, which has been shown, is the distance retrieved from the hybrid function and it is the shortest one as compares to Euclidean, Canberra, Chebyshev [36,37]. The proposed work gives better results and a better way than existing methods of single function result. Figure 20 shows the cluster distribution using K-Means.

**Algorithm: -**

Notations: - D-distance calculated by matrices.
Input: V1, V2 – Point between these two the distance is calculated.
1. Calculate distance with each distance metrics among V1 and V2.
2. Calculate minimum of d among all distance matrices.

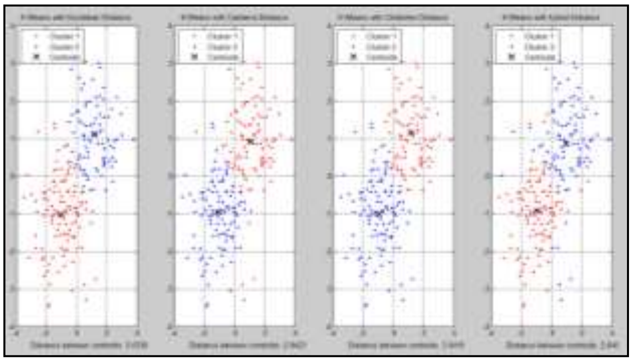

**Figure 20. Cluster Distribution using K-Means**

### 4.3. Outcome of Analysis

It is concluded from the experimental study that hybridization of distance function over LVQ gives improved results as compared to usage of conventional Euclidean distance over LVQ. In the above experiment and below table 8, it is noticed that hybrid distance method applied to k-Means algorithm produces crisp cluster and calculates minimum distance as compared to conventional K-Means algorithm. It is observed from the below calculation table that the distance from centroid is better in proposed modification over conventional K-Means and allows minimization of distances between cluster centers.

**Table 8. Comparison of Distance from Centroid over K-Means Algorithm**

| | Comparisons of Distance from Centroid | | | |
|---|---|---|---|---|
| | Euclidean | Canberra | Chebyshev | Hybrid |
| Centriod Distance | 3.0338 | 2.8423 | 2.9901 | 2.645 |

### 4.4. Improvement over Canberra:

Let a be the distance of centroid from Canberra distance (Minimum distance among Euclidean, Canberra and Chebyshev distance)

Let b be the distance of centroid from Hybrid distance

**Improvement =** $\frac{a-b}{a}$ (23)

= $\frac{2.8423-2.645}{2.8423}$*100 = 6.94 %

As per the analysis, centroid distance using the Hybrid distance technique is always less than the distance rescued by using Euclidean, Canberra and Chebyshev algorithm.

After the comparison of experimental results in the in this section, an increase by 6.94% in the overall accuracy of the algorithm is observed by providing the similar data set after execution of an algorithm, as shown in Table 9.

**Table 9. Comparison of Confusion Matrix Results Retrieve on Cancer Dataset**

| Parameter | Comparison of Distance function on Cancer Dataset | | | |
|---|---|---|---|---|
| | Euclidean | Canberra | Chebyshev | Hybrid |
| Negative Case | 33.7% | 35.7 % | 32.3% | 35.9 % |
| Positive Case | 97.6% | 99.68 % | 96.18 % | 99.84 % |
| False Negative Rate | 3.6% | 0.4% | 5.63% | 0.2% |
| True Positive Rate | 94.60% | 99.50 % | 94% | 99.75 % |
| Accuracy | 97.68 | 99.68 % | 98.1% | 99.84 % |

We have obtained the below mentioned results after investigation on basic K-Means Algorithm. As per result, distance from proposed algorithm is the shortest after comparing each individual applied distance function with method.

**4.5. Calculation Analysis:**
The shortest path among the centroids gets reduced by the use of the Hybrid distance algorithm in each case when compared to hybrid as shown in figure 21 and table-9, thereby indicating an improvement of maximum 6.94% by applying hybrid distance, which is better than existing Canberra distance function when applied on k-means algorithm. The proposed work gives better results and is a better technique than existing methodologies.

