An Improved Moth Flame Optimization Algorithm based on Rough Sets for Tomato Diseases Detection

Aboul Ella Hassanien[c,a], Tarek Gaber[b,c,1,*], Usama Mokhtar[c,d], Hesham Hefny[e]

[a]*Faculty of Computers and Information, Cairo University, Egypt*
[b]*Faculty of Computers and Informatics, Suez Canal University, Ismailia, Egypt*
[c]*Scientific Research Group in Egypt (SRGE), http://www.egyptscience.net*
[d]*Inst. of Stat. Studies and Res., Cairo University, Egypt*
[e]*Inst. of Stat. Studies and Res.,Cairo University, Egypt*

Abstract

Plant diseases is one of the major bottlenecks in agricultural production that have bad effects on the economic of any country. Automatic detection of such disease could minimize these effects. Features selection is a usual pre-processing step used for automatic disease detection systems. It is an important process for detecting and eliminating noisy, irrelevant, and redundant data. Thus, it could lead to improve the detection performance. In this paper, an improved moth-flame approach to automatically detect tomato diseases was proposed. The moth-flame fitness function depends on the rough sets dependency degree and it takes into a consideration the number of selected features. The proposed algorithm used both of the power of exploration of the moth flame and the high performance of rough sets for the feature selection task to find the set of features maximizing the classification accuracy which was evaluated using the support vector machine (SVM). The performance of the MFORSFS algorithm was evaluated using many benchmark datasets taken from UCI machine learning data repository and then compared with feature selection approaches based on Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) with rough sets. The proposed algorithm was then used in a real-life problem, detecting tomato diseases (Powdery mildew and early blight) where a real dataset of tomato disease were manually built and a tomato disease detection approach was proposed and evaluated using this dataset. The experimental results showed

* Corresponding author
Email addresses: aboitcairo@gmail.com (Aboul Ella Hassanien), tmgaber@gmail.com (Tarek Gaber), usamamokhtar@yahoo.com (Usama Mokhtar), hehefny@hotmail.com (Hesham Hefny)

[1] My present address is Faculty of Computers and Informatics, Suez Canal University, Ismailia, Egypt

that the proposed algorithm was efficient in terms of Recall, Precision, Accuracy and F-Score, as long as feature size reduction and execution time.

Keywords: moth flame optimization, rough set theory, particle swarm optimization (PSO) and genetic algorithms (GA), tomato's disease

1. Introduction

Plants are very crucial source of food and energy for humankind. Plant diseases can cause major economical, and ecological losses as well as reduction in both quantity and quality of agricultural products. Therefore, diagnosing and detecting plant diseases in a timely an accurate way is very important. Usually, the observation of experts using their naked eyes is the traditional approach followed in practice for the diagnosing and detection of plant diseases. Moreover, in some developing countries, small farmers could find difficulties to get experts making consulting these experts very expensive and time consuming. This could lead to the spreading of the disease into all crops. Thus, automatic/computer-based plant diseased detection approaches are of high importance.

The automatic detection system usually consists of two main phases. Firstly, the plant leaf image is captured using a digital camera. Secondly, the detection and classification of leaf diseases can be achieved through different steps: extracting the infected region, computing some features representing each disease and they classify these features to identify the diseases. The importance of automatic diagnosing and detection of plant diseases emerges as it could support benefits in monitoring big fields of crops, hence provide automatic detection of diseases based on the symptoms which appear on the plant leaves (24).

In last years, automatic detection of plant diseases attracts many researchers in different fields. Bauer et. al., (8), proposed an approach for the automatic classification of leaf (i.e.,sugar beet) diseases using high resolution multi-spectral and stereo images. In (36), Weizheng et al., introduced a new fast and accurate approach for grading plant diseases using computer image processing technique. They first used Otsu method to extract the leaf region, and then used Sobel operator to detect edges of the diseased spot. Finally, plant diseases are graded through the information of the quotient of disease spot and leaf areas as indicator. In another study (25), Naidu et al. suggested a method to identify virus infected grapevine using the discriminant analysis and they obtained a maximum accuracy of 81% of the classification results. Also, cotton diseases (10) were automatically identified using preprocessing operation and the use of SVM classifier to identify visual symptoms of cotton diseases. Moreover, in (20) a new method for wheat disease identification using image recognition was proposed. In this method, after computing features of diseased region of leaf images, samples are trained and recognized using the RBF-SVM classifier. In (29) to classify the leaf brown spot and the leaf blast diseases of rice plant, an automated system has been developed. This system is based on the morphological changes of the plants caused by the diseases and used the Bayes and SVM classifiers in the disease identification. Also an approach to detect the symptoms of nutrient diseases (4) was suggested and it is based on the vision system and pattern recognition.

