Journal of Fundamental and Applied Sciences

ISSN 1112-9867

Available online at http://www.jfas.info

Research Article

Special Issue

A KNN METHOD THAT USES A NON-NATURAL EVOLUTIONARY ALGORITHM FOR COMPONENT SELECTION

A. P. Pawlovsky

Department of Clinical Engineering, Toin University of Yokohama, Yokohama, Aoba-ku
Kurogane-cho 1614, 225-8503, Japan

Published online: 05 October 2017

ABSTRACT

This paper details an evolutionary algorithm that forms a new population by combining genes of three members of the current population. The first member is the best member of the population, the second one is the current member to be replaced and the third one is a member chosen randomly from the current population. We used this algorithm for component selection of a kNN (k Nearest Neighbor) method for breast cancer prognosis. Results with the UCI prognosis data set show that we can find components that help improve the accuracy of kNN by almost 3%, raising it above 79%.

Keywords: kNN; classification; evolutionary algorithm; breast cancer, prognosis.

Author Correspondence, e-mail: pawlovsky@toin.ac.jp

doi: http://dx.doi.org/10.4314/jfas.v9i4s.10

1. INTRODUCTION

Breast cancer is one of the most common cancers in the world and with a 30% figure, it has the highest incidence in women in Japan. However, its early treatment reduces its mortality rate, reaching it only around 9% in Japan. Advances in VLSI technologies make possible now the access to powerful computers and the development of many algorithms that in the past were hard to implement efficiently. Therefore, many methods exist for breast cancer prognosis and diagnosis [1-4].

In this paper, we show one way of improving the average accuracy of the kNN method. It is a machine learning method that has high accuracy, it is easy to implement and could be used to detect different stages of breast cancer [5] and physiological characteristics with high accuracy [6]. We can improve the accuracy of kNN in several ways [7-9]. One survey on its several variants could be found in [10]. In the kNN method the similarity metric used for classification is usually the Euclid distance. However, the adoption of other distances could lead to accuracy improvements [11-12].

For the evaluation of our approach we used the breast cancer data set of the UCI site [13]. This set contains data that is composed of 35 features (components), which 32 are usually used to perform classification.

Evaluating all the combinations of the components to find an optimal one is one possible option. But due to the large number of possible settings, it is not practical.

Principal component analysis (PCA) is one way of selecting them, and it can be used together with other methods [14-15]. Heuristic algorithms could also be used to search for near-optimal combinations. Genetic algorithms (GA) can also help to reduce the dimensionality of the data [16].

We have implemented a kNN method that evaluates combinations generated by an evolutionary algorithm (EA). The EA generates combinations of components from which only the best ones are selected. These selected combinations of components are evaluated again, in a more exhaustive way, by the kNN method to determine their accuracy characteristics. The UCI data is used pre-processed in two ways. We normalize and standardize it before using it for classification.

2. KNN METHOD AND EVOLUTIONARY ALGORITHM

We explain in this section characteristics of the kNN method we implemented and show details of the EA we developed for component selection.

2.1. kNN Implementation

The kNN (k Nearest Neighbor) method is one of the most popular algorithms used for classification tasks. Although it is simple to implement, it has shown to be very effective with several data sets and types of data.

Some of its shortcomings are the difficulty in determining the number of neighbors k to be used in the classification, choosing the metric to measure the similarity and deciding if all the attributes (components) of the data must be equally weighted in the classification process.

There are many modifications and variants of the kNN that try to overcome these and other weak points of the kNN method [10]. There exist approaches that try to optimize the weight of each attribute and others that aim to improve the performance of kNN by targeting also the distances [17-18].

The kNN method does not have a training phase neither builds a classifier. It needs classified data to work.

The usual way of measuring the accuracy of a kNN implementation is dividing all the available and already classified data in ten equal-size groups (i.e., each group contains 10% of all the data), then we take together nine of them as the data used for classification and use the last (remaining) group as the data for testing. After the first evaluation of the accuracy, we change the group used for testing and measure the accuracy again. This process is repeated until every group has been used for testing. This is what is called a ten-fold evaluation of the classification method. This kind of evaluation uses 90% of all the data for classification, uses the 10% for testing and repeat the accuracy evaluation just ten times (once for each group).

We have implemented the kNN method in such a way that it lets us to control the percentage of all the data that it uses for classification. We can set this percentage to any possible value. In the evaluation of our approach, we used nine settings of this percentage for evaluating each member (combination of components) generated by our evolutionary algorithm.

The maximum number of neighbors we can use in the classification stage changes with the size of the data used for classification. Our implementation of the kNN method evaluates the accuracy for each and all possible values of k. It also evaluates the accuracy using simultaneously six different similarity metrics. Our kNN method uses the Euclid, Manhattan, Chebyshev, Sorensen, Canberra and Mahalanobis distances. We can also repeat the accuracy evaluation for more than 10 times. We made its setting (number of trials) independent of the percentage of the data used for classification. In our implementation of the kNN method, random sampling forms the classification set.

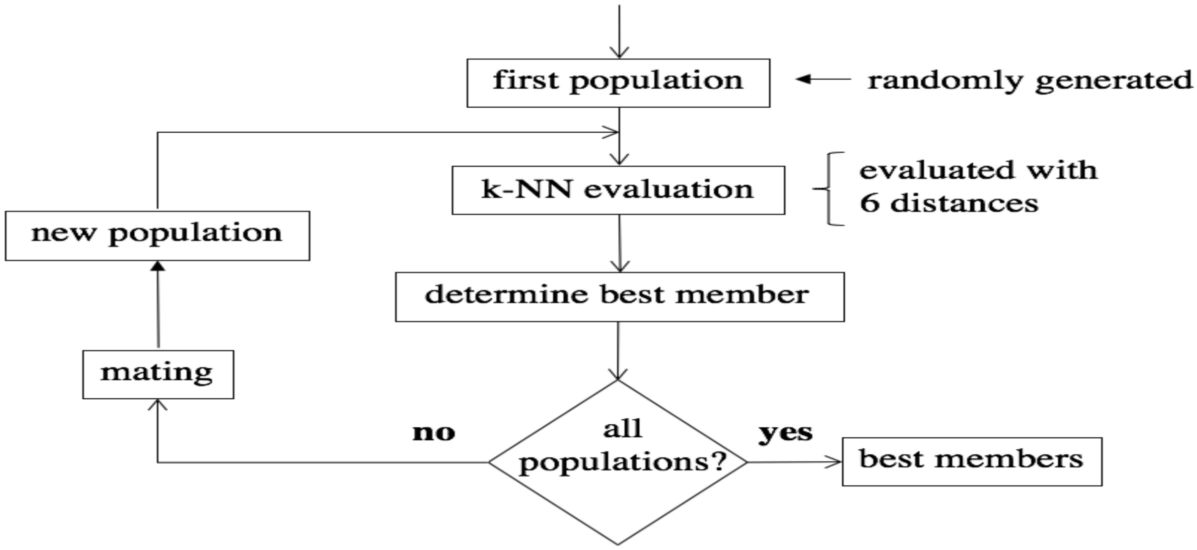

Fig.1. Selection process: uses kNN for the evaluation of EA members

2.2. Evolutionary Algorithm Details

In our approach, we try to find and optimal combination of components that will give us the best average accuracy. To do this, we use an evolutionary algorithm (EA) to find the best combination of components. The EA is part of the selection process of Fig. 1. In it we use the evolutionary algorithm to generate populations that are evaluated using kNN.