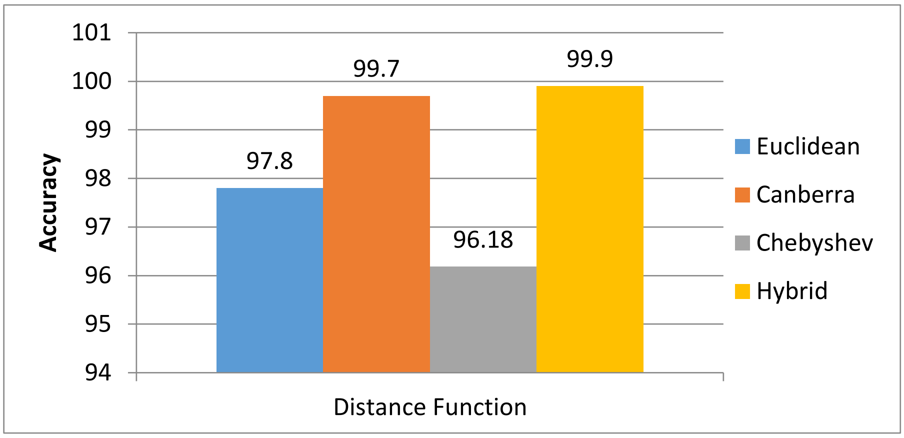


**Figure 21. Outcome of the Distance Functions on Two Different Datasets**

**4.6. Analysis Summary of Experimental result**
a) Application of Hybrid distance function over Cancer data-set: It is concluded from the above analysis that hybridization of LVQ algorithm for pattern classification gives more accurate result as compared to conventional distance function, as an increment of 2.16% was observed in the overall accuracy of LVQ algorithm.
b) Application of Hybrid distance function in K-Means algorithm: It is concluded from the above experimental work that the distance from centroid is better as compared to distance calculated from Canberra distance function the shortest path among the centroid gets reduced by applying hybrid distance function. An Increment of 6.94% was observed as overall accuracy of K-Means algorithm 5.

## 5. Conclusion

In this work, we have found that the proposed Hybrid LVQ performance is enhanced from 87.1% of Canberra Distance to 90.6%. In addition, we have found that Canberra gives minimum distance between points from centroid as compared to Euclidean and Chebyshev Distance Function. The proposed Distance Function gives minimum distance of points from centroid as compared to Canberra Distance. In this dissertation, the work on three distance function has been shown which uses for uniform data, origin data and grid, hence, it is concluded that the proposed hybridize technique proves more guaranteed result in all cases. Hence, this approach gives more accurate results on all distance functions if applies to their distance measuring parameter (continuous or categorical). The research carried out on K-Means algorithm results in the increasing of performance up to 6.94% as compared to Canberra. Therefore, it can be a better option to predict about the diseases like Cancer and Diabetes.

**6. Current and Future Development**
The contribution of this work is that a new solution of all distance function can be made apart from their application area (dimension, area) which gives more accurate result, because Canberra distance function proves best in cases where values taken for a data set are nearer to zero.
Therefore, its procedure is limited to the data sets only having categorical data. The continuous data requires in some diagnostic cases. Here, technique was used in the proposed work to test the sustainability of algorithm proposed with hybridization of distance function. Other techniques shall be implemented as a future work to obtain better diagnostic outcomes. Further, decreasing the time in hybrid function and hybridization of different mining technique shall be done, implementation based on certain

data Categorization and classification algorithms as per patterns is implemented using hybridization on different data mining techniques as discussed.