The feature selection process is one of the most important tasks for pattern recognition and classification systems, e.g. plant disease detection system. The main goal of this process is to find a minimal feature subset from a problem domain such that to give a high accuracy in representing the original features (12). It improves the predictive accuracy of algorithms by reducing the number of features, removing irrelevant, noisy and redundant features. It is also helps in the improvement of the classification performance. The feature selection mechanism has been successfully employed to effectively solve classification problem in various areas, such as bioinformatics (32), image processing (31), data mining (22), pattern recognition (34), medical diagnosis (2; 33).

Different techniques were used to achieve feature selection. This includes the rough set theory (28) and bio-inspired techniques. The basic idea of using rough set-based for feature selection is to generate all possible feature reductions and then choose the one with minimal cardinality (19). The rough set has already used to accomplish a features selection task in different area such as: (13; 38; 6). Also, many bio-inspired methods have been used for feature selection process and thes include genetic algorithm (GA) (21; 27), ant colony optimization (ACO) (7; 1), Bat Algorithm (BA) (26; 30) and Grey Wolf Optimizer (GWO) (14).

Efforts have been targeted to combine the RS approach with bio-inspired algorithms to improve the performance. Bello et al. (9) proposed an feature selection approach which integrates Ant Colony System with rough set. The approach firstly generates a number of ants which are placed randomly on the graph and then they traverse edges probabilistically until a traversal stopping criterion is satisfied to output the best rough set reduct. This method achieved a high ratio in features reduction but the classification accuracy and execution time are not good enough. Similar to the Bello's approach (9), Wang et al., (35) introduced an approach integrating between rough set and the particle swarm optimization (PSO)to achieve the feature selection task. They followed the same idea but only applied PSO instead of ACS. Wang's approach was able to find the optimal reducts on most of the used datasets and minimizing the execution time. In another effort, Guo et al., (18) proposed an approach combining between Genetic Algorithm, GA, and rough set for the feature selection. Firstly, rough set was used to carry out the feature selection, then to find the optimal subset in the remaining feature subset, they used the GA improved with Population Clustering. The SVM (Support Vector Machines) was then applied to evaluate the effectiveness of the selected feature subset.

In this paper we proposed a Moth-Flame Optimization (MFO) and rough set (MFORSFS) approach for automatically detecting some kinds of tomato disease. The tomato was chosen to be the application of the automatic disease detection in this study because of its importance. It is ranked number one among 40 vegetables/fruits in terms of "relative contribution to human nutrition" and contains a high nutrition value. To achieve tomato disease detection, feature selection is a important phase. Thus, we first have introduced a new feature selection technique based on MFO and Rough Set called MFORSFS.

This MFORSFS was evaluated to prove its robustness and then we have used in the detection of the tomato diseases. The proposed MFORSFS algorithm was compared against using (1) Particle Swarm Optimization (PSO) and (2) Genetic Algorithm (GA) with the rough sets. The results showed that the MFORSFS gave a higher accuracy of classification results while preserve low number of features compared to the other two optimization algorithms.

The rest of this paper is organized as follows: Section 2 gives an overview of the moth flame optimization and rough sets. Section 3 presents the details of the proposed system. In Section 4, experimental results and discussion are given. Finally in Section 5, conclusions and future work are presented.

2. Preliminaries

2.1. Gabor Features

Gabor filter-base method is an effective method for extracting texture feature. It has been used in many applications such as biometrics and segmentation. Gabor filters are known as convolution kernel, the product of a cosine and Gaussian functions. It enjoys the characteristic of specified orientation and spatial frequency. The 2-D Gabor filter is like a local band-pass filter with some localization properties in the spatial and frequency domain. Gabor filter is proved his efficiency in characterizing texture features (17), like in our case: extracting texture features from tomato's leaves.

A 2D Gabor function $g(x,y)$ is defined as follows:

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)+2\pi jWx\right] \tag{1}$$

where σ_x and σ_y characterize the spatial extent and frequency bandwidth of the Gabor filter, and W represents the frequency of the filter. Let $g(x,y)$ be the mother generating function for the Gabor filter family. A set of different Gabor functions $g_{m,n}(x,y)$ can be generated by rotating and scaling $g(x,y)$ to form an almost complete and non-orthogonal basis set, that is,

$$g_{m,n}(x,y)) = a^{-2m}g(x',y')) \tag{2}$$

Where $\acute{x} = a^{-m}(x\cos\theta_n + y\sin\theta_n)$,$\acute{y} = a^{-m}(-x\sin\theta_n + y\cos\theta_n)$, $a > 1$, $\theta_n = n\pi/K$, $m = 0, 1, \ldots, S-1$, and $n = 0, 1, \ldots, K-1$. Parameter S is the total number of scales, and parameter K is the total number of orientations. So, S and K represents the total number of generated functions.

