An Improved K-Nearest Neighbor Algorithm for Pattern Classification

Zinnia Sultana[1], Ashifatul Ferdousi[2], Farzana Tasnim[3] and Lutfun Nahar[4]
Dept. of Computer Science & Engineering
International Islamic University Chittagong
Chittagong, Bangladesh[1,2,3,4]

***Abstract*—This paper proposed a "Locally Adaptive K-Nearest Neighbor (LAKNN) algorithm" for pattern exploration problem to enhance the obscenity of dimensionality. To compute neighborhood local linear discriminant analysis is an effective metric which determines the local decision boundaries from centroid information. KNN is a novel approach which uses in many classifications problem of data mining and machine learning. KNN uses class conditional probabilities for unfamiliar pattern. For limited training data in high dimensional feature space this hypothesis is unacceptable due to disfigurement of high dimensionality. To normalize the feature value of dissimilar metrics, Standard Euclidean Distance is used in KNN which s misguide to find a proper subset of nearest points of the pattern to be predicted. To overcome the effect of high dimensionality LANN uses a new variant of Standard Euclidian Distance Metric. A flexible metric is estimated for computing neighborhoods based on Chi-squared distance analysis. Chi-squared metric is used to ascertains most significant features in finding k-closet points of the training patterns. This paper also shows that LANN outperformed other four different models of KNN and other machine-learning algorithm in both training and accuracy.**

***Keywords*—LANN algorithm; Standard Euclidian Distance; variance based Euclidian Distance; feature extraction; pattern classification**

I. Introduction

Nearest neighbor classifier is a simplest, oldest and wide-ranging method for classification. It classifies an unidentified pattern by choosing the adjacent example in the training set and measured by a distance metric. It is one of the most common instance-based learning method. Simplicity, transparency and fast training time are the advantage of this algorithm. Instances of nearest neighbor denoted as a point of Euclidian space. It is a conceptual method that can be used to approximate real-valued or discrete-valued target function. K nearest neighbor algorithm is best suited for small data sets and which datasets have less features. This algorithm considers close relationship for similar things. In other words, the similar things of neighbors are considered one of them. For example, if mangoes' appearances is more similar to apple, orange, and guava (fruits) than horse, dog and cat (animals), then most likely mango is a fruit.

In pattern recognition problem, a feature vector $x = (x_1, ——X_q) \epsilon \Re_q$, is considered as an object like J classes, and the goal is to form a classifier that allots x to the exact class from a given set of N training samples. The simplest and alluring approach to solve this problem is the K Nearest Neighbor (KNN) [1][2] classification. Rather than fixed data points this method works on continuous and overlapping neighborhoods [3]. This method uses different neighborhood for each single query so that all points in the neighborhood are adjacent to the query to the extent possible [4][5][6]. KNN uses Straight Euclidean distance to discover the k-closest points from query point [7][8][9][10]. This can influence a real less important feature more than that of others to classify a pattern and misclassify the pattern due to dissimilar metric in measuring the feature values [11][12]. It can seriously affect in the training set with high dimensional feature space [13]. Several biases are introduced in KNN for high dimensional input feature space with limited samples [14].

A modified metric of Standard Euclidean Distance is proposed here, which uses the variance of each feature to give identical influence on the decision to all dissimilar metrics in the feature values [15]. Distance is weighted as chi-squared metric that discovers most relevant features in finding k-closet points to the pattern under consideration from the training space [16].

A locally adaptive form of nearest neighbor classification (LANN) is proposed here to upgrade the obscenity of dimensionality [17]. An effective metric is used here to compute neighborhoods which determines the local decision boundaries from centroid information, and then shrink neighborhoods in directions orthogonal to these local decision boundaries, and extend them parallel to the boundaries [18][19] [20].

To give all features equal influences on the pattern classification a variance based Euclidean distance metric is used in the proposed algorithm instead of straight Euclidean distance metric. The variance of each feature is calculated during training.

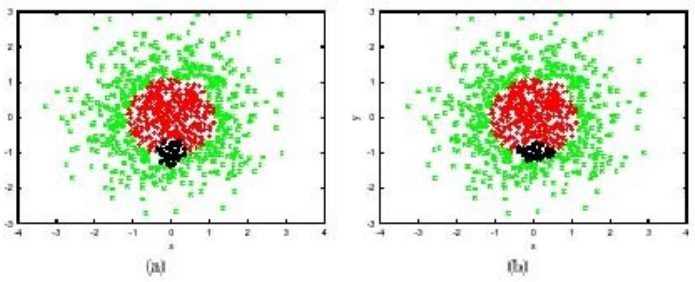

Fig. 1. Neighborhood of the Query Point.

Fig. 1 shows an example. There are two classes and both classes data are produced from a bivariate standard normal

distribution. The radius of class one data is less than or equal to 1.15, while radius of class two is greater than 1.15. As a result, class one is surrounded by class two. Fig. 1(a) shows the nearest neighborhood of size 50 of a query located at (0, -1) near the class boundary. This neighborhood is computed using the Euclidian distance metric Fig. 1(b) displays the neighborhood of same size computed by using the adaptive nearest neighbor classification algorithm. The amended neighborhood is elongated along the direction of the true decision boundary and constricted along the direction orthogonal to it, which is the most relevant direction for the given query.

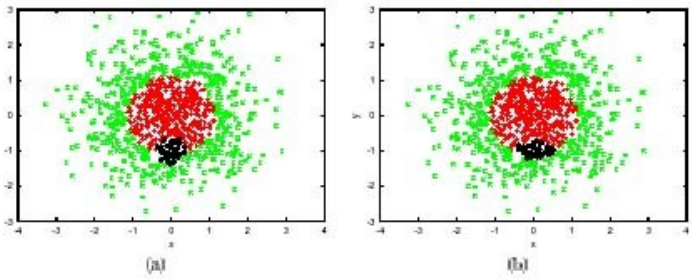

Fig. 2. Spherical Neighborhood of the Query Point.