**Acknowledgement**

Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2025R435), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia

## References

1. S. Harifi, M. Khalilian, J. Mohammadzadeh and S. Ebrahimnejad, "Optimizing a Neuro-Fuzzy System Based on Nature-Inspired Emperor Penguins Colony Optimization Algorithm," in IEEE Transactions on Fuzzy Systems, vol. 28, no. 6, pp. 1110-1124, June 2020.doi: 10.1109/TFUZZ.2020.2984201
2. A. Rikalovic, I. Cosic, R. D. Labati and V. Piuri, "Intelligent Decision Support System for Industrial Site Classification: A GIS-Based Hierarchical Neuro-Fuzzy Approach," in IEEE Systems Journal, vol. 12, no. 3, pp. 2970-2981, Sept. 2018. doi: 10.1109/JSYST.2017.2697043
3. S. Feng and C. L. P. Chen, "Fuzzy Broad Learning System: A Novel Neuro-Fuzzy Model for Regression and Classification," in IEEE Transactions on Cybernetics, vol. 50, no. 2, pp. 414-424, Feb. 2020.doi: 10.1109/TCYB.2018.2857815
4. P. Gil, T. Oliveira and L. Palma, "Adaptive Neuro–Fuzzy Control for Discrete-Time Nonaffine Nonlinear Systems," in IEEE Transactions on Fuzzy Systems, vol. 27, no. 8, pp. 1602-1615, Aug. 2019.doi: 10.1109/TFUZZ.2018.2883540
5. S. Al-Dabooni and D. Wunsch, "Convergence of Recurrent Neuro-Fuzzy Value-Gradient Learning With and Without an Actor," in IEEE Transactions on Fuzzy Systems, vol. 28, no. 4, pp. 658-672, April 2020.doi: 10.1109/TFUZZ.2019.2912349
6. I. Abdulrahman and G. Radman, "Wide-Area-Based Adaptive Neuro-Fuzzy SVC Controller for Damping Interarea Oscillations," in Canadian Journal of Electrical and Computer Engineering, vol. 41, no. 3, pp. 133-144, Summer 2018. doi: 10.1109/CJECE.2018.2868754
7. M. M. Ferdaus, M. Pratama, S. G. Anavatti, M. A. Garratt and Y. Pan, "Generic Evolving Self-Organizing Neuro-Fuzzy Control of Bio-Inspired Unmanned Aerial Vehicles," in IEEE Transactions on Fuzzy Systems, vol. 28, no. 8, pp. 1542-1556, Aug. 2020.doi: 10.1109/TFUZZ.2019.2917808
8. Y. Liu, F. Liu and L. Li, "Strong Convergence of Neuro-Fuzzy Learning with Adaptive Momentum for Complex System," in IEEE Access, vol. 7, pp. 39362-39368, 2019.doi: 10.1109/ACCESS.2018.2883381
9. V. Sumati and C. Patvardhan, "Interval Type-2 Mutual Subsethood Fuzzy Neural Inference System (IT2MSFuNIS)," in IEEE Transactions on Fuzzy Systems, vol. 26, no. 1, pp. 203-215, Feb. 2018.doi: 10.1109/TFUZZ.2016.2646750
10. S. Song, B. Zhang, J. Xia and Z. Zhang, "Adaptive Backstepping Hybrid Fuzzy Sliding Mode Control for Uncertain Fractional-Order Nonlinear Systems Based on Finite-Time Scheme," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 50, no. 4, pp. 1559-1569, April 2020.doi: 10.1109/TSMC.2018.2877042
11. M. Ali, M. Adnan, M. Tariq and H. V. Poor, "Load Forecasting Through Estimated Parametrized Based Fuzzy Inference System in Smart Grids," in IEEE Transactions on Fuzzy Systems, vol. 29, no. 1, pp. 156-165, Jan. 2021.doi: 10.1109/TFUZZ.2020.2986982
12. S. Song, J. H. Park, B. Zhang and X. Song, "Observer-Based Adaptive Hybrid Fuzzy Resilient Control for Fractional-Order Nonlinear Systems With Time-Varying Delays and Actuator Failures," in IEEE Transactions on Fuzzy Systems, vol. 29, no. 3, pp. 471-485, March 2021.doi: 10.1109/TFUZZ.2019.2955051
13. R. S. T. Lee, "Chaotic Type-2 Transient-Fuzzy Deep Neuro-Oscillatory Network (CT2TFDNN) for Worldwide Financial Prediction," in IEEE Transactions on Fuzzy Systems, vol. 28, no. 4, pp. 731-745, April 2020.doi: 10.1109/TFUZZ.2019.2914642
14. J. Liu, C. Jiang, J. He, Z. Tang, Y. Xie and P. Xu, "STA-APSNFIS: STA-Optimized Adaptive Pre-Sparse Neuro-Fuzzy Inference System for Online Soft Sensor Modeling," in IEEE Access, vol. 8, pp. 104870-104883, 2020.doi: 10.1109/ACCESS.2020.2998792
15. J. P. de Moura, J. V. d. F. Neto and P. H. M. Rêgo, "A Neuro-Fuzzy Model for Online Optimal Tuning of PID Controllers in Industrial System Applications to the Mining Sector," in IEEE Transactions on Fuzzy Systems, vol. 28, no. 8, pp. 1864-1877, Aug. 2020.doi: 10.1109/TFUZZ.2019.2923963
16. T. N. Kapetanakis, I. O. Vardiambasis, E. I. Lourakis and A. Maras, "Applying Neuro-Fuzzy Soft Computing Techniques to the Circular Loop Antenna Radiation Problem," in IEEE Antennas and Wireless Propagation Letters, vol. 17, no. 9, pp. 1673-1676, Sept. 2018.doi: 10.1109/LAWP.2018.2862939