Given an image $I(x, y)$, its Gabor-filtered images are

$$G_{m,n}(x, y) = \sum_{x_1}\sum_{y_1} I(x_1, y_1)g_{m,n}(x - x_1, y - y_1)) \tag{3}$$

2.2. Feature Selection Overview

In the past few decades, classification problems resolved using machine learning techniques usually contains high dimensional of data. Such high dimensionality lead to challenges such as the curse of dimensionality or a large number of features. These challenges tends to overfit problem which results in performance degeneration. To address this problem, feature selection has been introduced. The main purpose of feature selection is to determine a minimal feature subset of a problem domain such that retaining a suitably high accuracy in representing the original features (12).

According to using labeled or unlabeled training set, feature selection techniques can be classified into unsupervised (10), supervised (?), and semi-supervised feature selection (?). The supervised methods could be further categorized into wrapper-based methods, filter-based methods, and embedded-based methods. The wrapper-based methods, e.g., WLD (?), makes use of the predictive accuracy of a given learning algorithm to evaluate the quality of selected features. The filter-based methods, e.g. (11) depend on using some measures representing the general characteristics of given training data such as consistency, distance, dependency, and correlation. The embedded-based methods are a combination between the filter-based and wrapper-based methods. They firstly involve the statistical criteria, like the case of filter-based methods, to select a number of candidate features subsets having a particular cardinality. The embedded-based methods then choose the subset having the highest classification accuracy (?).

2.3. Rough set basics

Rough set theory (37) is a mathematical approach to imprecision, vagueness and uncertainty. Rough Set Attribute Reduction (RSAR) (11) provides a filter-based tool for extracting feature from a domain in a concise way whilst reducing the amount of knowledge involved. To formalize the rough set, consider $\mathbf{I} = (\mathbf{U}, \mathbf{A})$ is an information system, where $\mathbf{U}$ is a non-empty set of finite objects (the universe) and $\mathbf{A}$ is a non-empty finite set of attributes such that for $\forall a \in \boldsymbol{A}$ determines a function $f_a : \boldsymbol{U} \rightarrow \boldsymbol{V}_a$. With any $\boldsymbol{P} \subseteq \boldsymbol{A}$, there is an associated equivalence relation $\boldsymbol{IND(P)}$:

$$\boldsymbol{IND(P)} = \{(x, y) \in \boldsymbol{U} \times \boldsymbol{U} \mid \forall a \in \boldsymbol{P},\ f_a(x) = f_a(y)\} \tag{4}$$

The partition of $\mathbf{U}$, generated by $\mathbf{IND}(\mathbf{P})$, is denoted $\mathbf{U}/\mathbf{P}$. The equivalence classes of the $\mathbf{P}$-indiscernibility relation are denoted $[x]_p$. The indiscernibility relation is the mathematical basis of rough set theory.

Let $\boldsymbol{X} \subseteq \boldsymbol{U}$, the $\mathbf{P}$-lower approximation $\underline{\boldsymbol{P}}X$ and $\mathbf{P}$-upper approximation $\overline{\boldsymbol{P}}X$ of set Xcan be defined as:

$$\underline{\boldsymbol{P}}\boldsymbol{X} = \{x \in \boldsymbol{U} \mid [x]_P \subseteq \boldsymbol{X}\} \tag{5}$$

$$\overline{\boldsymbol{P}}\boldsymbol{X} = \{x \in \boldsymbol{U} \mid [x]_P \cap \boldsymbol{X} \neq \phi\} \tag{6}$$

Let $\boldsymbol{P,Q} \subseteq \boldsymbol{A}$ be equivalence relations over $\mathbf{U}$, then the positive, negative and boundary regions can be defined as:

$$\boldsymbol{POS}_P(\boldsymbol{Q}) = \bigcup_{X \in U/Q} \underline{\boldsymbol{P}}\boldsymbol{X} \tag{7}$$

$$\boldsymbol{NEG}_P(\boldsymbol{Q}) = \boldsymbol{U} - \bigcup_{X \in U/Q} \overline{\boldsymbol{P}}\boldsymbol{X} \tag{8}$$

$$\boldsymbol{BND}_P(\boldsymbol{Q}) = \bigcup_{X \in U/Q} \overline{\boldsymbol{P}}\boldsymbol{X} - \bigcup_{X \in U/Q} \underline{\boldsymbol{P}}\boldsymbol{X} \tag{9}$$

