International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 2 (2018) pp. 896-907

A Hybrid Approach for Prediction of Type-1 and Type-2 Diabetes using Firefly and Cuckoo Search Algorithms

R.Haritha[1], Dr. D. Suresh Babu[2] and Dr. P.Sammulal[3]

[1]Research Scholar, Computer Science & Engineering, Vaageswari College of Engineering, Telangana 506001, India

[1]ORCID: 0000-0001-8076-1042

[2]Assistant Professor, Computer Science & Engineering, Kakatiya Governament College, Telangana 506001, India.

[3]Associate Professor, Computer Science & Engineering, Jawaharlal Nehru Technological University, Hyderabad, India.

Abstract

Machine learning is the area of Artificial Intelligence that deals with develop techniques capable of learning, that is, automatically extract knowledge underlying information. Together with statistics, it constitutes the heart of the intelligent analysis of the data. The principles followed in machine learning and in the mining of data are the same: the machine generates a model from examples and use it to solve the problem. This paper proposes a firefly and cuckoo search based attribute selection algorithm with objective of higher accuracy and lower training overhead for PIMA Indian diabetic database from UCI. The experimental set up has been developed with UCI dataset using KNN classifier. The accuracy, precision and recall have been calculated as an evaluation parameter and result compared with Cuckoo search and Firefly algorithm optimized structure, the proposed structure claims higher accuracy the traditional approach.

Keywords: Cuckoo Search, Firefly Algorithm, KNN, Fuzzy-KNN, UCI.

INTRODUCTION

The World Health Organization defines diabetes as a metabolic disorder of multiple etiologic, characterized by chronic hyperglycaemia with disorders of carbohydrate, lipid and protein metabolism resulting from defects in insulin secretion, insulin, or both. [1]

A. Cause of Diabetes

The prevalence of this disease has increased five-fold in less than fifty years. This gradual increase is due to various factors[2]:

- The global aging of the population, the increase in the life expectancy of the diabetic, the increase in the fertility of diabetic women, the increase in obesity, increasing the consumption of refined sugars.
- As well as other factors that can serve as a trigger such as sedentary lifestyle, diets high in fat and protein, reduced fiber consumption, a diet deficient in complex carbohydrate and vitamin E, chronic stress, Smoking that can cause insulin resistance.

B. Diabetes classification

The criteria for the diagnosis and classification of Diabetes Mellitus (Diabetes Mellitus) were developed by an expert committee of the American Diabetes Association (ADA) [3] and a committee of the WHO.

The classification of diabetes is mainly based on its ethology and pathophysiological characteristic. Diabetes is classified into four types:

- Diabetes type 1 (DM1)
- Diabetes type 2 (DM2)
- Other specific types of diabetes
- Gestational Diabetes (DMG).

Frequently people with DM2 end up needing insulin at some point in their lives, on the other hand, some DM1 patients may progress slowly or have long periods of remission without the need for insulin. It is because of these cases that the terms insulin-dependent and non-insulin-dependent have been eliminated. [4]

The nearest neighbours method [5] is a geometric classification method that is widely used in pattern recognition because of its simplicity and robustness. The characteristics are exploited in a metric space of representation, generallyR_n provided with the Euclidean distance. Keller's method [6] is based on a fuzzy decision function that involves the distances of prototypes at the unknown point. We propose a new fuzzy decision rule in which the contribution of a prototype is not made individually, but collectively, considering its sub-neighbourhood, according to the rule: a prototype is all the more influential as it is closer from the unknown point.

Any geometric method of pattern recognition requires having a metric space of representation of characteristics. The choice is often made in practice on R_n with the Euclidean distance. The dimensions of this space can be reduced by a principal component analysis, which can also lead to a change of scale, so as to adjust the standard deviations according to each direction. We will not deal here with these issues of data analysis and pre-treatment, which form a complementary subject for study. The metric space$R_n$provided with the

Euclidean distance is isotropic: the Euclidean distance between two points is preserved for any translation and any rotation in the space of the characteristics. Moreover, it is natural in the sense of the view, allowing us to evaluate the distances in a plane. The probability density, which has the dimension of a probability per unit of volume, is a function of each point of the representation space. Deciding whether a point belongs to a class is equivalent to retaining the class with the highest probability density at that point. The problem posed in probabilistic terms is therefore to find a reliable estimator of the relative probability density of each class at any point in the representation space accessible by a test. In a "fuzzy" perspective, it is the degree of belonging to a class, which depends on the geometrical magnitudes of the representation space and parameters to be adjusted, makes it possible to decide. Also the function which expresses the degree of belonging can be regarded as a density of probability to a multiplicative constant.

To estimate the probability density, we can use global parametric methods: we give ourselves a global or semi-global analytical form of the distribution, and we estimate the parameters using statistics. Thus a probability distribution can be for example Gaussian or Rayleigh, pure or mixed. On the contrary, we can consider that the density of probability is a purely local function which depends only on a neighbourhood of the point where we try to evaluate it. In this case, only a few prototypes closest to the unknown point will be involved in the decision. Due to the incompleteness of the data, it is common to resort to a fuzzy formalization: the estimation of the local probability density is replaced by the calculation of the degree of membership of a class. The method we propose is a classification method, related to the nearest k, fuzzy type. We will situate it within the family of these methods, then we will present the experimental conditions of its validation.

PREDICTIVE CLASSIFICATIONS IN DIABETES LITERATURE REVIEW

Lot of works has been carried out, using supervise classifiers in medical diagnosis in recent years. Several researchers have claims the potential of data mining in patient data for system modelling of diabetic prediction. Supervise classification for Pima Indian diabetes disease diagnosis deals with the goal of improving accuracy. Different training and testing scenario has been proposed to define the learning rate of classifier further the impact of learning rate in terms of accuracy is evaluated. Pima Indian diabetes dataset has 752 instances out of 500 dataset is used for training and 252 was used for testing. Authors [7] claim accuracy 79.5% with neural network. [8] has achieved the accuracy of 67.6 % with 50 % of data used for training and 376 is used for testing. on Generalized Discriminate Analysis (GDA) and Least Square Support Vector Machine (LS-SVM) has been used for improving the accuracy in system by [9].A hybrid model is claimed for using KNN in Pima Indian data base claims accuracy of 76 % with learning rate of 90 % as training[10].