Fig. 2 Plot (a) shows the spherical neighborhood of the query point (0, -1) containing 50 points (shown as darker circles). Plot (b) shows the corresponding neighborhood found by the proposed algorithm also containing 50 points. After applying the adaptive procedure, the neighborhood is constricted along the most relevant dimension and elongated along the less important one.

This paper proposed an algorithm that can be used in many practical applications of pattern recognition problem in machine learning technology for pattern classification tasks. It has been compared experimentally with KNN, DANN and C4.5 in a large number of artificial and natural learning domains. Experimental result shows that use of Variance based Euclidean distance metric and FRW perfectly removes the problem of constant class conditional probabilities in KNN and improves the performance of KNN.

II. Literature Review

Locally adaptive KNN algorithms indicate the value of k that should be used to categorize an interrogation by accessing the outcomes of cross-validation calculations in the resident locality of the query [21] [22]. Local KNN procedures are exposed to complete analogous to KNN in experimentations with twelve frequently secondhand data sets.

Deepti et al. [23] proposed a Quad Division prototype for stirring uneven class distribution by using Selection based K-nearest Neighbor classifier. Here the performance of QDPS based on KNN technique is assessed in fraud detection in mobile advertising. The utility of the QDPSKNN is likened with base model KNN and other selection methods, namely NearMiss-1, NearMiss-2, NearMiss-3, and Condensed Nearest Neighbor (CNN).

Suyanto et al. [24]introduced a new variant of KNN called Multi-Voter Multi-Commission Nearest Neighbor to observe the profit by enhancing the Local Mean based Pseudo Nearest Neighbor. MVMCNN is gained extra nearby than LMPNN. And then compared it with two single voter models: KNN and BMFKNN, however it shows the multi voter model better decision than the other model.

Armand et al. [25] proposed a metaheuristic search algorithm named Simulated Annealing, to choose an optimal k, thus rejecting the prospect of an exhaustive search for optimal k. Hence, the result is compared with in four different classification method to determine a substantial development in the computational competence compared to the KNN methods.

D. Maruthi et al. [26] introduced an effective classification system for MRI brain tumor and for giving grade of brain tumor images. The images are classified by using the adaptive k nearest neighbor classifier. However, the classification and segmentation arrival method are valued by accuracy, sensitivity and specificity.

Jieying et al. [27] proposed a precise image interpolation with adaptive KNN for searching image on the input image patch and conduct them for nonlinear mapping among low resolution and high-resolution image patches.

Jianping et al. [28] offered a local mean representation based k nearest neighbor classifier to increase the performance of classification and exceed the primary issues of KNN classification. They used two databases UCI and KEEL and also three common databases that carried out by liken LMRKNN and KNN based. However, it shows the LMRKNN significantly outperforms the KNN based methods.

Some previous works on K-Nearest Neighbor Algorithm for Pattern Classification that we have discussed above (Table I). Apart from this, no such similar topic related work exists as far as our knowledge. Our primary focus is to propose an algorithm that can be used in many practical applications of pattern recognition problem in machine learning technology for pattern classification tasks. It has been compared experimentally with KNN, DANN and C4.5 in a large number of artificial and natural learning domains but there is no work found that use the comparison among AI and NLP domain. Besides, no relation is shown in any research as per our study with use of Variance based Euclidean distance metric and FRW which perfectly removes the problem of constant class conditional probabilities in KNN and improves the performance of KNN.

III. Methodology

LANN has three main components: Variance-based Euclidean distance Metric, Feature Relevance Weight (FRW), the best K value using the majority voting scheme [12] [13]. LANN uses a variance based Euclidean Distance metric to find the adjacent neighbors of a query point from the training space and then the class is assigned with the majority class of the neighbors. The component of each feature in the distance is normalized using the variance. While finding the nearest points, distance component of each feature is weighted with chi-squared distance metric to work out the most relevant features.

The main steps of the algorithm and the working procedure are as follows (Fig. 3):

TABLE I. RELATED WORKS ON LAKNN

Ref. No	Description	Model	Limitation
[23]	Proposed a Quad Division Prototype Selection based K-nearest Neighbor classifier for establishing stirring uneven class distribution.	QDPSKNN, PS method	This method is not works well over real time large sized datasets.
[24]	Introduced a new variant of KNN called MVMCNN which is planned to observe the profits by enhancing LMPNN.	KNN, MVMCNN, LMPNN, BMFKNN	It does not give complete inquiries for the definite datasets.
[25]	To choose an optimal K value proposed a metaheuristic search algorithm and also eliminate the prospect of an exhaustive search	KNN, Adaptive algorithms, Parameter Optimization	The adaptive KNN method can't achieve good performance.
[26]	Introduced an effective classification system to classify MRI brain tumor	AKNN, Median filter	Can't provide an explanation in optimization computation complexity problem.
[27]	Proposed an accurate image interpolation with adaptive KNN searching and nonlinear regression.	AKNN	Do not explore deep learning models.
[28]	Proposed a k nearest neighbor classifier based on local mean representation.	KNN, LMRKNN	Can't explore deep learning models.
[29]	Introduced a method named density based adaptive k nearest neighbor.	Nearest Neighbor Classification, Density based method DBANN	can't create extra artificial examples to recompense for smaller class
[30]	An adaptive procedure monitoring system was planned base2d on the KNN rule.	KNN	This method does not suitable for simple processes.

Step-1: Start several Leave-One-Out Tests (Test index "T") for a single neighbor (T=1) to a threshold value (T=10). For each Leave-One-Out test, each example in the training space is classified according to the step 2 to 7.