The positive region of the partition $\mathbf{U}/\mathbf{Q}$ with respect to $\mathbf{P}(\boldsymbol{POS}_P(\boldsymbol{Q}))$, is the set of all objects of $\mathbf{U}$ that can be certainly classified into blocks of the partition. An important issue in attribute reduction is discovering dependencies between attributes. $\mathbf{U}/\mathbf{Q}$ by means of $\mathbf{P}$

For $\boldsymbol{P},\boldsymbol{Q} \subseteq \boldsymbol{A}$, we say that $\mathbf{Q}$ depends on $\mathbf{P}$ in a degree k $(0 \leq K \leq 1)$ denoted $\boldsymbol{P} \Rightarrow_k \boldsymbol{Q}$, if

$$k = \gamma_p(\boldsymbol{Q}) = \frac{|\boldsymbol{P}OS_p(\boldsymbol{Q}|)}{|\boldsymbol{U}|} \tag{10}$$

If $k = 1$, $\mathbf{Q}$ depends totally on $\mathbf{P}$, if $0 < k < 1$, $\mathbf{Q}$ depends partially (in a degree k) on $\mathbf{P}$, and if $k = 0$ then $\mathbf{Q}$ does not depend on $\mathbf{P}$.

In a decision system, an attribute set includes two sets: decision attribute set D and condition attribute set C, i.e. $A = C \subset D$. The degree of dependency between these two sets, $\gamma_C(\boldsymbol{D})$, which is known as the quality of approximation of classification, is induced by the decision attributes set (37) .

When $\mathbf{P}$ is a set of condition attributes and $\mathbf{Q}$ is the decision, $\gamma_p(\boldsymbol{Q})$ is the quality of classification (37). The goal of attribute reduction is to remove redundant attributes so that the reduced set provides the same quality of classification as the original. A reduct is defined as a subset $\mathbf{R}$ of the conditional attribute set $\mathbf{C}$ such that $\gamma_R(D)$= $\gamma_C(D)$. The set of all reducts is defined as:

$$\boldsymbol{Red} = \{\boldsymbol{R} \subseteq \boldsymbol{C} | \gamma_R(\boldsymbol{D}) = \gamma_C(\boldsymbol{D}), \forall \boldsymbol{B} \subseteq \boldsymbol{R}, \gamma_B(\boldsymbol{D}) \neq \gamma_C(\boldsymbol{D})\} \tag{11}$$

In rough set attribute reduction, a reduct with minimal cardinality is the one being searched for. To locate a single element of the minimal reduct set $\boldsymbol{Red}_{min} \subseteq \boldsymbol{Red}$, the following equation is used :

$$\boldsymbol{Red} = \left\{\boldsymbol{R} \in \boldsymbol{Red} | \forall \boldsymbol{R}' \in \boldsymbol{Red}, |\boldsymbol{R}| \leq |\boldsymbol{R}'\right\} \tag{12}$$

The intersection of all reducts is called the core, the elements of which are those attributes that cannot be eliminated. The core is defined as:

$$\boldsymbol{Core}(\boldsymbol{C}) = \cap \boldsymbol{Red} \tag{13}$$

———————————————————————

2.4. Moth Flame Optimization

Moth-Flame Optimization (MFO) is a new optimization algorithm which simulate the moths navigation manner in nature. The main inspiration of this optimizer is the navigation method of moths in nature called transverse orientation (23). It is a population-based evolutionary computation search technique which mimics the behavior of moths in their special navigation methods at night. The idea of the MFO is based on a mechanism called transverse orientation for navigation in night throw the moon light. Using this mechanism, moth flies with a fixed angle with respect to the moon. When moths see a human-made artificial light, they try to maintain a similar angle with the light to fly in straight line. Since such a light is extremely close compared to the moon, maintaining a similar angle to the light source causes a useless or deadly spiral fly path for moths (15).