17. A. Jafarifarmand, M. A. Badamchizadeh, S. Khanmohammadi, M. A. Nazari and B. M. Tazehkand, "A New Self-Regulated Neuro-Fuzzy Framework for Classification of EEG Signals in Motor Imagery BCI," in IEEE Transactions on Fuzzy Systems, vol. 26, no. 3, pp. 1485-1497, June 2018.doi: 10.1109/TFUZZ.2017.2728521
18. Y. Lin and H. L. T. Nguyen, "Adaptive Neuro-Fuzzy Predictor-Based Control for Cooperative Adaptive Cruise Control System," in IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 3, pp. 1054-1063, March 2020.doi: 10.1109/TITS.2019.2901498
19. R. Rohani and A. Koochaki, "A Hybrid Method Based on Optimized Neuro-Fuzzy System and Effective Features for Fault Location in VSC-HVDC Systems," in IEEE Access, vol. 8, pp. 70861-70869, 2020.doi: 10.1109/ACCESS.2020.2986919
20. M. Yeganejou, S. Dick and J. Miller, "Interpretable Deep Convolutional Fuzzy Classifier," in IEEE Transactions on Fuzzy Systems, vol. 28, no. 7, pp. 1407-1419, July 2020.doi: 10.1109/TFUZZ.2019.2946520
21. F. Santoso, M. A. Garratt, S. G. Anavatti, O. Hassanein and T. Stenhouse, "Entropy Fuzzy System Identification for the Heave Flight Dynamics of a Model-Scale Helicopter," in IEEE/ASME Transactions on Mechatronics, vol. 25, no. 5, pp. 2330-2341, Oct. 2020.doi: 10.1109/TMECH.2019.2959279
22. S. D. Nguyen, S. Choi and T. Seo, "Recurrent Mechanism and Impulse Noise Filter for Establishing ANFIS," in IEEE Transactions on Fuzzy Systems, vol. 26, no. 2, pp. 985-997, April 2018.doi: 10.1109/TFUZZ.2017.2701313
23. A. Caliskan, Z. A. Çil, H. Badem and D. Karaboga, "Regression-Based Neuro-Fuzzy Network Trained by ABC Algorithm for High-Density Impulse Noise Elimination," in IEEE Transactions on Fuzzy Systems, vol. 28, no. 6, pp. 1084-1095, June 2020.doi: 10.1109/TFUZZ.2020.2973123
24. T. Kamal, M. Karabacak, S. Z. Hassan, H. Li and L. M. Fernández-Ramírez, "A Robust Online Adaptive B-Spline MPPT Control of Three-Phase Grid-Coupled Photovoltaic Systems Under Real Partial Shading Condition," in IEEE Transactions on Energy Conversion, vol. 34, no. 1, pp. 202-210, March 2019.doi: 10.1109/TEC.2018.2878358
25. P. Sombune et al., "Automated Cerebral Emboli Detection Using Adaptive Threshold and Adaptive Neuro-Fuzzy Inference System," in IEEE Access, vol. 6, pp. 55361-55371, 2018.doi: 10.1109/ACCESS.2018.2871136
26. Z. Ben Hazem, M. J. Fotuhi and Z. Bingül, "A Comparative Study of the Joint Neuro-Fuzzy Friction Models for a Triple Link Rotary Inverted Pendulum," in IEEE Access, vol. 8, pp. 49066-49078, 2020.doi: 10.1109/ACCESS.2020.2978025