Step-2: For each test point x0 in training space in each leave-one-out test (Query point index "j" of each "T" value, Given input parameters: K0, K1, K2, L), Initialize a feature relevance weight "wi" to 1 for each feature component in Euclidian distance measure in equation 1.

$$D(x,y) = \sqrt{\sum_{i=i}^{q} w_i \frac{x_i - y_i^2}{variance(i^{th} feature}} \tag{1}$$

$$Variance(ithfeature) = \frac{\sum_{i=1}^{N}(x_i - \overline{x})^2}{N} \tag{2}$$

$\overline{x}$is the mean value of ith feature, where q is the number of features of each point. x, y are the two data points and distance between x and y data point is D (x, y). xi and yi are the ith and ith feature value of x and y data point respectively. Equation 1 measures Euclidean distance with the normalized weight of each feature according to the variance of that feature that are in training data set. wi is the feature relevance weight for each feature.

Step-3: Compute K0 nearest neighbors of x0 by means of the variance based weighted Euclidian distance metric using equation 1 for $w_i = 1$.

Step 4: For each feature i, i =1…..q, compute feature relevance measure through equation 3 to equation 7.

$$\overline{r_i}(x_0) = \frac{1}{k} \sum_{z \varepsilon N(x_0)} r_i(z) \tag{3}$$

where $N(x_0)$ represents the neighborhood of x_0 holding the K_0-nearest training point.$r_i(x_0)$ denotes the capability of feature i to predict $Pr(j|z)s$ at $x_i = z_i$ and defined as follows:

$$r_i(z) = \sum_{j=1}^{J} \frac{[Pr(j|z) - \overline{Pr}(j|x_i = z_i)]^2}{\overline{Pr}(j|x_i = z_i)} \tag{4}$$

The nearer $\overline{Pr}(j|x_i = z_i)$ is to $Pr(j|z)$, the additional information features i carries for predicting the class posterior probabilities locally at z.$\overline{Pr}(j|x_i = z)$ is the conditional expectation of $p(j|x)$, given that xi assumes value z, where xi represents the ith feature of x.$Pr(j|z)$ and $\overline{Pr}(j|x_i = z_i)$ is estimated as follows:

$$Pr(j|z) = \frac{\sum_N^{n=1} 1(x_n \varepsilon N_i(z)) 1(y_n = j)}{\sum_N^{n=1} 1(x_n \varepsilon N_1(z))} \tag{5}$$

1(.) is function which acts as indicator, such that if the argument is true it returns 1 and if false then returns 0. $N_1(z)$ is the neighborhood centered at z containing K_1 nearest training points.

$$\overline{Pr}(j|x_i = z_i) = \frac{\sum_{x_n \varepsilon N_2(z)} 1(\mid x_{ni} - z_i \mid \leq \triangle_i) 1(y_n = j}{\sum_{x_n \varepsilon N_2(z)} 1(\mid x_{ni} - z_i \mid \leq \triangle_i)} \tag{6}$$

$N_2(z)$ is the neighborhood centered at z containing K_2 nearest training points, the value of $\triangle_i$ is selected from the interval containing a fixed number of L points:

$$\sum_{n=1}^{N} 1(\mid x_{ni} - z_i \mid \leq \triangle_i) 1(x_n \epsilon N_2(z)) = L \tag{7}$$

Step 5: Update Feature Relevance Weight (FRW) "w_i" according to equation 8 to equation 9. Feature Relevance Weight (FRW) is calculated by:

$$w_i(x_0) = (R_i(x_0))^t / \sum_{q}^{l=1} (R_i(x_0))^t \tag{8}$$

where $R_i(x_0)$ is defined by

$$R_i(x_0) = (max) Tj(r_i)(x_0) - (r_i)(x_0) \tag{9}$$

t = 1, 2 giving quadratic weighting scheme. In all our experiments we obtained optimal value for input parameters $K_1 = 5$, $K_2 = 10\%$ of N, K_0= 15% of N. L is set to half of the K_2.

Step 6: Iterate steps 2 to 5 again, in this situation each feature has some FRW value.

Step 7: Using Step 2 to 6 a FRW for each feature is obtained. Using FRW in variance based Euclidian distance metric; distance of all examples in training data set with query point x0 is calculated. The examples are ordered in ascending according to their distance value. Among them, a total of "T" examples are chosen from lowest distance to T th point. The class value with maximum number of examples is taken as the class value (majority voting scheme) of the query point x_0.

Step 8: All examples in the training space are classified following the steps from 2 to 7.

Step 9: Error rate is calculated for Tth Leave-One-Out test.

Step 10: All (T=10) Leave-One-Out Tests are completed and error rate is recorded for each test. Test with minimum error rate is chosen as best k-value for the training data set.

Step 11: Using the best k-value; classify any query point following the steps from 2 to 7.

The algorithm of LANN appears to be complex, but the core of LANN is the application of three main components: Variance based Euclidean distance metric, Feature relevance weight, Choice of the best k-value.

Algorithm (Dtraining, x0)

INPUT: $D_{training}$: a set of training examples.

x_0: a query point to be predicted.

OUTPUT: A predicted class value for x_0.

q=No. of Features in the training data set.

N=Total no. of Example in Training dataset.

for T=1 to threshold value (T=10) **do**
  for j=1 to N **do**
    x_j=An example from $D_{training}$
    D= $D_{training}$ - x_j
    Initialize FRW w_i=1 //Label-1//
    for m=1 to 2 **do**
      P=compute weighted distance of x_j by the equation 1 from D.
      $N(x_j)$ =Sort the examples(D) in ascending on P and choose K_0 neighbors from lowest distance.
      Q=compute weighted distance of $z \varepsilon N(x_j)$ by the equation 1 from D.
      $N_1(z)$ = Sort the examples(D) in ascending on Q and choose K_1 neighbors from lowest distance.
      $N_2(z)$ = Sort the examples(D) in ascending on Q and choose K_2 neighbors from lowest distance.
      for each dimension i , i=1...q, compute Relevance Measure $\overline{r_i}(x_i)$ through equation 3 to 7. **do**
        Update FRW w_i according to the equation 7 to 9.
      end for
    end for
    Compute weighted distance of x_j using the new FRW w_i by equation 1 from D. //Label-2 //
    E=Choose "T" neighbors from D Apply majority voting on E and classify x_j.
  end for
  Calculate error rate for "T" test.
end for
K = Choose best T with lowest error rate.
Compute a FRW for x_0 following steps from "Label-1 to Lablel-2".
F = Choose "K" neighbors from $D_{training}-x_0$ Class C =Apply majority voting on "F"
Return "C"