The mathematical model for the MFO is based on two components, moth and flame. The moths are actual search for agents that move around the search space, whereas flames are the best position of moths that obtains so far. As mentioned above the inspiration of this algorithm is the transverse orientation. In order to mathematically model this behaviour, the position of each moth is updated with respect to a flame using the following equation:

$$\boldsymbol{M}_i = \boldsymbol{S}(\boldsymbol{M}_i, \boldsymbol{F}_j) \tag{14}$$

where $\boldsymbol{M}_i$ indicates the *i-th* moth, $\boldsymbol{F}_j$ refers to the *j-th* flame, and $\boldsymbol{S}$ is the spiral function. The logarithmic spiral for the MFO algorithm is defended as follows:

$$\boldsymbol{S}(\boldsymbol{M}_i, \boldsymbol{F}_j) = \boldsymbol{D}_i \cdot \boldsymbol{e}^{bt} \cdot \cos(2\pi t) + \boldsymbol{F}_j \tag{15}$$

Where $\mathbf{D}_i$ indicates the distance of the *i-th* moth for the *j-th* flame and is as defined in 16, b is a constant for defining the shape of the logarithmic spiral, and t is a random number in [-1, 1]. $\mathbf{D}$ is calculated as follows:

$$\boldsymbol{D}_i = |\boldsymbol{F}_j - \boldsymbol{M}_i| \tag{16}$$

Where $\mathbf{M}_i$ indicates the *i-th* moth, $\mathbf{F}_i$ denotes the *j-th* flame and $\mathbf{D}_i$ refer to the distance between $\mathbf{M}_i$ and $\mathbf{F}_i$.

The $\boldsymbol{t}$ parameter in the spiral equation 15 controls the direction of moth navigation around the flame. ($\boldsymbol{t}$ = -1 is the closest position to the flame, while $\boldsymbol{t}$

= 1 shows the farthest) The spiral equation allows a moth to fly around a flame and not necessarily in the space between them. Therefore, the exploration and exploitation of the search space can be guaranteed.

In order to further emphasize exploitation, $\boldsymbol{t}$ is defined as random number in $[\boldsymbol{r}, 1]$ where $\boldsymbol{r}$ is linearly decreased from -1 to -2 over the course of iteration. According to equation 15,each moth is restricted to move towards a flame that may lead to local optimum stagnation. In order to prevent this, at each iteration, a list of flames must be updated and sorted based on their fitness values. The moths then update their positions with respect to their corresponding flames. Since the position updating of moths with respect to n different locations in the search space may degrade the exploitation of the best promising solutions, an adaptive mechanism for the number of flames has been proposed as in the following formula:

$$\boldsymbol{flame\ no} = \boldsymbol{round}\left(\boldsymbol{N} - \boldsymbol{l} * \frac{\boldsymbol{N}-1}{\boldsymbol{T}}\right) \tag{17}$$

where $\boldsymbol{l}$ is the current number of iteration, $\boldsymbol{N}$ is the maximum number of flames, and $\boldsymbol{T}$ indicates the maximum number of iterations.

3. The proposed MFO-based rough set tomato diseases detection approach

The proposed MFO-based rough set tomato diseases detection approach is comprised of five fundamental phases: image acquisition, pre-processing, feature extraction, feature selection and finally classification. These phases are described in details below. The overall architecture of the proposed system is illustrated in Figure 1.

3.1. Image acquisition phase

The first phase of the proposed MFO-based rough tomato diseases detection approach is the image acquisition phase. This phase plays an important role in any image classification system. These images must select carefully to achieve

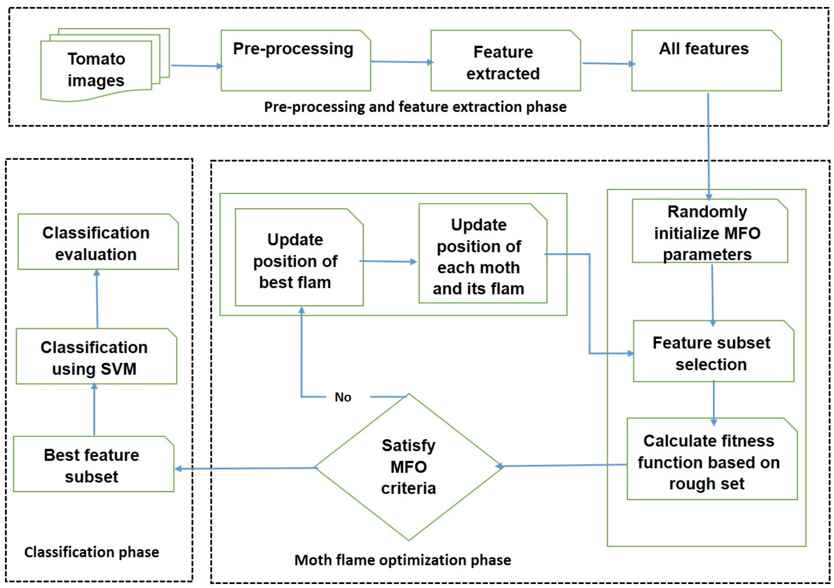


Figure 1: Layout structure of the proposed MFO-based rough set approach

the intended task. The datasets used for experiments were constructed based on real sample images of tomato leaves infected with two types of tomato diseases including Powdery mildew and early blight. this dataset were collected from different farms using sonny digital camera with 14 MP resolution, at temperature between 16 and 20 degree. Fig. 2 illustrates some examples of these dataset.