The proposed methods in this paper claims higher accuracy with minimal number of training.

IN this paper KNN has been used as classifier, and further KNN classifier is improved with fuzzy KNN [11] and with respect to different learning rate accuracy is calculated.to reduce the training overhead soft computing based approach is applied for optimal attribute selection .The paper organizes in five section: 1) deals with system architecture, 2) discuss the materials and methods used as soft computing for optimal attribute selection.3) Deals with classification 4) Deals with experimental results and discussion and last 5) deals with conclusion and future scope.

SYSTEM ARCHITECTURE

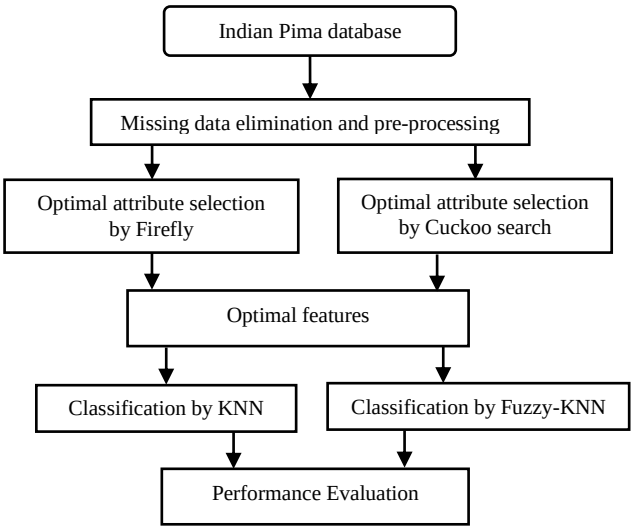

Figure 1: Flow of the automatic diabetic detection using KNN classifier

The above figure explains the flow of automatic detection of diabetes in Pima Indian database. Where feature setare optimally selected by cuckoo search and firefly algorithm.According to selected features further data is trainedand tested with KNN classifier.

A. Attribute Selection by Firefly

Fireflies are small winged beetles capable of producing a flashing cold light for a mutual attraction. Females can imitate the light signals of other species in order to attract males that they catch and devour. Fireflies have a capacitor-type mechanism, which unloads slowly until certain threshold is reached, they release the energy in the form of light. The phenomenon repeats itself cyclically. Firefly Algorithm developed by [12] is inspired by the attenuation of light over the distance and mutual attraction but he considers all fireflies as unisex.

1) Algorithm for Firefly Approach

The firefly algorithm is the popular algorithm in soft computing. It was developed by Yang [13]. A validation of the algorithm continuous fireflies on the optimization of stochastic functions is given in [14]. The pseudo-code of the Firefly Algorithm or Firefly can be as follows:

Algorithm 1: *Metaheuristics FF*

Procedure FF (Nbr_iter: number of iterations)
beginning
Generate an initial population of fireflies xi (i = 1 ..., n)
Determine the intensities of light Ii at xi via f (xi)
As long as (t <Nbr_iter) do
For i = 1 to n // all fireflies
For j = 1 to n // all fireflies
if (Ij> Ii) then
Attractiveness βi, j varies according to the distance ri, j
Move firefly I to j with attractiveness βi, j
Otherwise move I randomly
end if
Evaluate the new solution
Update Intensity Ii
Check if i firefly is the best.
End j, End i
Find the best firefly according to objective
t ++
Fact
End procedure

In the firefly algorithm, there are 4 important points:

1. Intensity of light

In the simplest case for minimization problems, brightness, or the light intensity of a firefly at a particular place x can be chosen as: $I(x) \alpha 1/f(x)$.

2. Attractiveness

In the firefly algorithm, the main form of the attractiveness function can to be any decreasing monotonous function such as the general form next:

$$\beta_{i,j} = \beta_0^* e^{-\gamma r_{i,j}^m} \quad (1)$$

Where r represents the distance between two fireflies, β_0 represents the attractiveness at r = 0 and γ is a constant coefficient of light absorption.

3. Distance

The distance between 2 fireflies i and j at x_i and x_j can be the Cartesian distance as follows:

$$r_{i,j} = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{i,k})^2} \quad (2)$$

where $x_{i,k}$ is the k^{th} component of the i^{th} firefly.

4. Movement

Moving a firefly i attracted by a brighter (attractive) firefly j, is determined by

$$x_i = (1 - \beta_{i,j}) x_i + \beta_{i,j} x_j + \alpha \left(rand - \frac{1}{2} \right) \quad (3)$$

Where the first term and the second term is due to the attraction. The third term is the randomization. α is the random parameter and can be constant. "rand" is a random number generator uniformly distributed in [0, 1].

The Firefly algorithm can be further illustrated in the following flowchart:

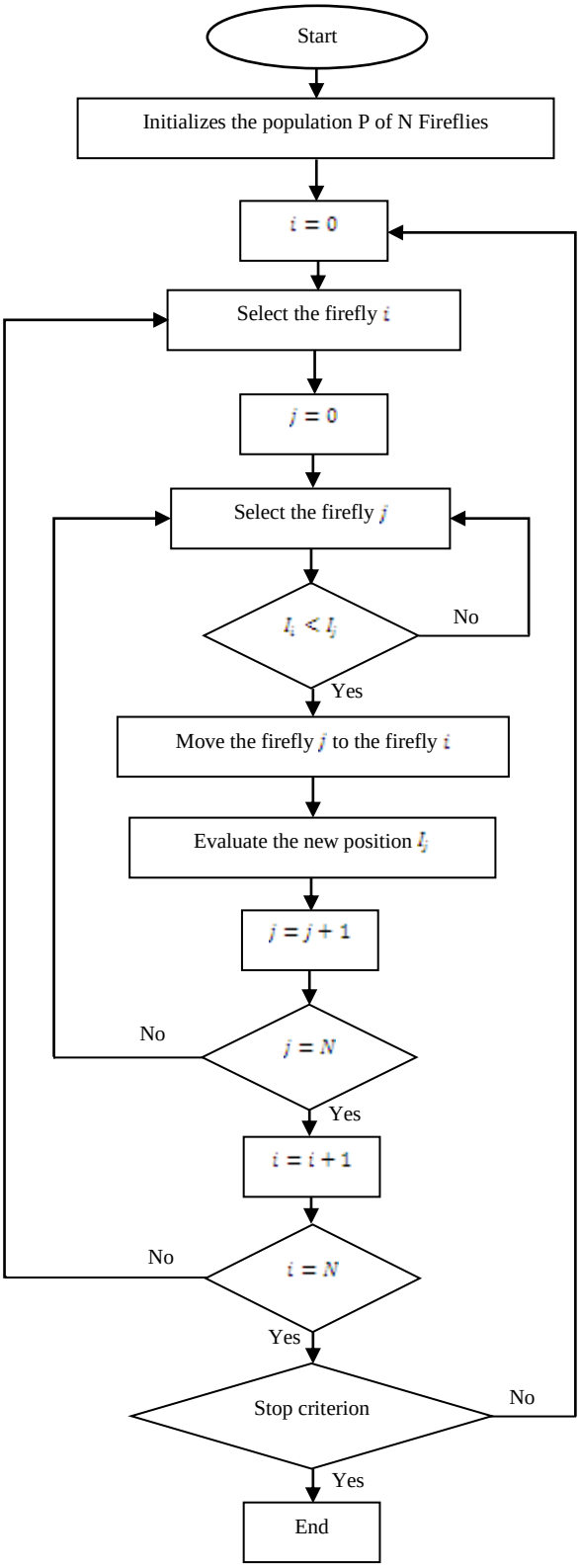

Figure 2: Firefly Algorithm Flowchart

The Firefly algorithm is formulated with two important things: The variation of the intensity of the light and the formulation of the attraction. For simplicity, the attraction of fireflies is determined according to the brightness, where the brightness is determined with the objective function.

B. Attribute Selection by Cuckoo Search

Cuckoo Search (CS) is a recent metaheuristic [15] that is inspired by the reproduction pattern of some cuckoo species.

The CS algorithm is based on the following rules:

- Each cuckoo lays only one egg at a time and places it in a nest chosen randomly.
- The best nests with eggs (solutions) of high quality are kept for the next generations.
- The number of host nests is fixed and the egg laid by a cuckoo can be discovered by the host species with a probability$p_a \in [0,1]$. In this case, the host bird either takes the egg out of the nest, or leaves the nest and builds a new one. For simplicity, this last hypothesis can be approximated by the replacement of a fraction pa of n nests by new ones.

In CS, each egg in a nest represents a solution and each cuckoo can lay a single egg (which represents a solution), the goal is to use the new and potentially better solution to replace a less good solution in a nest. Although the algorithm can be extended to the more complex case where each nest contains several eggs representing a set of solutions, we use here the simplest version where each nest contains only one egg. In this case, there is no more distinction between egg, nest or cuckoo, and each nest corresponds to an egg that also represents a cuckoo.

1) Levy flight

In CS, the pitch of cuckoo movement is determined by Levy Flight The Levy flight is a random walk in which the steps have a length having a certain distribution of probability (distribution of Levy), the direction of the steps being isotropic and random. Levy Flight is a random walk class in which jumps are distributed according to the Levy distribution which consists of a power law with infinite variance and mean of the type:

$$Levy(\beta) \sim y = x^{-\beta}, 1 < \beta \leq 3 \quad (4)$$

The generation of a direction can be made from a uniform distribution, while the generation of steps is more delicate. There are several methods to achieve this, but one of the simplest and most effective is to use Mantegna formulas to determine the pace:

$$S = \frac{u}{|v|^{1/\beta}} \quad (5)$$

Where u and v are centered Gaussian distributions such as:

$$u = N(0, \sigma_u^2), \quad v = N(0, \sigma_v^2) \quad (6)$$

With,

$$\sigma_u^2 = \frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma((1+\beta)/2)\beta 2^{(\beta-1)/2}}, \quad \sigma_v^2 = 1 \quad (7)$$

Where $\Gamma(z)$is the Gamma function.

$$\Gamma(z) = \int_0^{+\infty} t^{z-1} e^{-1} dt \quad (8)$$

2) Initialization of the Algorithm

The initial population consists of aN_H number of host nests generated randomly in the search space. Recall that the number of host nests, cuckoos and eggs are equal.

Assuming that the i^{th}host is represented by $X_i = (x_{i1}, \ldots, x_{ij}, \ldots, x_{in})$. Where n is the dimension of the problem, so each nest is generated by:

$$x_{ij} = X_{\min_j} + rnd(0,1)\left(X_{\max_j} - X_{\min_j}\right), j = 1, \ldots, n, i = 1, \ldots, N_H \quad (9)$$

The values of the corresponding cost functions are evaluated.

3) Description of the algorithm

Step 1: Global search:

The first step is to do a global search. For this, we generate new cuckoo (solutions) tests from existing cuckoo (solutions) by making Levy Flight. This is to use the following evolution law for each cuckoo:

$$\left.\begin{matrix} X_{new\ i} = X_i + \alpha \otimes Levy(\beta) \\ F_{new\ i} = f(X_{new\ i}) \end{matrix}\right\}, i = 1, \ldots, N_H \quad (10)$$

Where $\alpha > 0$ is the size of the displacement step which depends on the problem considered and such that $\alpha \approx O(L/10)$, where L is the characteristic scale of the problem considered. The $\otimes$ product is the term product term.

In order to know where / if the new i^{th} cuckoo is going to drop an egg, he chooses randomly a host nest $X_j (j \neq i)$ of cost function F_j.