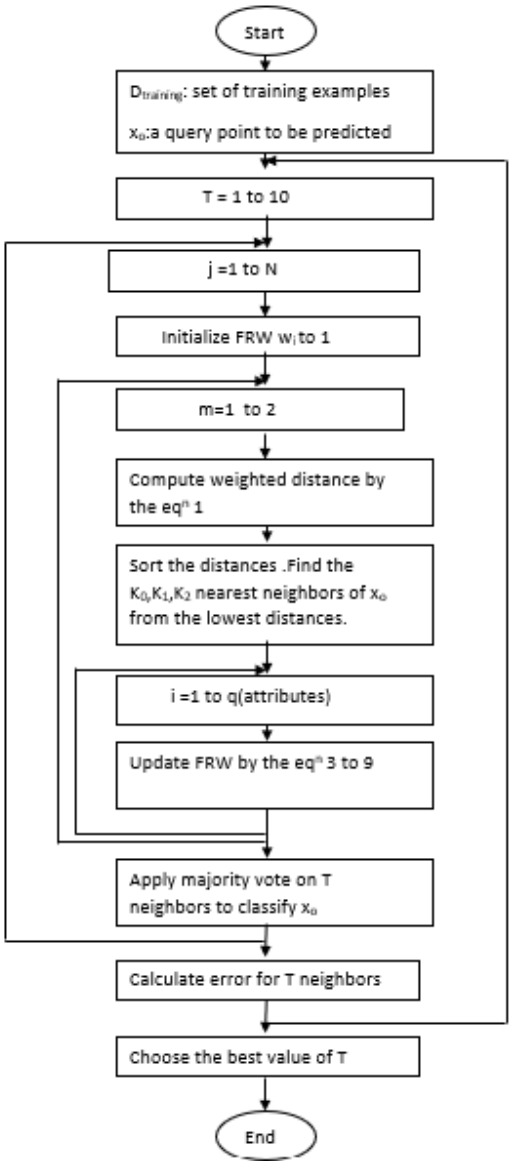

Fig. 3. Flowchart of LANN.

IV. Experimental Result

Twelve different real data sets are studied for experimental analysis of LANN. The Breast Cancer, Iris, Diabetes, Glass,

Vowel, Sonar, Hepatitis, Wine, Segmentation, Lymphography, Liver-Disorder and Lung-Cancer data are taken from UCI Machine Learning Repository [4]. All for the datasets we perform Leave-One-Out test to measure performance (Table II).

TABLE II. DOMAINS USED IN THE PROPOSED ALGORITHM (LANN)

Description of the domains used in experimental study.			
Domain name	Size	No. of classes	No. of Attributes
Breast Cancer	699	2	9
Iris	150	3	4
Diabetes	768	2	8
Glass	214	6	9
Vowel	528	11	10
Sonar	208	2	60
Hepatitis	150	2	19
Wine	178	3	13
Segment	2310	7	19
Lymphography	148	4	18
Liver-Disorder	345	3	6
Lung-Cancer	32	3	56

Table III shows the leave one out test result for 12 datasets. Table III depicts the Leave-One-Out error rates for the four methods under consideration on the twelve real world data.

The above table shows error rates (%) for different K-values. Column 1 of Table III shows that the minimum error rate is 2.43 for K=4 in breast cancer dataset. Column 2 of Table III shows minimum error rate 3.33 for K=2 for Diabetes dataset, minimum error rate for Iris dataset is 3.33 that shown in column 3 of Table III, So, the best K-value is 6. Minimum error rate for Glass dataset is 24.76 for k value 4 is shown in column 4 of Table III, 9.13 is the minimum error rate for k value 4 for sonar dataset shown in column5 of Table III, for k value 2 minimum error rate 0.56 is found for Vowel dataset that shown in column 6 of Table III, column 7 of Table III shows the minimum error rate of Hepatitis dataset which is 21.33 for k value 2. Minimum error rate of Wine dataset, Segment dataset, Lymphographic dataset, Liver disorder dataset and Lung Cancer dataset is 1.68 for k value 2, 1.63 for k value 4, 8.10 for k value 2, 22.31 for k value 4, 37.5 for k value 4 are shown in column 8, 9,10,11 and 12 respectively of Table III.

After completion of all Leave-One-Out tests we calculate the error rate of LANN by the following:

$$Errorrate(\%) = \frac{No.of\backslash failures" * 100}{TotalNo.ofInstances}$$

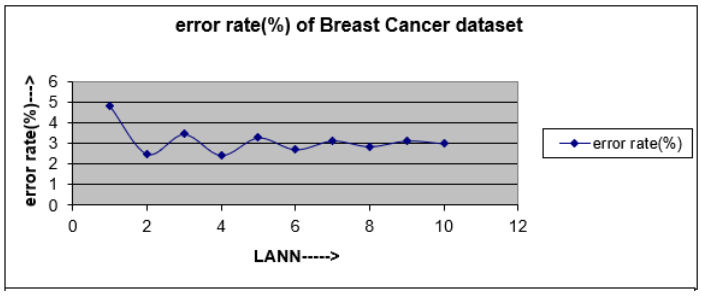

Fig. 4. Error Rate (%) Graph for Breast-Cancer Dataset.