The members in one population are really a mask composed of zeros and ones (Fig. 2)

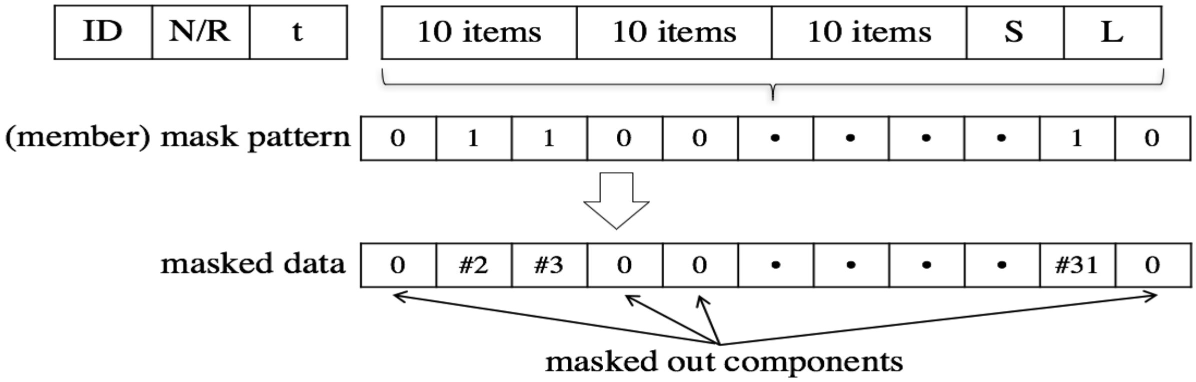

Fig.2. Masking of components using an EA member

That mask is used to select the components of the combination that the kNN method will evaluate.

We use all members in one population to generate the new members of the next one. All the populations are composed of 100 members. Each member is used to generate a masked data that in turn is evaluated using the kNN method.

The average accuracy obtained with that data becomes the evaluation (fitness) of the corresponding member. After evaluating all the members in one population we sort all them to determine the best member (Fig. 3).

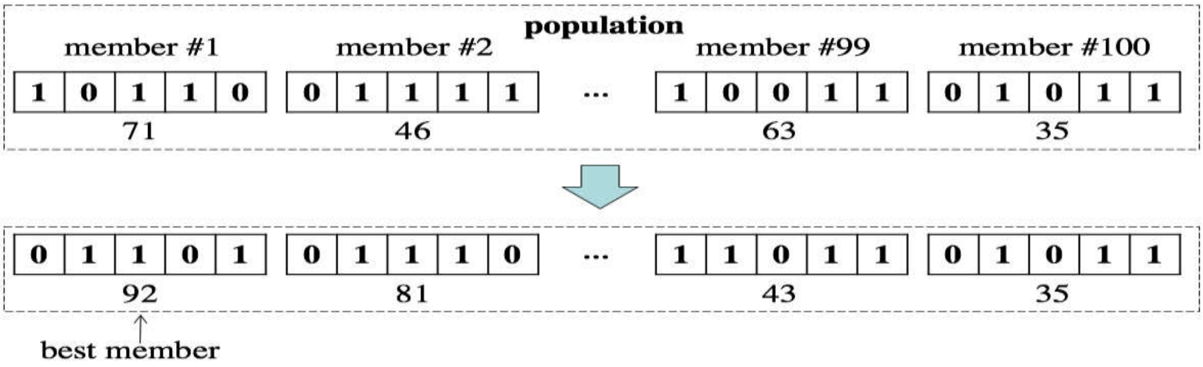

Fig.3. Sorting of EA members after its evaluation

In the selection process, we use only 10 populations. New populations are formed mating the best member with all other ones (Fig. 4).

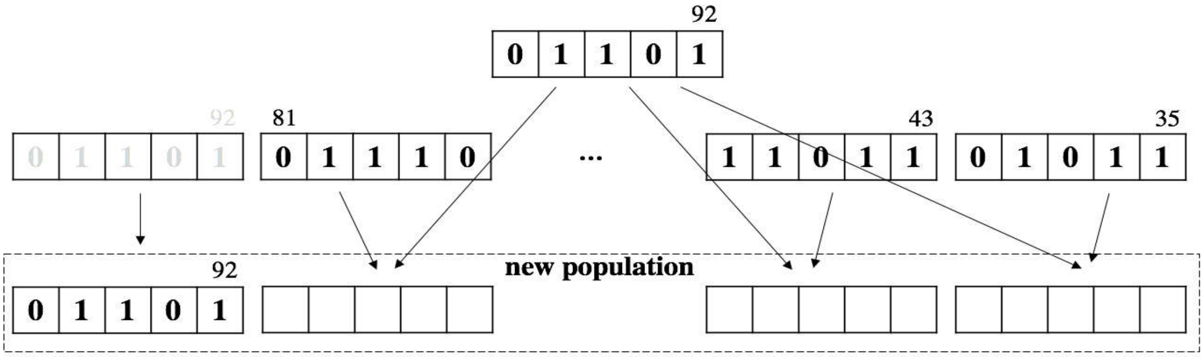

Fig.4. Mating process with the best member

The process of forming a new member is done one gene at a time controlled by a parameter, we call probability of best inheritance pbi and calculated as shown in Equation (1).

$$p_{bi} = \frac{e_b}{e_b + e_c} \tag{1}$$

Here, eb is the evaluation of the best member and ec the evaluation of its current partner.

To determine if the new member will inherit a gene from the best one we generate a random number r between 0 and 1. If r is smaller than pbi, then the gene of the best member will pass to the new one. If the member mating with the best one has a high evaluation, p_{bi} would take a value close to 0.5 and the new member will have half of its genes from the best member and the other half from members of the current population (Fig. 5).

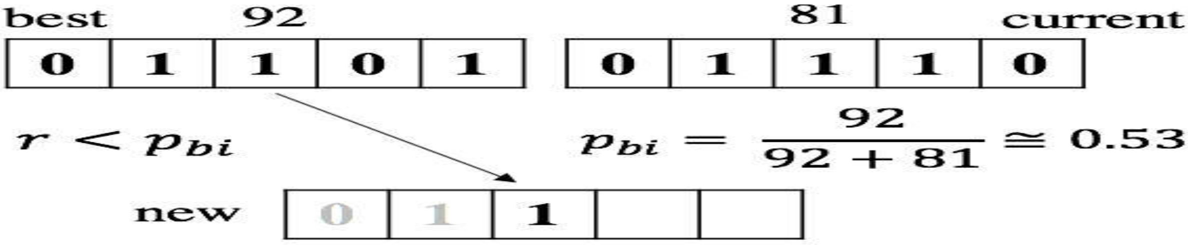

Fig.5. Controlled inheritance from the best member

When the best member mates with low evaluation members, the corresponding members of the new population will inherit most of its genes from the best one. This really is subject to the differences in evaluation between the best member and the worst ones in the population. If the difference is large, p_{bi} will be higher than 0.5 and the new member will inherit genes from the best member. As depicted in Fig. 6, for the worst member of the example population of Fig. 3, p_{bi} will be equal to 0.72 (probability of inheriting from the best member will be 72%) and the new member's genes will probably be set to those of the best one.