The new cuckoo pond if:

$$F_{new\ i} < F_j \quad (11)$$

In this case, the solution $X_{new\ i}$ replaces the solution X_j in the population.

Step 2: Local search:

New cuckoo eggs have a probability of being discovered.

If this is the case $(rnd\,(0,1) < p_a)$, new eggs are generated. Several methods can be envisaged for this. For CS, the following is the case where hybridization between elements of the randomly selected population is carried out:

$$X'_{new\ ij} = x_{ij} + rnd\,(0,1)(x_{pj} - x_{mj}), j = 1, \ldots, n, i = 1, \ldots, N_H)$$
$$F'_{new\ i} = f(X'_{new\ i}) \quad (12)$$

Where $m \in \{1,2,3, \ldots, N_H\}$ and $p \in \{1,2,3, \ldots, N_H\}$ such that $p = m$ are randomly chosen.

A selection identical to that of the previous step is then performed.

4) Pseudo-code of the Algorithm

The simplified pseudo code of the algorithm is given below.

Generation of an initial population of N_H host nests X_i

As long as the stopping criterion is not satisfied, repeat:

Generate a cuckoo by Levy Flight (10) and evaluate its cost function F_i,

Randomly choose a nest X_j among the N_H host nests, if $F_i < F_j$

Replace F_j with this new solution,

End If

A fraction p_a of the worst nests is dropped and others are generated instead (12), and make a selection.

Identify the best X_{best} solution

End As long as

Figure 3: Pseudo-code of the Firefly algorithm

C. Fitness Function for both Firefly and Cuckoo Search

1) Selection by an Aggregation of Objectives

The aggregation of the various objectives a weighted sum is the most used approach in the methods of selecting this category. Optimization method seeks to find a subset of minimum size characteristics that lead to a lower classification error a predicted threshold is (t). The function of fitness for an individual a_i is defined by:

$$fitness(a_i) = J(a_i) - (\mu j - \eta \sigma j) \quad (13)$$

Where μ and σ represent respectively the mean and the standard deviation of the values of j applied to all individuals in the population.ηrepresents a very small value which ensures that the$\min(fitness(a_i))$) is always positive. Finally the score $j(a)$ for an individual a is calculated by:

$$J(a) = length(a) - P(e(a)) \quad (14)$$

Where $length(a)$ is the number of bits that have "1" as a value in a and $P(e)$is a function of inequality on the classification error of the individual (e (a)). If the error rate is below the threshold t then $P(e)$is negative. The value of P increases exponentially depending of e. The function (e) is given by (to simplify the formula, we have replaced $e(a)$ by e which represents the error of classification of an individual (a) :

$$P(e) = \frac{\exp\left(\frac{(e-t)}{\xi}-1\right)}{\exp(1)-1} \quad (15)$$

Where ξ is a scaling parameter of small value (generally 0.01).

D. Fuzzy KNN Classifier

The closest neighbour method [16] is the simplest local nonparametric method. It is only assumed that the neighbourhood that influences the classification decision of the unknown point is limited to a ball containing only the nearest neighbour. The probability density of all classes is zero except for the one assigned to the nearest neighbour who is naturally chosen for the decision. This seemingly very simple method is robust. Its error is increased by two times the minimal error obtained by the Bayesian classification, under asymptotic conditions. It makes it possible to recognize shapes when the class distribution is not convex. The basic algorithm is easy to write and program. In return, it requires a large volume of training data, which means a large number of learning points to store and examine, that is to say high memory resources and a significant execution time in a raw implementation of the algorithm.

For several decades, researchers have tried to improve the method by reducing the amount of data needed and by optimizing algorithms to reduce the processing time (condensing), and by extending and enriching the process. Local context for refining the decision (editing rules). For example, is it appropriate to choose the nearest neighbour if the second and third closest neighbours belong to another class and are almost the same distance as the nearest neighbour. Moreover, the error or the uncertainty about the measurement of each prototype, the insufficient number of points learned, the teacher's labelling errors, are all causes that disturb the results.

Solutions to improve the recognition rate have been explored in two complementary directions: Enrich the neighbourhood analysis and refine the criterion of decision, correct the learning data.

1. Extend the neighbourhood to several points, for the purpose to refine the decision, which implies:
    a. The existence of a criterion or a function of decision-making: majority vote, Dudani distance weighting [17], blurred membership of Keller [18] and neighbourhood by class of Hattori [19].
    b. The choice of the number of neighbours k, or the exploitation of several values of k.
2. Correct Learning Data

    This article proposes a new decision rule (editing rule) derived from that of Keller, and is related essentially at the first point cited above. Alone the introduction of the angular distance is related to the second point. In order to better situate our contribution, we remember first the classic decision rules.

E. Main Decision Rules

The majority voter method presupposes the assumption of a uniform probability density at inside the k-neighbourhood, and a value of k sufficiently high for the estimated density to be stable. In practice, it is rare that these two conditions are fulfilled: indeed, the number of points learning is not only finished, but still it is desirable that it be limited for reasons of data to be collected, stored and used. In one lower density context, increase the number k leads to an increase in the volume of the neighbourhood, which weakens the hypothesis of uniform local distribution.

The optimal value of k expresses the best compromise which reconciles uniform distribution and little neighbourhood dense. However, when the representation is sparse, it becomes more plausible to assume that the Probability density is not uniform in the neighbourhood. Logically, the closer a neighbour is, the more it influences the decision. The weighting method seems the first to register in this optics. The w_i weighting of the xi point is expressed so:

$$w_i = \frac{d(x,x_k)-d(x,x_i)}{d9x,x_k)-d(x,x_1)} where\ d(x,x_i) \neq d(x,x_1) \quad (16)$$

$$v_i = 1\ where\ d(x,x_k) = d(x,x_1) \quad (17)$$

Wherex_1 and $x_k$are respectively the most near and far. Keller's method [6] proposes to compute for each point a degree of membership of the point unknown, depending on the distances of x to each prototypex_i, according to the wording:

$$f_{ci} = \frac{\sum_j u_{ij} d(x,x_j)^{\frac{2}{1-m}}}{\sum_{i,j} u_{ij} d(x,x_j)^{\frac{2}{1-m}}} \quad (18)$$

Where$u_{ij}$is the degree of belonging of the point x_j to the class C_i. In the case of net initialization, u_{ij} is 1 or 0 depending on whether the learned pointx_j belongs to the class or not C_i In the case of fuzzy initialization, u_{ij} is a degree fuzziness, associated with each prototype, which depends on an expert opinion, or the neighbourhood learned from prototype. The parameter m adjusts the influence of the distance.