3.2. Pre-processing Phase

In this phase, after collecting the dataset, the images were enhanced by removing noise that caused by defects of camera flash or hight lights to increase the efficiency of classification and prediction process. Firstly, every leaf was isolate and extract in single image. Secondly, captured images were resized to 512 x 512 resolution, thus minimizing the storage capacity and reduce the computational time in the post-processing. Finally, the background of each image was removed using background subtraction technique with some morphological operations. Gaussian Mixture-based Background/Foreground Segmentation Algorithm (39) was used to subtract the background and morphological techniques

Figure 2: Samples of infected tomato using in this work

(dilation followed by erosion) to remove noise.

3.3. Feature extraction phase

In this phase, Gabor transform was used to describe the textural pattern of diseased tomato leaves. The total number of extracted features are 402. For more details of this phase reader can refer to (24).Each of used Gabor filters was implemented as a 8 x 8 convolution mask for each of its real and imaginary components. The acquired images were converted to HSV color space and 6 components of the image (R,G,B,H,S,V) have been extracted. To construct feature vector of each image components; a vector of 64 length was obtained from the average output for every i^{th}filter. Vector of 3 length consisted of: cost function $J(i)$, maximum average output D^i_{max} and minimum average output D^i_{min}. At the end of this step feature vector of (64+3) x 6 = 402 length that describe the image has been obtained.

3.4. Moth flame based features selection phase

As it was mentioned above, the output of the feature extraction phase is 402 features. Such large number of features usually contains irrelevant and redundant features. To achieve the feature selection phase, the MFO algorithm was employed through using both of rough set and SVM classifier as a fitness function for the MFO to evaluate the best set of features helping achieving the highest accuracy. The MFO algorithm was adopted in this paper for the following reasons. Firstly, in the original paper introducing the MFO (23), it is reported that the MFO algorithm has advantages on other related algorithms such as PSO, GA, and GSA in the context of optimization problems. Secondly, it is proved that MFO has the ability to solve real problem such as marine propeller design (so it could be useful algorithm in our case too (the detection of tomatoâĂŹs diseases). Thirdly, the MFO convergence is guaranteed since the moths always have the habit of updating their positions according to flames which are the most promising solutions.

The overall proposed MFO based rough set feature selection algorithm is described in Algorithm 1.

In the MOF-rough-set feature selection approach, the solution space represents all possible selections of features selection. Each moth position represents binary selection of feature sets of length $\boldsymbol{N}$, where $\boldsymbol{N}$ is the total number of attributes. Every bit represents an attribute where the value '1' means that the corresponding attribute is selected while '0' means it is not selected. Each position is an attribute subset. The frequency of a position updating for each moth is represented as a positive integer, varying between 1 and max-update. It implies how many of the moth's bits (features) should be changed, at a particular moment in time.

The maximum range of position updating serves is a constraint to control the global exploration ability of a moth. After many tests, it was found that an appropriate maximum of position updating of each moth value is $(1/3)^{*}\boldsymbol{N}$. Also, this maximum range was proven to achieve good results as reported in (35). Figure 1 illustrates the Layout structure of the proposed MFO-based rough set approach.

It is important to highlight the used parameters in the feature selection approach, as given in Table 1. The parameters in this table are selected based

Algorithm 1 MFO based rough set feature selection algorithm

Initialize MFO parameters
for (i=1 : No. of moth) **do**
  Initialize the population of solutions by formula (18)
  Evaluate the fitness of each moth by formula (19)
end for
Sort the first population of moths
Update the position of best flame obtained so far
while (Iter < MaxIter and GFlamFit < MaxFit) **do**
  Update flame number by formula 17
  Sort moths according their fitness values and assign the values of the first value (highest accuracy results)
  Update flames positions according to the moth
  Decrease the parameter a from -1 to -2
  for (i=1 : No. of moth) **do**
    Update position of each moth (feature set) restricted into the region [1,N/3] by formula. 15
    Update position of each flame with respected to the best moth
  end for
  Evaluate the fitness of each moth by using the following formula: (19)
  $Rough-sets-fitness-function = \alpha * \gamma_R(\boldsymbol{D}) + \beta * \frac{|\boldsymbol{C}|-|\boldsymbol{R}|}{|C|}$
  iter = iter + 1;
end while
Produce the best flame position

on the ones in [PSO-Rough Set] where our method is very close to it and it is also compared with our proposed method and below.