"Fig. 4"' shows the error rate (%) graph for Breast-Cancer dataset for C4.5, DANN, KNN, LANN where the error rates

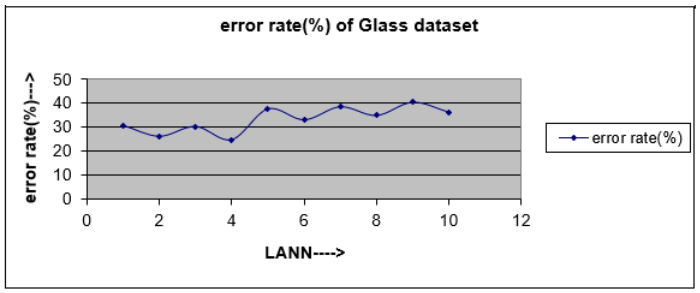

Fig. 5. Error Rate (%) Graph for Glass Dataset.

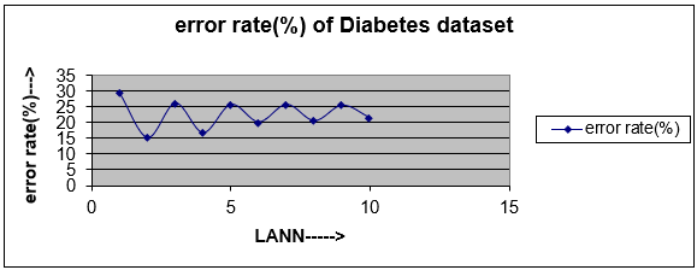

Fig. 6. Error Rate (%) Graph for Diabetes Dataset.

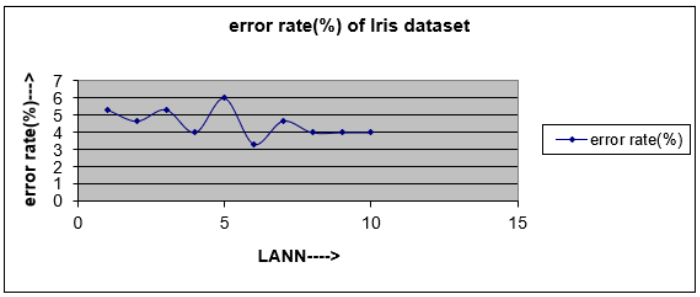

Fig. 7. Error Rate (%) Graph for Iris Dataset.

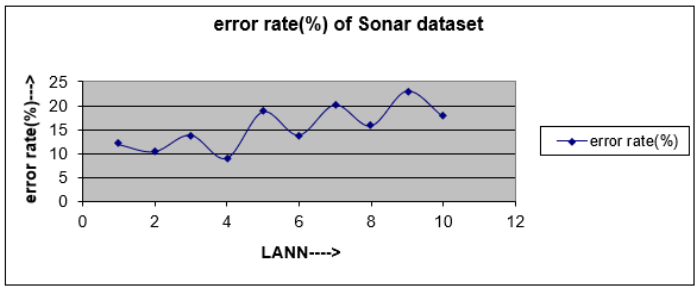

Fig. 8. Error Rate (%) Graph for Sonar Dataset.

4.70, 3.10, 4.10, 2.43 respectively. "Fig. 5" is for Diabetes dataset where the error rate is 25.00 for C4.5, 18.10 is for DANN, 24.40 is for KNN and 15.10 is for.Error rates for Iris dataset is shown in "Fig. 6". For this dataset the error rate is 8.00 for C4.5, 6.00 for DANN, 8.00 for KNN and for LANN it is 3.33; Glass dataset's error rate is shown in "Fig. 7" where the error rates for C4.5, DANN, KNN, LANN are 31.80, 27.10, 28.00, 24.76 respectively.Error rates for Sonar dataset is shown in "Fig. 8" which shows 23.10, 7.70, 12.50, 9.13 for

TABLE III. THE LEAVE-ONE-OUT TEST RESULTS FOR 12 DATASETS ARE GIVEN BELOW

	Breast cancer	Diabetis	Iris dataset	Glass dataset	Sonar dataset	Vowel dataset	Hepatitis dataset	Wine dataset	Segment dataset	Lympho graphy dataset	Liver Disor-der dataset	Lung Cancer Dataset
1	4.86	29.16	5.33	30.84	12.01	0.75	40.00	3.37	3.10	21.62	37.68	50.00
2	2.80	15.10	4.66	25.54	10.50	0.56	21.33	2.80	1.90	8.10	23.23	38.10
3	3.86	26.04	5.33	30.37	13.94	2.08	33.33	3.93	2.82	14.18	35.65	65.62
4	2.43	16.66	4.00	24.76	9.13	1.51	26.66	1.68	1.63	9.45	22.31	37.50
5	3.29	25.65	6.00	37.85	18.75	5.68	32.00	2.80	3.19	17.56	35.94	65.62
6	2.71	20.05	3.33	33.17	13.94	3.97	26.00	1.68	2.90	13.51	24.63	46.87
7	3.14	25.65	4.67	38.78	20.19	8.71	30.00	3.37	4.42	18.91	39.71	70.00
8	2.86	20.44	4.00	35.04	15.86	7.00	26.00	2.81	3.10	15.54	26.95	68.75
9	3.14	25.39	4.00	40.65	23.07	13.06	32.86	2.81	3.15	18.91	37.39	72.50
10	3.00	21.35	4.00	36.44	17.78	10.22	30.00	2.24	3.12	16.89	29.85	70.50

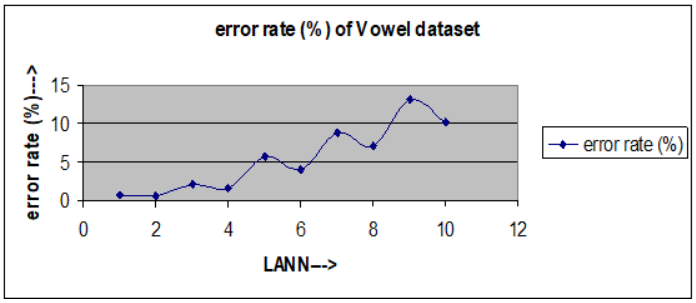

Fig. 9. Error Rate (%) Graph for Vowel Dataset.