The decision function depends on parameters to adjust and points of this k-neighbourhood. As part of an analytical approach, a formulation of the degree of belonging of the unknown point x to the class $C_i$is proportional to$1/\left(1+\frac{1}{b_i}\right)$, with:

$$b_i = \frac{N_i^{\alpha} v(x,x_i)^{\gamma} \sqrt{R(x_i)}}{d(x,x_i)^{\beta}} \quad (19)$$

Wherex_i is the point of the class C_i closest to x,N_i is the number of k-neighbours belonging to the class C_i, $R(x_i)$is the ranking rank of $x_i$in the k-neighbourhood, $v(x,x_i)$is a neighbourhood-dependent weighting restricted to$x_i$and x_j points closer than him, having for expression:

$$v(x,x_i) = \frac{d(x,x_i)}{\sum_{j=1,..i} d(x,x_j)} \quad (20)$$

Where α, β, γ are exponents to be adjusted by optimizing the recognition rate.

$\alpha = 1, \beta = 1, \gamma$Adjustable, to reduce to 1 the number of parameters to adjust.

As we can see, recent methods are introducing at least one parameter to be adjusted in addition to the number of neighbours k. We propose to modify the method of Keller by replacing the Euclidean distance with a distance that exploits the sub-neighbourhood of each prototype.

1) Sub-Neighbourhood Method

Principle of the decision rule the approach we propose is part of following postulate: in the case of low density, the contribution from a neighbouring prototype belonging to a same class is even better than:

- The prototype is near,
- There are more prototypes closer than him belonging to the same class.

The contribution of an internal prototype to the neighbourhood will be done using a different distance than the Euclidean distance, respecting the rule: an inner neighbourhood prototype of the same class artificially brings the prototype under consideration, especially since this internal prototype is close to unknown point. A prototype of another class does not affect not the prototype considered.

2) Formulation of the Decision Rule

In the following we will give the formulas a more great generality by attributing to a prototype x_m a fuzzy mass u_{im}, which denotes the degree of belonging of x_m to class C_i. In the case of net labelling of the Learning basis, $u_{ij}$is 1 if x_j belongs to the classC_i, otherwise 0.

We deduce principles previously stated a formula of the distance relative to the class C_i defined on the sub-

neighbourhood (x, x_j), denoted $D_i(x, x_j)$

$$D_i(x, x_j) = \frac{\sum_{x_m \in v(x,x_j)} u_{im}{}^{\alpha}(x,x_m)}{Card_{x_m \in ci, v(x,x_j)}(u_{im}, x_m)} \quad (21)$$

Let's detail the constituents of this definition of distance on the sub-neighbourhood. $d(x, x_j)$is the distance of the metric space, usually the Euclidean distance in R^n . $Card_{x_m \in ci, v(x,x_i)}(u_{im}, x_m)$represents the mass prototypes of class C_i in the sub-neighbourhood $D_i(x, x_j)$ is the arithmetic mean of the distances prototypes of class C_i

As it is obvious that:

$$d(x, x_m) \leq d(x, x_j) \forall x_m \in V(x, x_j) \quad (22)$$

the arithmetic mean of the distances verifies the inequality that characterizes virtual reconciliation:

$$d(x, x_1) \leq D_i(x, x_j) \leq d(x, x_j) \quad (23)$$

Let us write the formula of the degree of belonging of x to the class$C_i$with the same formalism as the distance of Keller (see relation (18)):

$$f_{ci} = \frac{\sum_j u_{ij} D_i(x,x_j)^{-\alpha}}{\sum_i \sum_j u_{ij} D_i(x,x_j)^{-\alpha}} \quad (24)$$

In this relation the Euclidean distance, which translates the influence of a single point has been replaced by a "Sub-neighbourhood distance".α adjusts the influence of the distance, as m in Keller's formula. The meaning of α is: if α is zero, we get the rule of the majority voter; if α is infinite, we get the nearest neighbour rule. In terms of storage of data, the sub-neighbourhood method is based, as Keller's method, on the distances table ordered from nearest neighbours. Only exploitation of this table is slightly more complex because it takes recurrently calculate the distance $D_i(x, x_j)$.Finally, note that in this method, such as Keller's method, the degree of belonging to a class is a monotone function of the neighbourhood:

$$V_{k,C_i}^{(1)}(x) \subset V_{k,C_i}^{(2)}(x) \rightarrow f_{c_i}^{(1)}(x) < f_{c_i}^{(2)}(x) \quad (25)$$

If, in a first neighbourhood, all prototypes are either closer or closer to the point unknown than in a second neighbourhood, and if the points other classes are invariant, the degree of membership of$x$ is greater in the case of the first neighbourhood.

EXPERIMENTAL SETUP

A. Diabetes Recognition

1) K-Fold Cross Validation

Cross validation is a statistical method that allows evaluation and comparison of learning algorithms by dividing the data into two segments: one used for learning the model and the other used for model validation. Typically during cross validation, the learning and validation sets must interchange in successive cycles so that each element of the database has a chance to be validated.

The most commonly used method of cross-validation is the "k-Fold Cross Validation" method. In this method the learning set is divided into k partitions of the same size. Subsequently, the learning and validation is done k times, with each iteration$i$ the partition k_i is used for the test and the remaining k-1 partitions are used for learning.