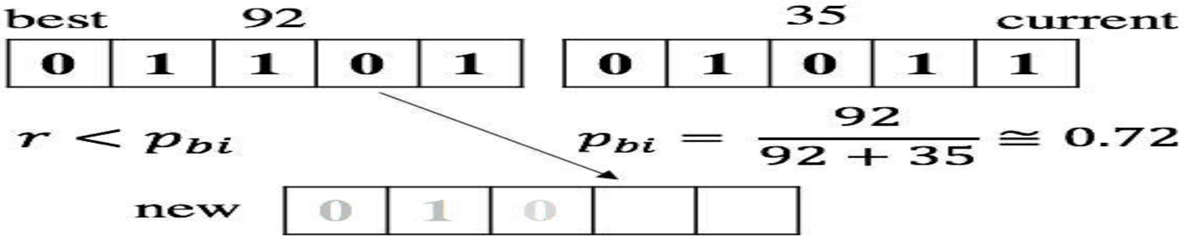

Fig.6. Low evaluation members will inherit from the best member

If r is larger than or equal to p_{bi}, first we choose randomly another member from the current population and calculate a parameter we call probability of current inheritance p_{ci} that is given by Equation (2). In it, e_o is the evaluation of the randomly chosen member.

$$p_{ci} = \frac{e_c}{e_c + e_o} \tag{2}$$

In this case, we also generate a random number r between 0 and 1 and compare it to p_{ci}. If r is smaller than p_{ci}, the new gene will be the one of the current member. Otherwise, it will be the gene of the randomly chosen member (Fig. 7). If the third member's evaluation is close to the current's one, p_{ci} will take a value close to 0.5 and nearly half of the genes determined in this way will be those genes of the member chosen randomly from the current population.

On the other hand, if the current member and the third one have a high difference in evaluation, then the new member will inherit most of its genes from the current one.

We have to recall that the random value r compared to p_{bi} is generated for each gene of the new member and not just once.

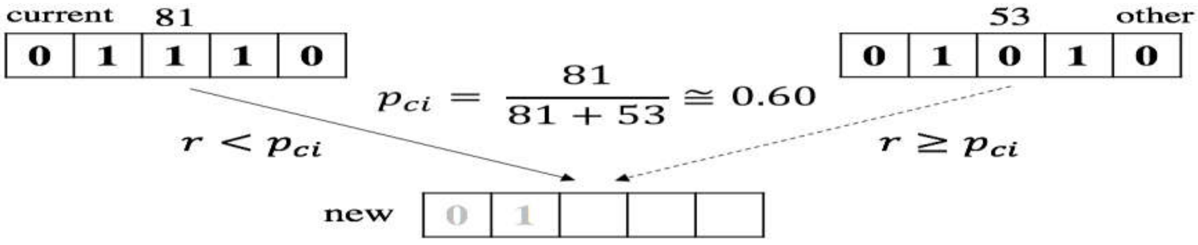

Fig.7. Inheriting a gene from the current or another member

This scheme will cause that members with close evaluations to the best member will produce new members that will inherit, in the best case, half of the genes of the best member. However, the final type of the genes will also depend on the third member (the one randomly chosen).

This mechanism of inheriting from a third member is what makes our approach unique (non-natural) and allows us bring diversity to the new population.

The selection process showed in Fig. 1 evaluates each member in a population using six different metrics (distances), and nine settings for the size of the classification set. The sizes start at 10% in increments of 10% up to 90% of all the available (already classified) data.

After the evaluation of all members in a population, the best one is not only used in the generation of a new population but it is also kept as one of the best members found in the selection process using the EA. Each one of the best members gets recorded together with the percentage for which it gave the highest evaluation. Since we used only 10 populations, 10 members for each distance, 60 members in total will form the output of the selection process. At the end of the evaluation of all the populations, we will sort all the best members for each distance and only the best two members (mask patterns) for each distance will pass to the detailed evaluation process of Fig. 8. In it, each one of

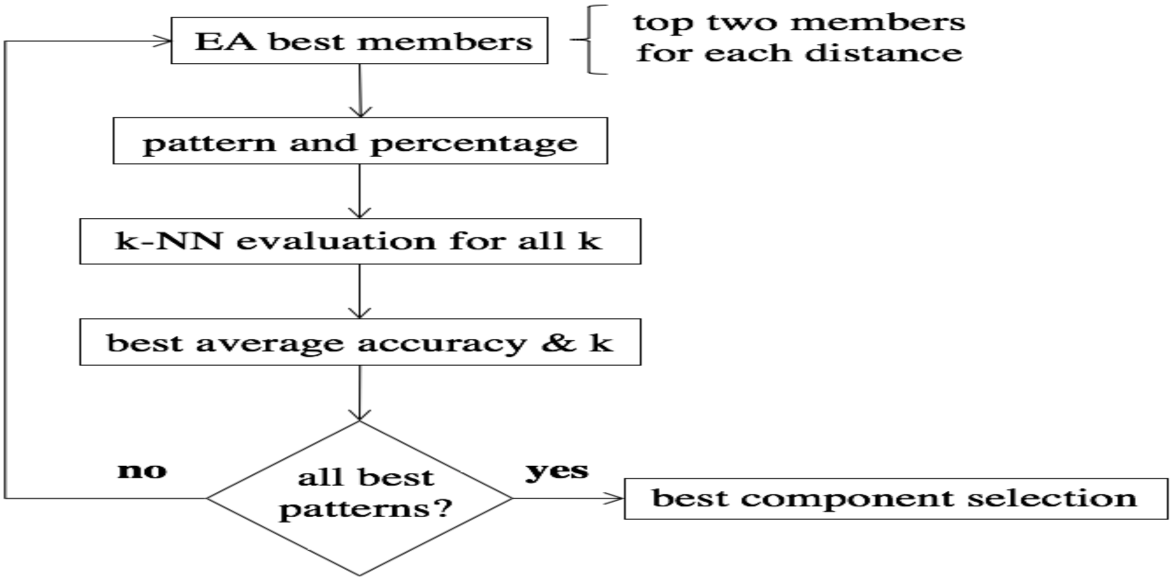

Fig.8. Detailed evaluation process of the top patterns

The 12 selected members (patterns) and its corresponding percentage are used to set kNN to evaluate them again.

Since the data for the classification set is chosen randomly from the whole data set, this set could be highly biased. In some cases, it could even contain only one class of the data.

To avoid this and other cases that could bias the results, in the detailed evaluation of the target member (pattern) we repeat the evaluation, with the same classification set size, one hundred times, for the six distances, and for all possible values of k. The only parameter that is fixed is the percentage of the data taken for the classification set. The results of the detailed evaluation are given in the next section.

3. RESULTS AND DISCUSSION

Processing the (raw) data, before using it for classification, diminishes the effect of components with different ranges of values. As indicated in the introductory section, we processed the data in two ways. One of them is normalization and the other one is standardization. The results shown here are those of the detailed evaluation process.

The accuracy results were obtained using six distances and in some of the tables, of this section, a number indicates a distance. Those are Euclid: 1, Manhattan: 2, Chebyshev: 3, Sorensen: 4, Canberra: 5 and Mahalanobis: 6.

3.1. Results Using Data Normalization

Normalization makes the data to fit it in the range [0,1]. We used the following formula to process each component value.

$$v_{norm} = \frac{v_{raw} - v_{\min}}{v_{\max} - v_{\min}} \tag{3}$$

In Equation (3), vraw is the original (raw) component value. The vmax and vmin are the maximum and the minimum values the component takes in all available data.

The 12 patterns selected by the EA using normalized data are shown in Table 1. Each one was selected using 9 classification set sizes, repeating the simulations a hundred times for all values of k and with all the six distances implemented in the kNN method.