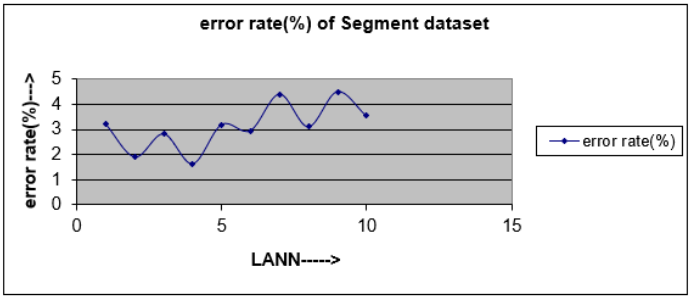

Fig. 12. Error Rate (%) Graph for Segment Dataset.

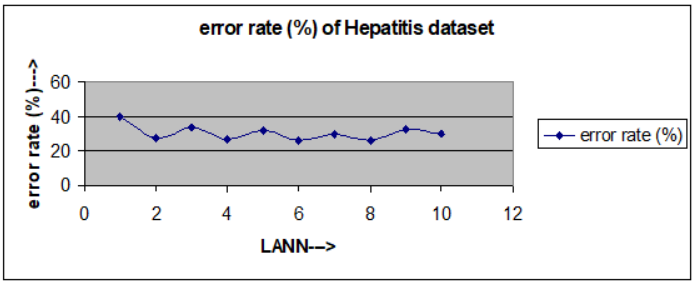

Fig. 10. Error Rate (%) Graph for Hepatitis Dataset.

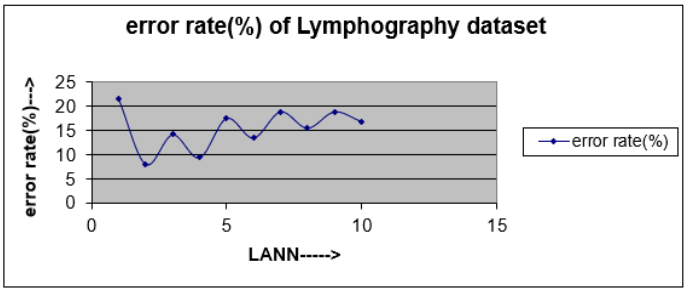

Fig. 13. Error Rate (%) Graph for Lymphography Dataset.

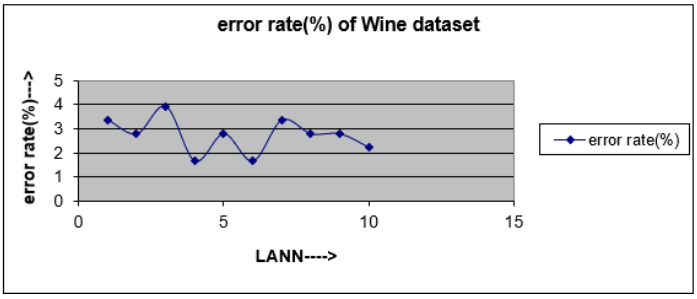

Fig. 11. Error Rate (%) Graph for Wine Dataset.

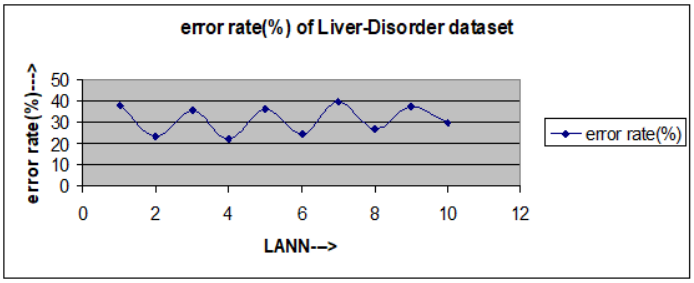

Fig. 14. Error Rate (%) Graph for Liver-Disorder Dataset.

four algorithms. “Fig. 9” shows the error rate for Vowel dataset for C4.5, DANN, KNN, LANN where error rates are 36.70, 12.50, 11.80, 0.56 respectively. Error rate (%) for Hepatitis dataset is shown in “Fig. 10” where the error rate is 18.40 for C4.5, 20.40 is for DANN, 22.30 is for KNN and 21.33 is for LANN. Wine datasets error rate is shown in “Fig. 11” the error rate (%) for C4.5 is 12.10, for DANN error rate is 13.50,14.60 is for KNN and 1.68 is for LANN. “Fig. 12” the error rate (%) where the error rates are 3.70, 2.50, 3.60, 1.63 for C4.5, DANN, KNN, LANN respectively. From “Fig. 13” error rates has been observed for Lymphography dataset where the error rate is 21.90 is is for C4.5, 17.70 for DANN, 19.30 is for KNN and 8.10 is for proposed LANN. Error rates for Liver-Disorder dataset is shown in “Fig. 14” where the error

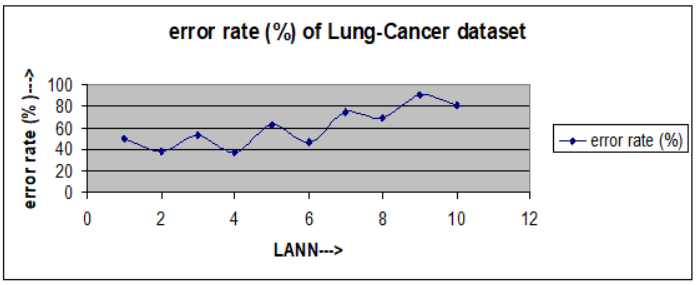

Fig. 15. Error Rate (%) Graph for Lung-Cancer Dataset.

rates are 35.10, 32.30, 34.50, 22.31 for C4.5, DANN, KNN, LANN respectively. "Fig. 15" depicts thr error rate for Lung-cancer dataset, where the error rate for for C4.5 is 57.50, for DANN it is 45.90, for KNN it is 47.90 and for LANN it is 37.50.