B. Database Used

The Pima Indian Diabetes [20] Foundation consists of 768 cases of which 268 are diabetic and 500 non-diabetic. Each case is made up of 9 attributes, 8 of which represent risk factors and the 9th represents the patient's class. Table 6 presents a description of these attributes:

No. Attribute Description attribute Average Standard deviation

1. Number of pregnancies (Ngross) 3.8 3.4
2. Plasma glucose concentration (mg / dl) 120.9 32.0
3. Diastolic blood pressure (mm Hg) (PAD) 69.1 19.4
4. Thickness of the skin at triceps (mm) (Epai) 20.5 16.0
5. Insulin level after 2 hours (mU¼ ml) (INS) 79.8 115.2
6. Body mass index (weight in kg / m ^ 2) (BMI) 32.0 7.9
7. Pedigree Function of Diabetes (Ped) 0.50 0.3
8. Age (years) 33.20 11.8

C. Performance Evaluation Parameters

- **Accuracy (AC)** is the proportion of the total number of predictions that were correct. It is determined using the equation:

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \quad (15)$$

Where

- **Precision (P)** is the proportion of the predicted positive cases that were correct, as calculated using the equation:

$$P = \frac{TP}{(TP+FP)} \quad (20)$$

Where TP is true positive, TN is true negative, FP is false positive and FN is false negative.

D. Simulation Parameter

1) Firefly Parameter

NumFireflies (Population size)	**30**
Maximum number iteration	**100**
Randomness	**0.2**
Attractiveness	**1**
Randomness reduction	**0.97**
Absorption coefficient	**0.2**

2) Cuckoo Search Parameter

(Population size)	**30**
Maximum number iteration	**100**
Probability of discovery of alien egg	**0.5**
Levy parameters	
Beta	**1.5**
Alpha	**1**

RESULTS

A. Accuracy

Table 1: KNN accuracy, KNN firefly and KNN cuckoo search accuracy

Database	KNN Features	Accuracy	Firefly-KNN Features	Accuracy	Cuckoo-KNN Features	Accuracy
Diabetes	8	68.4 %	8	75.23 %	8	76.9%
Diabetes2	19	66.086%	11	68.56 %	10	69.56 %

Table 2: Fuzzy KNN accuracy, Fuzzy KNN firefly and Fuzzy KNN cuckoo search accuracy

Database	Firefly-KNN Features	Accuracy	Firefly-Fuzzy-KNN Features	Accuracy	Cuckoo-Fuzzy-KNN Features	Accuracy
Diabetes	8	73.71 %	8	78.6 %	8	80.3 %
Diabetes2	19	67.8%	11	71.3%	10	74.8%

Table 3: Accuracy in percentage for Change of learning rate (50% training and 50 % testing) or K-foldverification

Database	KNN	Fuzzy KNN	Firefly KNN	Cuckoo KNN	Firefly-Fuzzy-KNN	Cuckoo-Fuzzy-KNN
Diabetes	67.44 %	71.6%	72.1%	72.11%	73.4%	75%
Diabetes2	62.3 %	62.6 %	66.8%	67.62%	68.4%	69.2%

Table 4: Accuracy in percentage for Change of learning rate (60% training and 40 % testing) or K-foldverification

Database	KNN	Fuzzy-KNN	Firefly-KNN	Cuckoo-KNN	Firefly-Fuzzy-KNN	Cuckoo-Fuzzy-KNN
Diabetes	67.44%	71.7%	72.91%	72.96%	74.4%	77.2%
Diabetes2	62.4%	62.6%	66.3%	67.6%	68.4%	69.3%

Table 5: Accuracy in percentage for Change of learning rate (70% training and 30 % testing) or K-foldverification

Database	KNN	Fuzzy KNN	Firefly-KNN	Cuckoo-KNN	Firefly-Fuzzy-KNN	Cuckoo-Fuzzy-KNN
Diabetes	67.8%	72.6%	72.9 %	73.04 %	74.4%	75.7%
Diabetes2	60.9%	61.4 %	63.7%	68.4%	68.5%	70.4%

Table 6: Accuracy in percentage for Change of learning rate (90% training and 10 % testing) or K-fold verification

Database	KNN	Fuzzy KNN	Firefly-KNN	Cuckoo-KNN	Firefly-Fuzzy-KNN	Cuckoo-Fuzzy-KNN
Diabetes	68.4%	73.7%	75.23 %	76.9 %	78.6%	80.3%
Diabetes2	66.086%	67.8%	68.56 %	69.56%	71.3%	74.8%

B. Precision

Table 7: Calculation of precision in percentage for two classes

Database	Classes	KNN	Fuzzy KNN	Firefly-KNN	Cuckoo-KNN	Firefly-Fuzzy-KNN	Cuckoo-Fuzzy-KNN
Diabetes	2	75 %	76.9%	76.94 %	77.6 %	78 %	81 %
Diabetes2	2	62 %	63.2 %	69.2	64 %	65 %	66 %

C. Recall

Table 8: Calculation of Recall in percentage for two classes

Database	Classes	KNN	Fuzzy KNN	Firefly-KNN	Cuckoo-KNN	Firefly-Fuzzy-KNN	Cuckoo-Fuzzy-KNN
Diabetes	2	72 %	76.9%	75.9%	76 %	78%	79 %
Diabetes2	2	61%	62.3%	63.2%	64.2%	65%	66%