27. T. Gao, J. Wang, B. Zhang, H. Zhang, P. Ren and N. R. Pal, "A Polak-Ribière-Polyak Conjugate Gradient-Based Neuro-Fuzzy Network and its Convergence," in IEEE Access, vol. 6, pp. 41551-41565, 2018.doi: 10.1109/ACCESS.2018.2848117
28. S. El-Sappagh, F. Ali, A. Ali, A. Hendawi, F. A. Badria and D. Y. Suh, "Clinical Decision Support System for Liver Fibrosis Prediction in Hepatitis Patients: A Case Comparison of Two Soft Computing Techniques," in IEEE Access, vol. 6, pp. 52911-52929, 2018.doi: 10.1109/ACCESS.2018.2868802
29. M. M. Ferdaus, M. Pratama, S. G. Anavatti and M. A. Garratt, "PALM: An Incremental Construction of Hyperplanes for Data Stream Regression," in IEEE Transactions on Fuzzy Systems, vol. 27, no. 11, pp. 2115-2129, Nov. 2019.doi: 10.1109/TFUZZ.2019.2893565
30. F. Haque, M. B. I. Reaz, M. E. H. Chowdhury, F. H. Hashim, N. Arsad and S. H. M. Ali, "Diabetic Sensorimotor Polyneuropathy Severity Classification Using Adaptive Neuro Fuzzy Inference System," in IEEE Access, vol. 9, pp. 618-7631, 2021.
31. C. Chen and C. Liu, "Reinforcement Learning-Based Differential Evolution With Cooperative Coevolution for a Compensatory Neuro-Fuzzy Controller," in IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 10, pp. 4719-4729, Oct. 2018.doi: 10.1109/TNNLS.2017.277287
32. P. N. Montes Dorantes, P. H. Ireta Sanchez, J. M. Velarde Cantu, E. Linan Garcia and G. M. Mendez, "Design and Optimization of Distribution Routes Using Evolutionary Strategy and Type-1 Singleton Neuro-Fuzzy Systems," in IEEE Latin America Transactions, vol. 16, no. 5, pp. 1499-1507, May 2018.doi: 10.1109/TLA.2018.8408447
33. S. Forouhari and A. Abu-Siada, "Application of adaptive neuro fuzzy inference system to support power transformer life estimation and asset management decision," in IEEE Transactions on Dielectrics and Electrical Insulation, vol. 25, no. 3, pp. 845-852, June 2018.doi: 10.1109/TDEI.2018.006392
34. H. Xue, Z. Zhang, M. Wu and P. Chen, "Fuzzy Controller for Autonomous Vehicle Based on Rough Sets," in IEEE Access, vol. 7, pp. 147350-147361, 2019.doi: 10.1109/ACCESS.2019.2946663
35. Bhaskaran, S.; Marappan, R.; Santhi, B. Design and Analysis of a Cluster-Based Intelligent Hybrid Recommendation System for E-Learning Applications. Mathematics **2021**, 9, 197.

https://doi.org/10.3390/math9020197
36. Suman, S.; Kumar, D.; Kumar, A. Fuzzy Based Convolutional Noise Clustering Classifier to Handle the Noise and Heterogeneity in Image Classification. Mathematics **2022**, 10, 4056. https://doi.org/10.3390/math10214056
37. Pramanik, A.; Das, A.K.; Pelusi, D.; Nayak, J. An Effective Fuzzy Clustering of Crime Reports Embedded by a Universal Sentence Encoder Model. Mathematics **2023**, 11, 611. https://doi.org/10.3390/math11030611