From the comparison Table IV it is observed that the average error rate (%) of proposed algorithm (LANN) is 12.32 whereas the average error rate (%) for C4.5, DANN, KNN are 23.17, 17.23, 19.10. Thus it can be said that, the efficiency of LANN is better than the above algorithms (Fig. 16).

TABLE IV. COMPARISON OF LANN W. R. TO OTHER ALGORITHMS

Domain no.	Domain name	C4.5	DANN	KNN	LANN
1	Breast cancer	4.70	3.10	4.10	2.43
2	Diabetes	25.00	18.10	24.40	15.10
3	Iris	8.00	6.00	6.00	3.33
4	Glass	31.80	27.10	28.00	24.76
5	Sonar	23.10	7.70	12.50	9.13
6	Vowel	36.70	12.50	11.80	0.56
7	Hepatitis	18.40	20.40	22.30	21.33
8	Wine	12.10	13.50	14.60	1.68
9	Segment	3.70	2.50	3.60	1.63
10	Lymphography	21.90	17.70	19.30	8.10
11	Liver Disorder	35.10	32.30	34.50	22.31
12	Lung-Cancer	57.50	45.90	47.90	37.50
	Average	23.17	17.23	19.10	12.32

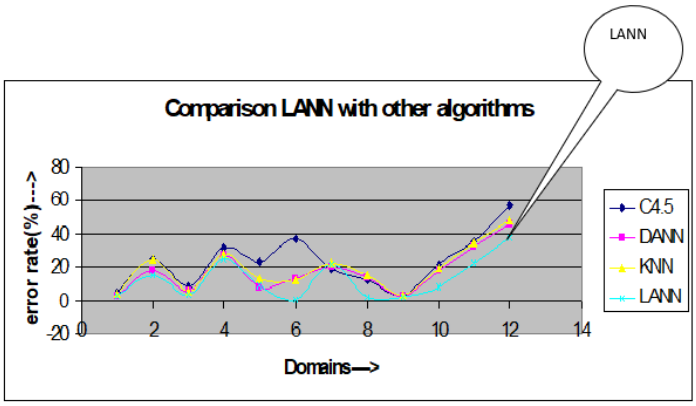

Fig. 16. Error Rate (%) of Different Domains. Horizontal Axis gives the Domain's no. and the Vertical Axis gives the Corresponding Error Rate.

V. DISCUSSIONS

There are basically two parts for pattern classification. By using an algorithm the first part creates feature vector from a given image and these features are used in the second part to learn a machine to classify an unknown pattern.

These two parts are not completely independent, this means machine learning algorithms may be benefited by knowing how the features are extracted from an image and feature extraction may be more fruitful if the type of machine leaning algorithm is known. However, the limitation of this paper is it only explored second part. That is, this work emphasis on to build a system which can classify an unknown image or pattern by using machine learning from a given set of database, all of which feature vectors have already been broken down into by an image processing algorithm. For example, the Segment dataset that is used in this work is an image classification problem. After applying the proposed algorithm (LANN) on the Segment dataset, the classification error rate is observed as 1.6%, whereas the error rates for C4.5, DANN, KNN are 3.7%, 2.5%, 3.6%, respectively. It proves that the LANN performs better than other existing algorithms in image-classification problems (Fig. 17.)

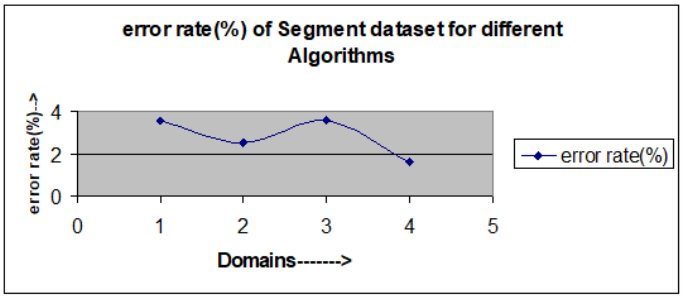

Fig. 17. Comparison of Error Rates (%) of Proposed Algorithm (LANN) with Other Algorithms for Segment Dataset.

VI. CONCLUSION

LANN presents a new variant of nearest neighbor method to classify pattern effectively. To produce neighborhood, it uses a flexible metric that are elongated along less relevant feature dimensions and constricted along most influential ones. By using this technique, the class conditional probabilities tend to be more homogeneous in the modified neighborhoods. From the experimental result it is clearly shown that LANN can potentially improve the performance of K-NN and recursive partitioning methods in some classification problems. The results are also in favor of LANN over other adaptive methods such as C4.5 and DANN.

REFERENCES

[1] A. J. Gallego, J. R. Rico-Juan, and J. J. Valero-Mas, "Efficient k-nearest neighbor search based on clustering and adaptive k values," *Pattern Recognition*, vol. 122, p. 108356, 2022.

[2] S. Uddin, I. Haque, H. Lu, M. A. Moni, and E. Gide, "Comparative performance analysis of k-nearest neighbour (knn) algorithm and its different variants for disease prediction," *Scientific Reports*, vol. 12, no. 1, pp. 1–11, 2022.

[3] R. R. Rajammal, S. Mirjalili, G. Ekambaram, and N. Palanisamy, "Binary grey wolf optimizer with mutation and adaptive k-nearest neighbour for feature selection in parkinson's disease diagnosis," *Knowledge-Based Systems*, vol. 246, p. 108701, 2022.

[4] "Journal of Ambient Intelligence and Humanized Computing, 12(2), 2867-2880. UCI Repository of Machine Learning Databases: ," http://www.ics.uci.edu/ mlearn/MLRepository.html, accessed: 2010-09-30.