Table 1. Patterns obtained with our EA with normalized data

Pattern	Components' Masks
1	11000101101000100010010010100010
2	11001101101100100010010010100000
3	00100110101111100100000001100101
4	00000110111011100000010101101101

5	11101100111111011101000100000011
6	11101101111111111101000100000010
7	00101100101110001101110100000000
8	00101100101110001101110100000000
9	00000001101001010101111101011010
10	00100001101001110000101100011010
11	00001000011110011010111111011110
12	00101000011110011011111110011110

We must note that the best top patterns for the Sorensen distance are identical ones (patterns 7 and 8). The best average accuracy values obtained in the detailed evaluation of these patterns are shown in Table 2.

The best average accuracy was given by pattern 2 using 12 neighbors and the Manhattan distance. The best pattern gave an average accuracy of 79.1% with a classification set size of 90%.

Table 2. Best accuracy results using normalization

Pattern	k	Mean	Distance
1	12	78.4	2
2	12	**79.1**	2
3	12	76.2	1
4	23	76.2	2
5	5	77.6	2
6	14	77.6	1
7	16	77.4	1
8	15	77.1	1
9	17	76.5	3
10	19	76.4	3
11	22	78.5	3
12	17	77.3	3

Its average accuracies, for all the possible values of k, are shown in Fig. 9. The average accuracy for this pattern remains constant, at a value of 76.7%, for a number of neighbours larger than 31.

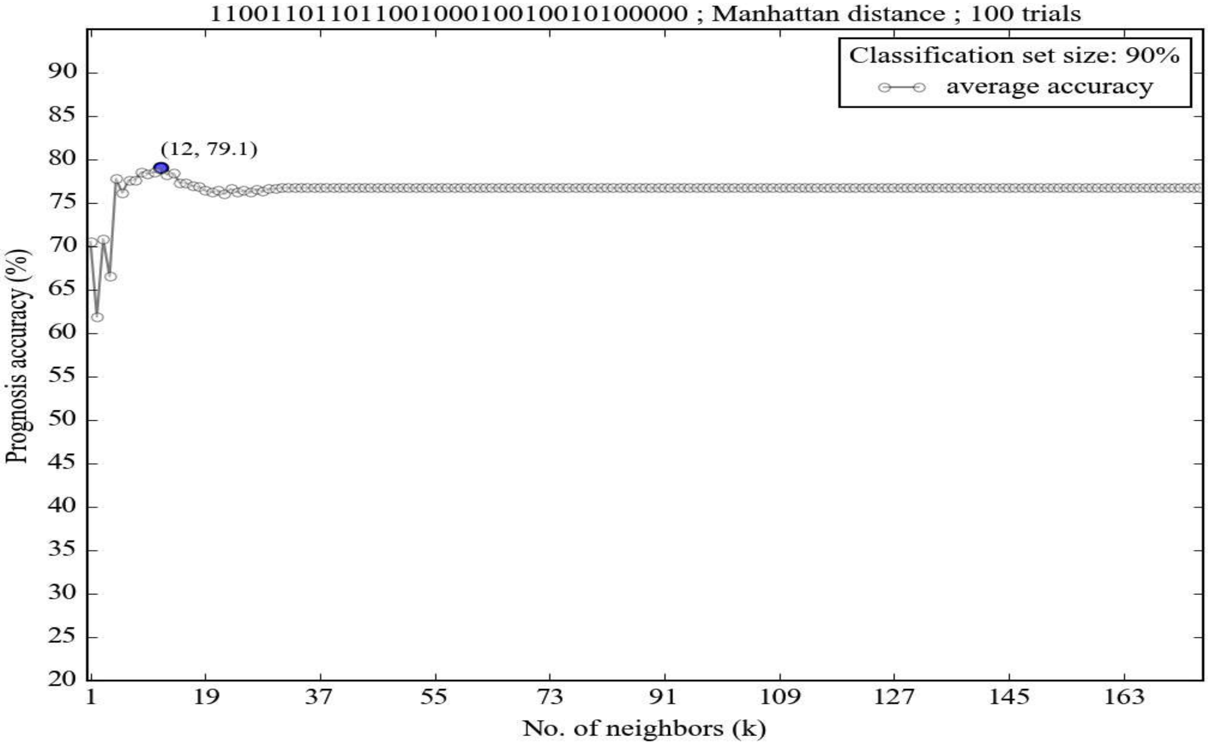

Fig.9. Average accuracy of the best pattern for normalized data

When choosing a setting for kNN, it is also important to determine the maximum and minimum accuracy values (the range of variation of the accuracy). The range of change of the accuracy for this pattern is shown in Fig. 10.

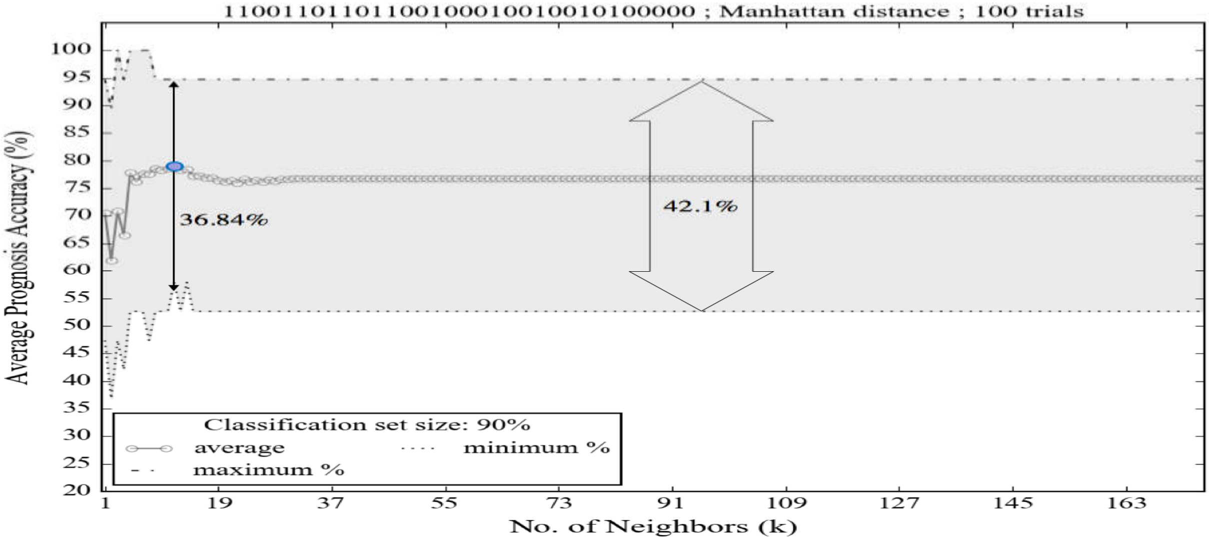

Fig.10. Minimum and Maximum values of the best pattern

This range for the best accuracy is 36.84% and the smallest one of all. The range of the accuracy fluctuates a lot for small values of k and the largest difference is of 52.6%. However, it is constant for a number of neighbours k larger than 15, showing a constant difference value of 42.1%.

We must also take into account the variation of the standard deviation (σ) of the average accuracy values. The corresponding ones are shown in Fig. 11. The standard deviation for the best average accuracy is just 8.4% and the second smallest one. The standard deviation values are constant for all values of k greater than 31 and take a value of 9.6%.