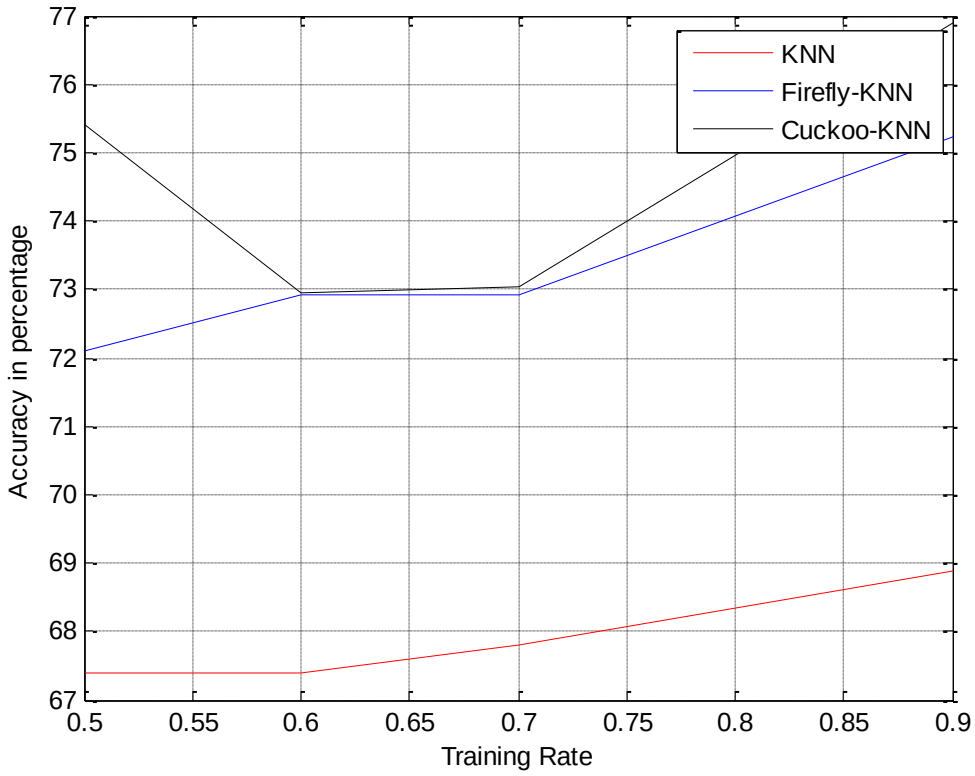

Figure 4: Accuracy vs. Learning rate (no of instances used for training) Diabetes 1 dataset

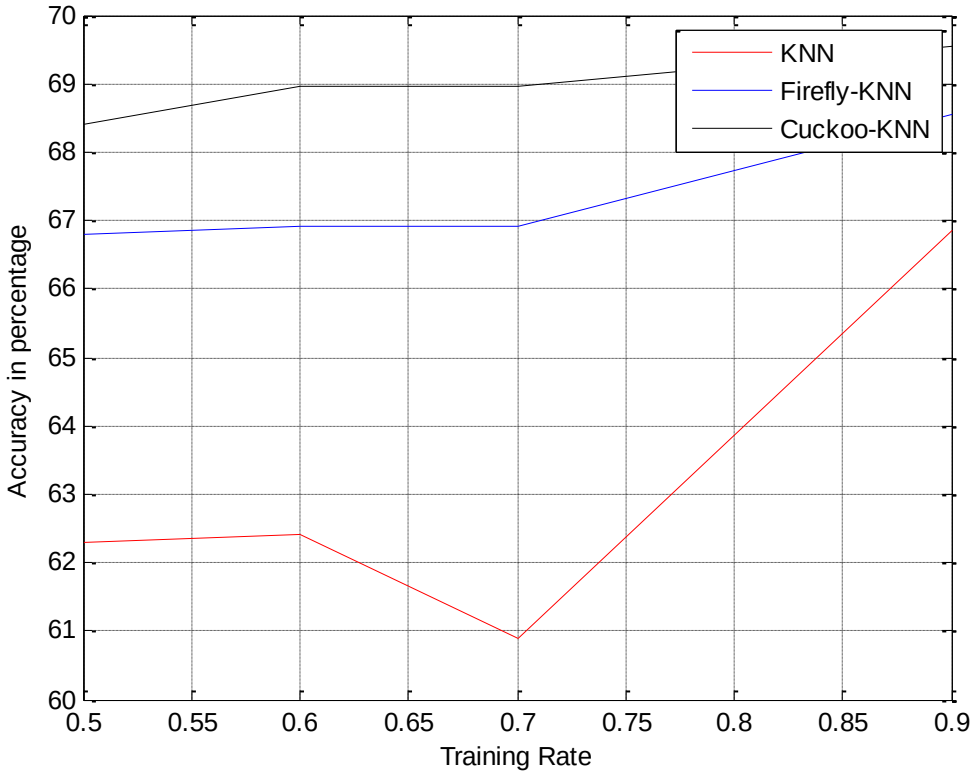

Figure 5: Accuracy vs. Learning rate (no of instances used for training) Diabetes 2 dataset

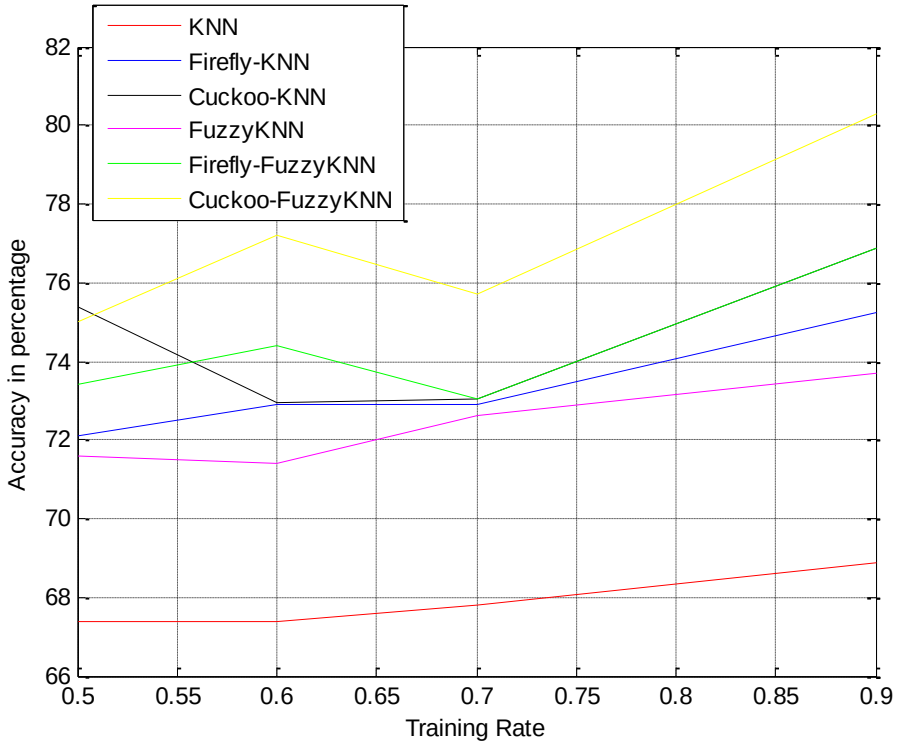

Figure 6: Accuracy vs. Learning rate (no of instances used for training) Diabetes dataset