For the population initialization: The population initialization mechanism was used in the proposed algorithm and in all PSO and GA based ones using in the experimental evaluation, see Section 4. When population is randomly initialized, a feature subset (solution) should be produced randomly by

Table 1: Parameters values used in experiments

Parameters	**MFORSFS**	**PSO**	**GA**
No. of Population	30	30	30
No. of Generation	50	50	50
Velocity	1∼ $(1/3) * N$	1∼ $(1/3) * N$	
weight	$1.4 \sim 0.4$	$1.4 \sim 0.4$	
Mutation probability			0.4
Crossover probability			0.6

the following expression

$$X_{ij} = \begin{cases} 1 & rand() > 1 \\ 0 & otherwise \end{cases} \tag{18}$$

Where $i \in \{1, 2, ...PN\}$ and $j \in \{1, 2, ...FN\}$, where PN is population size and FN is number of feature.

For the fitness function: it was a measure to determine the goodness or quality of a single solution in a population. At the end of each iteration, fitness value is calculated of each agent for evaluating quality search. In this paper, classification accuracy was adopted as fitness function and the Support vector machine SVM classifier was used to evaluate the performance of each solution. The classification accuracy obtained was based on the average of the 10-fold cross validation method. Since we must take into account two important issues, the classification quality and feature subset length. So, the fitness function is calculated according to the following equation:

$$Rough - sets - fitness - function = \alpha * \gamma_R(\boldsymbol{D}) + \beta * \frac{|\boldsymbol{C}| - |\boldsymbol{R}|}{|C|} \tag{19}$$

Where γ_R is the classification quality of condition attribute set $\mathbf{R}$ relative to decision $\mathbf{D}$, $|\mathbf{R}|$ refer to the length of elected attribute subset. $|\mathbf{C}|$ is the total number of features. α and β are two parameters corresponding to the importance of classification quality and subset length, $\alpha \in [0, 1]$ and $\beta = 1 - \alpha$. We adopt this approach based on the work done in (35), they states that classification quality is more significance than the size of subset, as a result both parameters have been set as follow: $\alpha = 0.9$, $\beta = 0.1$.

3.5. SVM-based classification phase

In the classification phase, the SVM was employed to assess whether features selected using MFORSFS method can help in detecting infecting tomato leaves. The inputs of this phase are trained feature vectors, whereas the outputs are the decision of whether the tomatoâĂŹs leaf is infected or not and if it is infected, it determines the type of disease (Powdery mildew and early blight). It is worth to mention that the SVM was used in two different phases. In the feature selection, it was used as a fitness function to evaluate which set of features is best to represent the leaf (infected or healthy). In the classification phase, the SVM was also used to classify between the infected and healthy leaves.

To evaluate the performance of a classification system, the k-cross-validation, a common method to deal with small training sets in machine learning (3), was used. Cross-validation is a method to evaluate classifier or predictive models. In this method, the original sample is partitioned into two sets: a training set to train a given model, and another test set to evaluate this model. The general type of this method is k-fold cross-validation in which the original sample is divided randomly into k subsamples of equal size. From all these k subsamples, one subsample is used as the validation data to test the model while the remaining $k - 1$ subsamples are used as training data. The process of the k-fold cross-validation is repeated k folds (times) where each k subsamples is used as the validation data only one time. The main advantage of this validation method is that all samples are used for both training and validation, and each samples is used for validation exactly once.

4. Experimental Results and Discussion

To evaluate the proposed approach, two main scenarios were designed and tested. The first scenario was for the evaluation of the MFO-Rough-Set based feature selection approach using benchmark datasets. Also, in this scenario, to make the MOF+rough set feature selection approach comparable with related work, PSO and GA were also combined with the rough sets to achieve the feature selection. The three proposed features selection algorithms (MOF+rough set, PSO+rough sets, and GA+rough sets) were compared with each other to select the best one to choose a suitable combination of features in wrapper mode for maximizing classification performance and minimizing the data dimensionality. To make the results of the three algorithms are comparable, it was important to unify bases for all adopted bio-inspired algorithms. Thus, Population Initialization, Fitness Function are setup as described in Section 3.4 and the other parameters given in 1. All adopted bio-inspired algorithms were initialized identically and the used fitness function was the same.