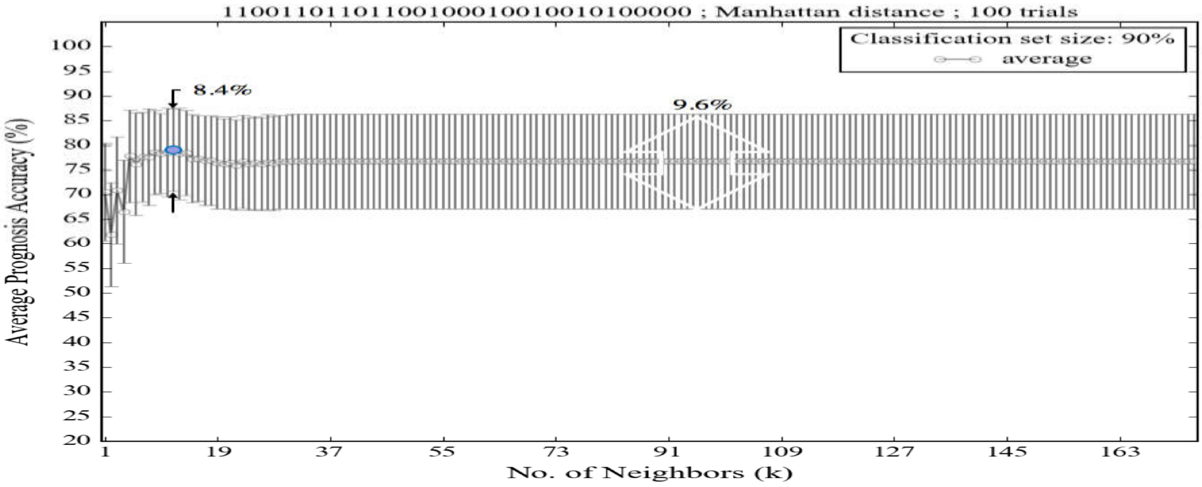

Fig.11. Standard deviation of the average accuracy of the best pattern

3.2. Results Using Data Standardization

We also processed the data standardizing it before it was used in the kNN method. In standardization, we subtract the mean value $\bar{v}$ from each value v_{raw} and divide that result by the corresponding standard deviation σ (Equation (4)). This makes the average of the resulting data equal to 0.

$$v_{std} = \frac{v_{raw} - \bar{v}}{\sigma} \tag{4}$$

The 12 patterns obtained in the selection process by the EA, using standardized data, are shown in Table 3.

Table 3. Patterns obtained with our EA with standardized data

Pattern	Component Masks
1	00001010100001011110100001100100
2	00001010100001011110100001100100
3	00010110011110101010100000110101

4	00101100001000001010110001101101
5	01000100010010001000110010000010
6	01000000010010001000111010000010
7	11011100111110100011011111110010
8	11001100111100100011111101110110
9	01000100111001011001001010111010
10	11110100110001001011101001001100
11	01000011001101010001111110010101
12	01100100001101100101001100110101

Again, we must note that the best two patterns (patterns 1 and 2) obtained with the Euclidean distance are the same. This is because the best pattern in one population is preserved and pass to a new population without changes. If it is also the best pattern of the new population, it will get recorded twice in the list of best patterns. Therefore, if it happens to also have the best evaluation of all patterns it will appear as the best two ones for a given distance. We are aware of this issue and plan to modify the selection process adding a mechanism that check for this kind of duplication.

The best average accuracies, using data standardization, for each pattern are shown in Table 4.

Table 4. Best accuracy results using standardization

Pattern	k	Mean	Distance
1	18	77.9	3
2	18	77.7	3
3	19	75.5	3
4	34	76.2	3
5	11	77.2	2
6	13	77.1	1
7	13	76.6	1
8	7 11	**79.5**	1 2

9	13	76.6	3
10	21	77.3	2
11	18	75.3	2
12	23	76.6	2

It is worth nothing that while the best two patterns for the Euclidean distance are the same, their best average accuracies differ slightly. This is due to the random method used to form the classification sets in the kNN method. Their classification sets use the same percentage of the whole data and do have the same size, but their composition will be different.

The best pattern, using standardization, was pattern 8. It gave the best average accuracy with two different distances. It gave a 79.5% average accuracy with the Euclid distance and 7 neighbors and the same highest accuracy with the Manhattan distance and 11 neighbors.

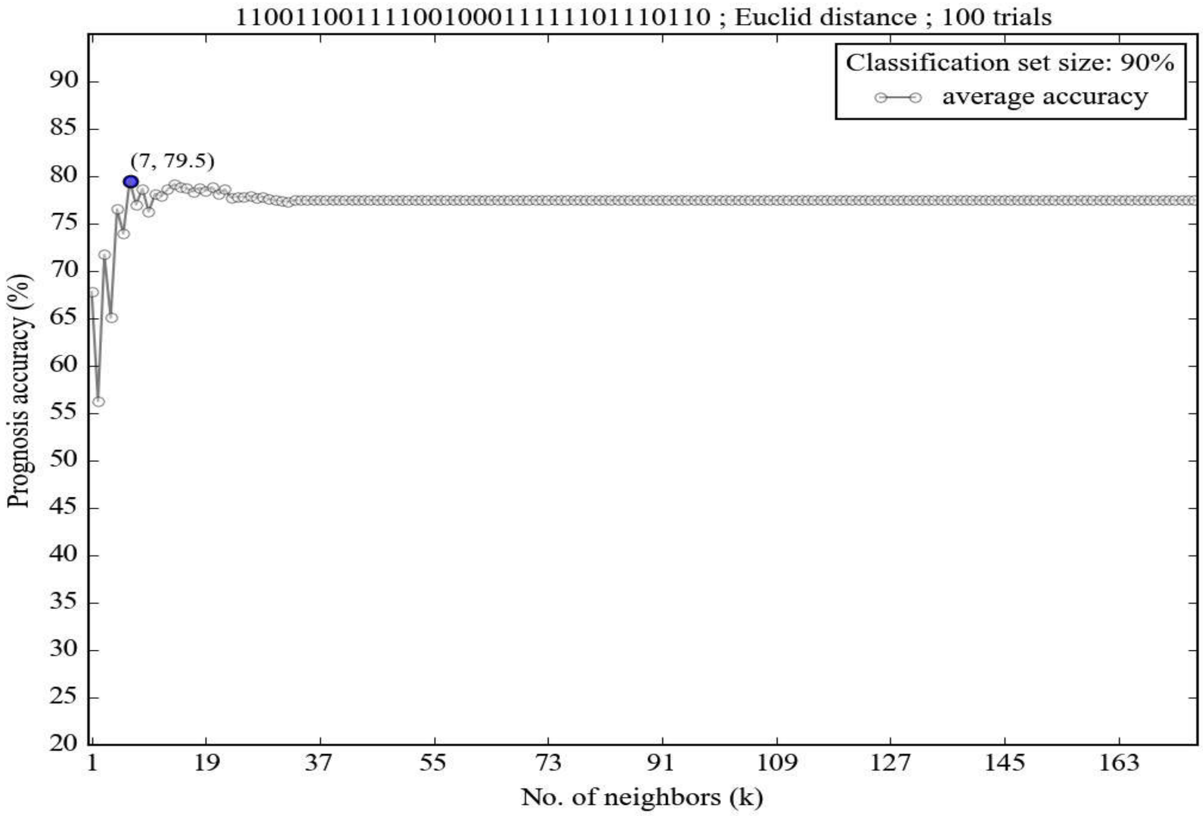

Fig.12. Results for pattern 8: Standardized data and the Euclid distance

The average accuracies of pattern 8 for the Euclid distance and all values of k are shown in Fig. 12.