D. Explanation

In this section, we present the different results obtained during the experimentation of our method of extracting attributes on the basis of UCI. In fact, our attribute extraction approach involves applying classification to group similar attributes. Following the formation of attribute groups similar, a center will be calculated and become itself the new extracted attribute. The optimal combination of the new attribute that will decide the classification quality observations later.

The classification performance of proposed method was evaluated using the KNN classifier. Originally there is 10 features used to train KNN and achieved accuracy is 61.9 % further this feature is selected using PSO, which is 6 feature for training and achieved accuracy is 98.5 %.

CONCLUSION

In this paper, we are interested in dimension reduction, in the framework of the automatic classification for a decision-making purpose, the industrial diagnosis among others. Our goal is to develop an extraction approach of attributes that can be applied for different types of data and different distributions.

It is based on the classification of attributes into different groupings. In each group, an optimal combination of these attributes and, will subsequently considered as a new representative attribute. New attributes thus constructed will be the new descriptors of the data and which will be brought at the learning and classification stage. The application of our approach on different UCI databases has made it possible to extract the good attributes for the classification in comparison with firefly and cuckoo search based methods. Proposed method outperform the traditional classification with increment of 21.1 %.By changing the different learning rate Fuzzy KNN and KNN is evaluated where Fuzzy KNN outperform then the traditional KNN method. Cuckoo search based approach fuzzy KNN gives 81% accuracy in diabetes 1database and 66% in diabetes 2 database.

REFERENCES

[1] Lorenzo, C., Williams, K., Hunt, K.J. and Haffner, S.M., 2007. The National Cholesterol Education Program–Adult Treatment Panel III, International Diabetes Federation, and World Health Organization definitions of the metabolic syndrome as predictors of incident cardiovascular disease and diabetes. *Diabetes care*, *30*(1), pp.8-13.

[2] Vlassara, H. and Uribarri, J., 2014. Advanced glycation end products (AGE) and diabetes: cause, effect, or both?. *Current diabetes reports*, *14*(1), pp.1-10.

[3] American Diabetes Association, 2014. Diagnosis and classification of diabetes mellitus. *Diabetes care*, *37*(Supplement 1), pp.S81-S90.

[4] Boland-Freitas, R., Howells, J., Lee, J., Liang, C., Corbett, A., Nicholson, G. and Ng, K., 2017. Sarcolemmal excitability attributes of the myotonic dystrophies as assessed by muscle velocity recovery cycles (mvrcs).

[5] Muja, M. and Lowe, D.G., 2014. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *36*(11), pp.2227-2240.

[6] Keller, J.M., Gray, M.R. and Givens, J.A., 1985. A fuzzy k-nearest neighbour algorithm. *IEEE transactions on systems, man, and cybernetics*, (4), pp.580-585.

[7] Naik, B., Nayak, J., Behera, H.S. and Abraham, A., 2016. A self-adaptive harmony search based functional link higher order ANN for non-linear data classification. *Neurocomputing*, *179*, pp.69-87.

[8] Heydari, M., Teimouri, M., Heshmati, Z. and Alavinia, S.M., 2016. Comparison of various classification algorithms in the diagnosis of type 2 diabetes in Iran. *International Journal of Diabetes in Developing Countries*, *36*(2), pp.167-173.

[9] Aishwarya, S. and Anto, S., 2014. A medical decision support system based on genetic algorithm and least square support vector machine for diabetes disease diagnosis.

[10] Purwar, A. and Singh, S.K., 2015. Hybrid prediction model with missing value imputation for medical data. *Expert Systems with Applications*, *42*(13), pp.5621-5631.

[11] Chetty, N., Vaisla, K.S. and Patil, N., 2015, May. An improved method for disease prediction using fuzzy approach. In *Advances in Computing and Communication Engineering (ICACCE), 2015 Second International Conference on* (pp. 568-572). IEEE.

[12] Gandomi, A.H., Yang, X.S., Talatahari, S. and Alavi, A.H., 2013. Firefly algorithm with chaos. *Communications in Nonlinear Science and Numerical Simulation*, *18*(1), pp.89-98.

[13] Yang, X.S., 2010. Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation*, *2*(2), pp.78-84.

[14] Yang, X.S., 2013. Multiobjective firefly algorithm for continuous optimization. *Engineering with Computers*, *29*(2), pp.175-184.

[15] Gandomi, A.H., Yang, X.S. and Alavi, A.H., 2013. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with computers*, *29*(1), pp.17-35.

[16] Borges, V. and Jeberson, W., 2016. Fuzzy kNN Adaptation to Learning by Example in Activity Recognition Modeling. In *Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics* (pp. 29-36). Springer, New Delhi.

[17] Huang, M., Lin, R., Huang, S. and Xing, T., 2017. A novel approach for precipitation forecast via improved K-nearest neighbor algorithm. *Advanced Engineering*

Informatics, *33*, pp.89-95.

[18] Gharaibeh, N.Y. and Alshorman, A.A., 2016. An Effective Diagnosis of Diabetic Retinopathy with Aid of Soft Computing Approaches. *David Publishing Company www. davidpublisher. com*, p.474.

[19] Hattori, K. and Gilmore, G., 2015. Vertical kinematics of the thick disc at 4.5≲ R≲ 9.5 kpc. *Monthly Notices of the Royal Astronomical Society*, *454*(1), pp.649-656.

[20] https://archive.ics.uci.edu/ml/datasets/pima+indians+diabetes.