[5] B.-W. Yuan, X.-G. Luo, Z.-L. Zhang, Y. Yu, H.-W. Huo, T. Johannes, and X.-D. Zou, "A novel density-based adaptive k nearest neighbor method for dealing with overlapping problem in imbalanced datasets," *Neural Computing and Applications*, vol. 33, no. 9, pp. 4457–4481, 2021.

[6] W. Zhu, W. Sun, and J. Romagnoli, "Adaptive k-nearest-neighbor method for process monitoring," *Industrial & Engineering Chemistry Research*, vol. 57, no. 7, pp. 2574–2586, 2018.

[7] S. S. Mullick, S. Datta, and S. Das, "Adaptive learning-based k-nearest neighbor classifiers with resilience to class imbalance," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 11, pp. 5713–5725, 2018.

[8] B. Tu, S. Huang, L. Fang, G. Zhang, J. Wang, and B. Zheng, "Hyperspectral image classification via weighted joint nearest neighbor and sparse representation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 11, pp. 4063–4075, 2018.

[9] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.

[10] J. H. Friedman *et al.*, "Flexible metric nearest neighbor classification," Citeseer, Tech. Rep., 1994.

[11] I. Gazalba, N. G. I. Reza *et al.*, "Comparative analysis of k-nearest neighbor and modified k-nearest neighbor algorithm for data classification," in *2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*. IEEE, 2017, pp. 294–298.

[12] C. Patgiri and A. Ganguly, "Adaptive thresholding technique based classification of red blood cell and sickle cell using naíve bayes classifier and k-nearest neighbor classifier," *Biomedical Signal Processing and Control*, vol. 68, p. 102745, 2021.

[13] L. Xiong and Y. Yao, "Study on an adaptive thermal comfort model with k-nearest-neighbors (knn) algorithm," *Building and Environment*, vol. 202, p. 108026, 2021.

[14] J. Zheng, W. Song, Y. Wu, and F. Liu, "Image interpolation with adaptive k-nearest neighbours search and random non-linear regression," *IET Image Processing*, vol. 14, no. 8, pp. 1539–1548, 2020.

[15] D. Jiang, W. Zang, R. Sun, Z. Wang, and X. Liu, "Adaptive density peaks clustering based on k-nearest neighbor and gini coefficient," *IEEE Access*, vol. 8, pp. 113 900–113 917, 2020.

[16] A. V. Kachavimath, S. V. Nazare, and S. S. Akki, "Distributed denial of service attack detection using naïve bayes and k-nearest neighbor for network forensics," in *2020 2nd International conference on innovative mechanisms for industry applications (ICIMIA)*. IEEE, 2020, pp. 711–717.

[17] Z.-X. Guo and P.-L. Shui, "Anomaly based sea-surface small target detection using k-nearest neighbor classification," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 6, pp. 4947–4964, 2020.

[18] H. Su, Y. Yu, Z. Wu, and Q. Du, "Random subspace-based k-nearest class collaborative representation for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 8, pp. 6840–6853, 2020.

[19] B. Wang and Z. Mao, "A dynamic ensemble outlier detection model based on an adaptive k-nearest neighbor rule," *Information Fusion*, vol. 63, pp. 30–40, 2020.

[20] J. Gou, W. Qiu, Z. Yi, Y. Xu, Q. Mao, and Y. Zhan, "A local mean representation-based k-nearest neighbor classifier," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 3, pp. 1–25, 2019.

[21] H. Ye, P. Wu, T. Zhu, Z. Xiao, X. Zhang, L. Zheng, R. Zheng, Y. Sun, W. Zhou, Q. Fu *et al.*, "Diagnosing coronavirus disease 2019 (covid-19): Efficient harris hawks-inspired fuzzy k-nearest neighbor prediction methods," *Ieee Access*, vol. 9, pp. 17 787–17 802, 2021.

[22] A. Lin, Q. Wu, A. A. Heidari, Y. Xu, H. Chen, W. Geng, C. Li *et al.*, "Predicting intentions of students for master programs using a chaos-induced sine cosine-based fuzzy k-nearest neighbor classifier," *Ieee Access*, vol. 7, pp. 67 235–67 248, 2019.

[23] D. Sisodia and D. S. Sisodia, "Quad division prototype selection-based k-nearest neighbor classifier for click fraud detection from highly skewed user click dataset," *Engineering Science and Technology, an International Journal*, vol. 28, p. 101011, 2022.

[24] S. Suyanto, P. E. Yunanto, T. Wahyuningrum, and S. Khomsah, "A multi-voter multi-commission nearest neighbor classifier," *Journal of King Saud University-Computer and Information Sciences*, 2022.

[25] J. Zhang, T. Wang, W. W. Ng, and W. Pedrycz, "Knnens: A k-nearest neighbor ensemble-based method for incremental learning under data stream with emerging new classes," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[26] Y. Wang, X. Cao, and Y. Li, "Unsupervised outlier detection for mixed-valued dataset based on the adaptive k-nearest neighbor global network," *IEEE Access*, vol. 10, pp. 32 093–32 103, 2022.

[27] Y. Cai, J. Z. Huang, and J. Yin, "A new method to build the adaptive k-nearest neighbors similarity graph matrix for spectral clustering," *Neurocomputing*, vol. 493, pp. 191–203, 2022.

[28] A. Onyezewe, A. F. Kana, F. B. Abdullahi, and A. O. Abdulsalami, "An enhanced adaptive k-nearest neighbor classifier using simulated annealing," *International Journal of Intelligent Systems and Applications*, vol. 13, pp. 34–44, 2021.

[29] D. M. Kumar, D. Satyanarayana, and M. Prasad, "Mri brain tumor detection using optimal possibilistic fuzzy c-means clustering algorithm and adaptive k-nearest neighbor classifier," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 2867–2880, 2021.

[30] Z. Pan, Y. Pan, Y. Wang, and W. Wang, "A new globally adaptive k-nearest neighbor classifier based on local mean optimization," *Soft Computing*, vol. 25, no. 3, pp. 2417–2431, 2021.