The average accuracies, of this pattern are constant for a number of neighbours k greater than 31 to a value nearly to 77.5%. The average accuracy of pattern 8 is better than the one obtained with the best pattern using normalization. However, the average accuracies of all other patterns are not as high as those obtained with normalization.

The range of variation (difference between maximum and minimum values) of the average accuracy of this pattern for small numbers of neighbors is different from those obtained with the best pattern using normalization (Fig. 13).

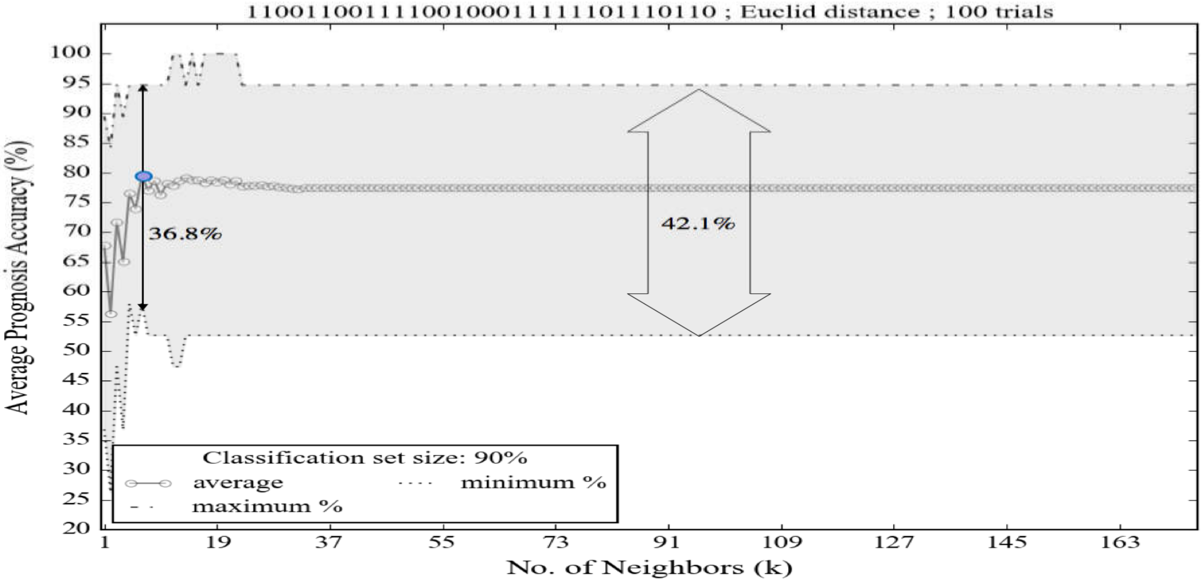

Fig.13. Maximum and minimum average accuracy values: Euclid distance

This pattern reaches average accuracy maximums of 100%, for settings of k greater than the one for the best accuracy. The best pattern of normalization also reached maximums of 100%, but for values of k smaller than the one of the best average accuracy. The variation of values for the setting giving the best average accuracy is just 36.8% and the smallest one for all values of k.

The standard deviation (σ) values of pattern 8 are shown in Fig. 14. For the best accuracy setting, the standard deviation is almost 9.3% and third to the smallest one of 9.04%.

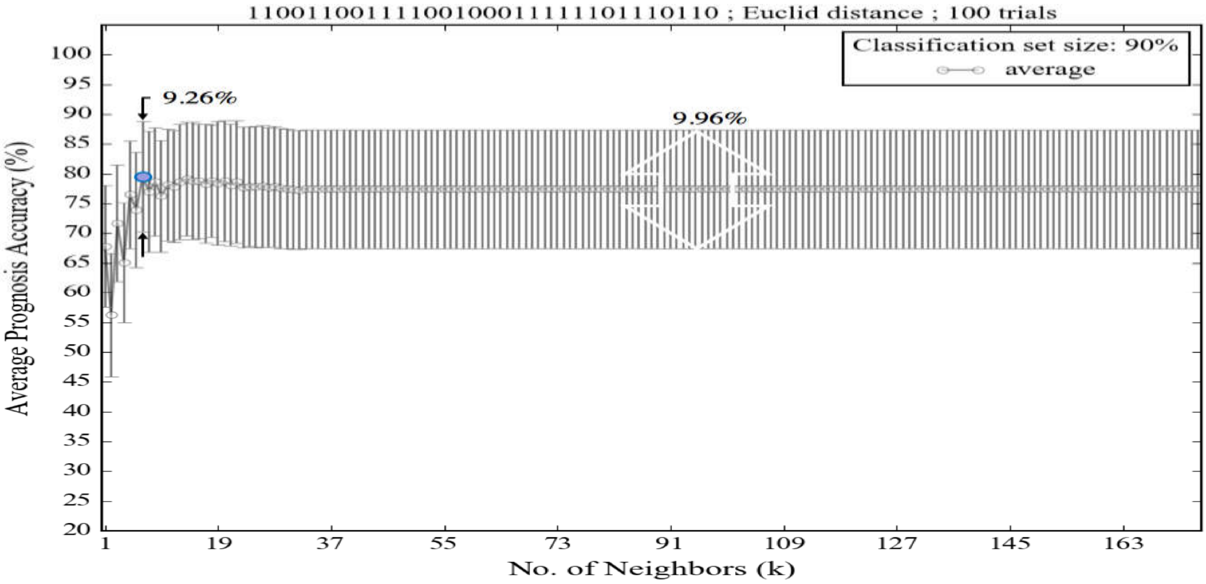

Fig.14. Standard deviation of the average accuracy: Euclid distance

The average accuracy values of the same pattern, but for the Manhattan distance are shown in Fig. 15. The average accuracy gets constant for k greater than 31 and takes the same value found for the Euclidean distance, almost 77.5%.

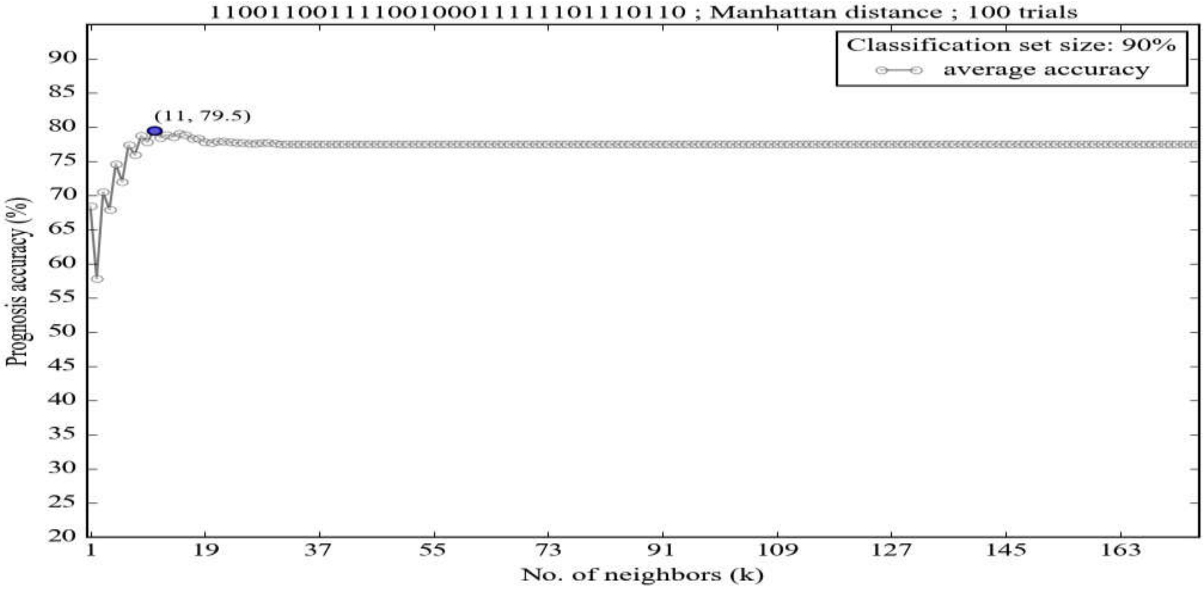

Fig.15. Pattern 8 Results: Standardized data and the Manhattan distance

The variations of the accuracy for this pattern are shown in Fig. 16. The range of variation is almost the same to that found with the Euclidean distance, constant to 42.1% for values of k larger than 19. However, for the best k the range of variation is 47.4%, second to the lowest one but larger than that of the Euclidean distance.

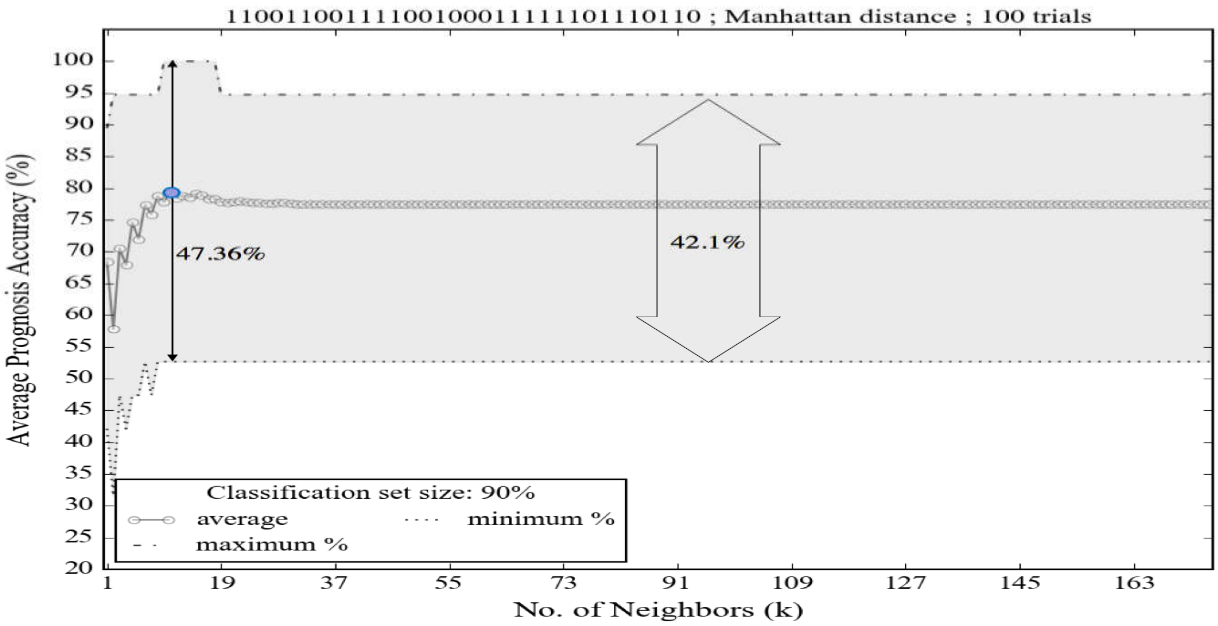

Fig.16. Maximum and Minimum values range: Manhattan distance

The standard deviation (σ) values, for this pattern and the Manhattan distance, are shown in Fig. 17. The standard deviation of the best average accuracy is 9.76%, 8th to the lowest with a difference of only 0.4%.

The comparison of the average accuracies, for the settings that give the highest average accuracy, of the two patterns discussed above is shown in Fig. 18.

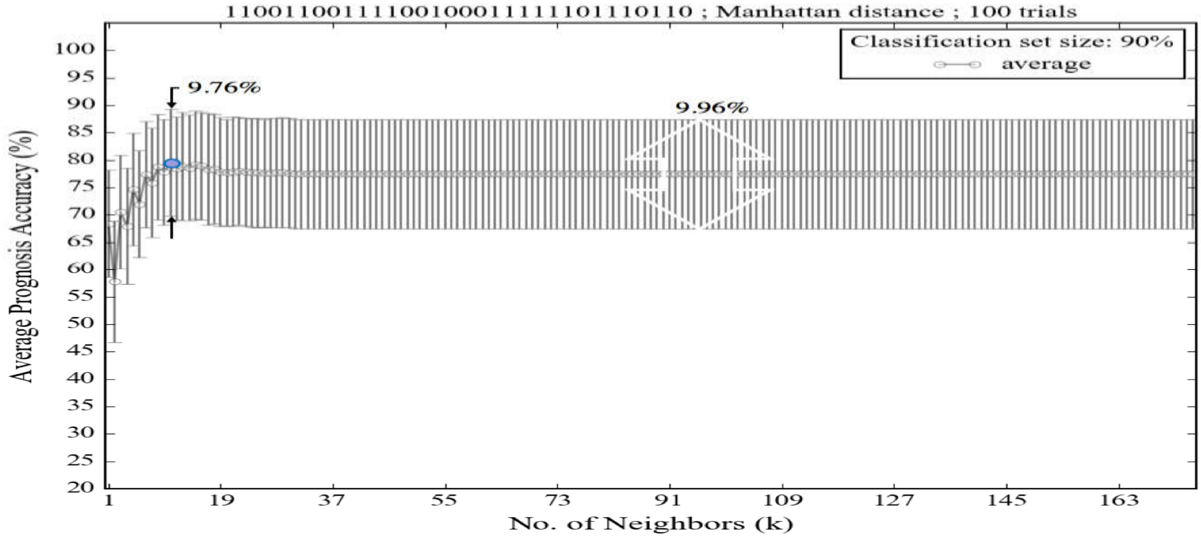

Fig.17. Standard deviation of the average accuracy: Manhattan distance

As could be seen in the box plots shown in this figure, the accuracies for the best settings of both patterns have almost the same inter-quartile range. With median and mean values almost the same. However, pattern 2 and pattern 8 with the Euclid distance show smaller dispersion of values when compared to the results of pattern 8 with the Manhattan distance. Hence, they would probably give a prognosis with better accuracy.

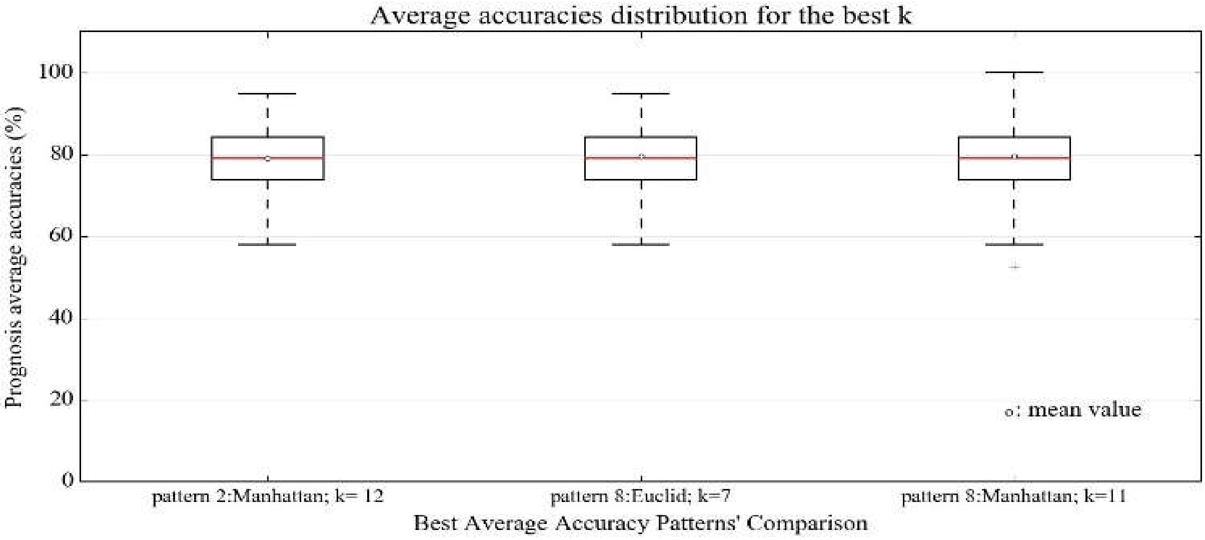

Fig.18. Comparison of the average accuracies of the best patterns

We also wanted to know if it were possible to get high average accuracies levels with the best patterns, but for other classification set sizes different from those found by the selection process. Those results are briefly explained in the next subsection.

3.3 Evaluation with Other Classification Set Sizes

We also evaluated the best patterns with classification set sizes different from those found by the selection process.

The best accuracy values for pattern 2 using data normalization are shown in Table 5 (it excludes the 90% size already showed).

Pattern 2 shows average accuracy values usually found with the kNN method for classification set sizes between 10% and 60% of all the available data. For set sizes between 20% and 50%, Sorensen, Canberra and Mahalanobis distances gave the same highest average accuracy for the same number of neighbours.

Table 5. Accuracy results of pattern 2

Classification Set Size (%)	k	Mean	Distance (s)
10	19	76.3	1~6
20	15	76.2	4,5,6
30	15	76.7	4,5,6
40	11	76.4	4,5,6
50	17	76.1	4,5,6
60	7	76.6	2

70	9	77.8	2
80	7	78.9	2

For sizes larger than 60%, the Manhattan distance gave the best results. The results found for pattern 8 using standardization are shown in Table 6.

Table 6. Accuracy results of pattern 8

Classification Set Size (%)	k	Mean	Distance (s)
10	19	76.5	1~6
20	13	76.2	4,5,6
30	15	76.2	4,5,6
40	21	76.5	4,5,6
50	11	76.8	1
60	11	76.9	1
70	9	77.2	2
80	11	77.8	2

With set sizes between 20% and 40%, Sorensen, Canberra and Mahalanobis distances gave the same highest average accuracy for the same number of neighbors. The Euclid distance gave the best results for sizes of 50% and 60%. Again, the Manhattan distance gave the best results for sizes larger than 70%.

4. CONCLUSION

We presented in this paper an implementation of an EA that uses the best member, the current member to be replaced and a third member randomly chosen from the population to set every and each gene of the corresponding member in a new population. Therefore, a new member and for the data set we used, could in same cases inherit from 32 different members of a population. Using an EA implementing this mechanism, for component selection of data to be used by a kNN method shows that we can get good combinations that would help to increase the usual 76% of average accuracy of the kNN method for breast cancer prognosis up to 79.5%.

Reviewing all the above results, we can say that any of the two ways of pre-processing the data will lead to good results with the patterns found. All of them show almost the same range of variations and values of standard deviation. We can also say that for obtaining good results, it is advisable to run the prognosis results at least one hundred times with the Euclidean and Manhattan distances.

There are still several possible improvements to implement in the EA, selection process, and detailed evaluation process. They are some of the topics left for future research.

5. REFERENCES

[1] Gupta S, Kumar D, Sharma A. Data mining classification techniques applied for breast cancer diagnosis and prognosis. Indian Journal of Computer Science and Engineering, 2011, 2(2):188-195

[2] Kharya S. Using data mining techniques for diagnosis and prognosis of cancer disease. International Journal of Computer Science and Information Technology, 2012, 2(2):55-66

[3] Salama G I, Abdelhalim M, Zeid M A. Breast cancer diagnosis on three different datasets using multi-classifiers. International Journal of Computer and Information Technology, 2012, 1(1):36-43

[4] Jacob S G, Ramani R G. Efficient classifier for classification of prognostic breast cancer data through data mining techniques. In World Congress on Engineering and Computer Science, 2012, pp. 24-26

[5] Sarkar M, Leong T Y. Application of K-nearest neighbors algorithm on breast cancer diagnosis problem. In AMIA Symposium, 2000, pp. 759-763

[6] Pagare S, Gupta S. Comparison of RBFNN, FBNN and KNN Algorithms for Face Recognition using PCA and Rectangular Feature. International Journal of Computer Applications, 2015, 115(9):42-48

[7] Pawlovsky A P, Nagahashi M. A method to select a good setting for the kNN algorithm when using it for breast cancer prognosis. In IEEE-EMBS International Conference on Biomedical and Health Informatics, 2014, pp. 189-192

[8] Odajima K, Pawlovsky A P. A detailed description of the use of the kNN method for breast cancer diagnosis. In 7th IEEE International Conference on Biomedical Engineering and Informatics, 2014, pp. 688-692

[9] Pawlovsky A P, Matsuhashi H. The use of a novel genetic algorithm in component selection for a kNN method for breast cancer prognosis. In IEEE Global Medical

Engineering Physics Exchanges/Pan American Health Care Exchanges, 2017, pp. 1-5

[10] Lamba A, Kumar D. Survey onKNN and its variants. International Journal of Advanced Research on Computer and Communication Engineering, 2016, 5(5):430-435

[11] Medjahed S A, Saadi T A, Benyettou A. Breast cancer diagnosis by using k-nearest neighbor with different distances and classification rules. International Journal of Computer Applications, 2013, 62(1):1-5

[12] Pawlovsky A P, Hiroki M A. A kNN method for breast cancer prognosis that uses a genetic algorithm for component selection. 桐蔭論叢」第 36 号 2017 年 6 月:181-186

[13] University of California (UCI). 383 data sets. Irvine: UCI, 2017

[14] Adam B, Jerzy Z, Jerzy K, Roman K. A principal component analysis of patients, disease and treatment variables: A new prognostic tool in breast cancer after mastectomy. Reports of Practical Oncology and Radiotherapy, 2000, 5(3):83-89

[15] Saxena S, Kirar V P, Burse K. A polynomial neural network model for prognostic breast cancer prediction. International Journal of Advanced Trends in Computer Science and Engineering, 2013, 2(1):103-106

[16] Deekshatulu B L, Chandra P. Classification of heart disease using k-nearest neighbor and genetic algorithm. Procedia Technology, 2013, 10:85-94

[17] Chen M, Guo J, Wang C, Wu F. PSO-based adaptively normalized weighted KNN classifier. Journal of Computational Information Systems, 2015, 11(4):1407-1415

[18] Yigit H. ABC-based distance-weighted k NN algorithm. Journal of Experimental and Theoretical Artificial Intelligence, 2015, 27(2):189-198

How to cite this article:
Pawlovsky AP. A knn method that uses a non-natural evolutionary algorithm for component selection. J. Fundam. Appl. Sci., 2017, 9*(4S), 173-192.*