In the second scenario, the performance of the overall MFO-rough-set based tomato diseases detection approach was investigated. Three sub-scenarios were also designed here. Firstly, a simple classifier, KNN, was used a fitness function of MFO and its results were compared to the SVM-based ones. Secondly, a traditional feature selection, i.e., mRMR, was used to select the best features and the classification results were reported and compared with our proposed method. Thirdly, three features selection algorithms (MOF+rough set, PSO+rough sets, and GA+rough sets) were applied in the feature selection phase to choose the best one. All algorithms were implemented using MatLab R2014b and all experiments were run under a computer with Intel(R) Core (TM) i7 CPU Q820@1.73 GHZ and 8 GB memory and the system is Windows 8 Professional.

To evaluate the results in both the mentioned scenarios, several measurements were used. These measurements are Accuracy, specificity, Recall and F-Score. They are defined mathematically at Equations (20), (21),(22) and (23) respectively (16). Using multi-level confusion matrix, each measure were calculated for each class, then the overall value were calculated on average of all classes.

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{20}$$

$$specificity = \frac{TN}{TN + FP} \tag{21}$$

$$Recall = \frac{TP}{TP + FN} \tag{22}$$

$$F - Score = \frac{2 * TP}{2 * TP + FP + FN} \tag{23}$$

4.1. Evaluating the proposed MFO-Rough-Set feature selection approach

To test our proposed feature selection approach, dataset from the UCI data repository (5) was used, Table 2 summarizes the 6 used data set for further experiments.

Table 2: Description of the data sets used in experiments

Dataset	**No. of samples**	**No. of features**	**No. of classes**
Adult	20	4	2
Iris	150	4	3
Zoo	101	16	7
Soybean-small	47	35	4
Lung	32	56	3
heart-scale	270	13	2
Monks	432	6	2

To evaluate the proposed MFO-Rough-Set selection algorithm, the average classification accuracy of the selected feature subsets was used and it was measured using the 10-fold cross-validation method was used. This means that all values were verified ten times to ensure the reliability of the experiment.

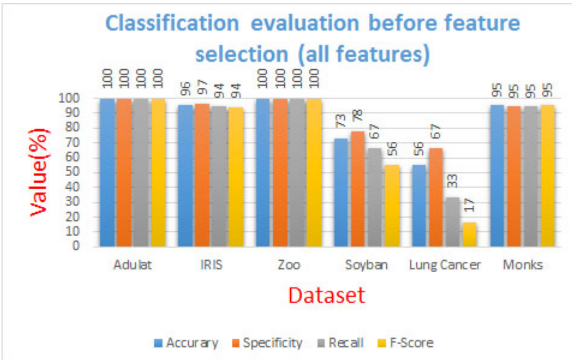
The dataset was randomly separated into 10 segments. In each iteration, one segment was selected as test data (nonrepetitively) and the others were used as training data. To obtain a value of classification accuracy, the average of the results in each iteration was calculated. All of the experimental results are averaged over the 10 runs of 10-fold Cross-Validation.

In this experiment, all of MFO-Rough-Set, PSO-Rough-set, and GA-Rough-Set were tested on the 6 datasets mention above for selecting the best subset of features that effectively describe the dataset. As we mention before, several measurements are used to evaluate the performance of the proposed features selection algorithms. Table 3 shows the number of features selected in the best solution obtained for each optimization technique. As it can be observed from this table, the best obtained results produced from the new MFO feature selection algorithm that for most of the used dataset. Also the number of features resulted after using the new MFO feature selection algorithm always smaller than (or equal in some cases) other algorithms.

Table 3: Number of features selected for each optimization technique

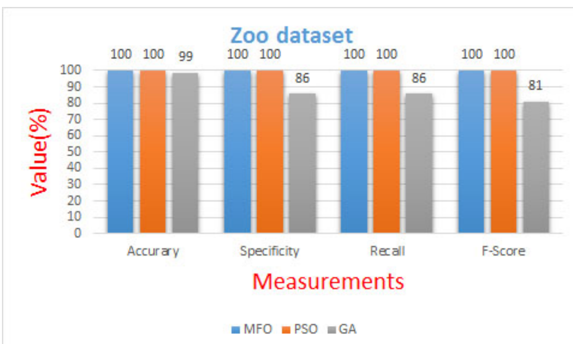
Dataset	**MFORSFS**	**PSO**	**GA**
Adult	2	2	2
Iris	1	1	1
Zoo	4	5	4
Soybean-small	2	2	5
Lung	14	30	20
Monks	3	3	3

Also in terms of the classification accuracy, Figure (3:a) the accuracy results before applying any feature selection (i.e. using all features) for all datasets. While figures (3:b-f) demonstrates the comparison, in terms of Accuracy, Recall, Precision and F-Score, results of classification evaluation after using the three feature selection algorithms. From these results, it can be seen that the classification evaluation results of the Monks dataset are the same as the Adult

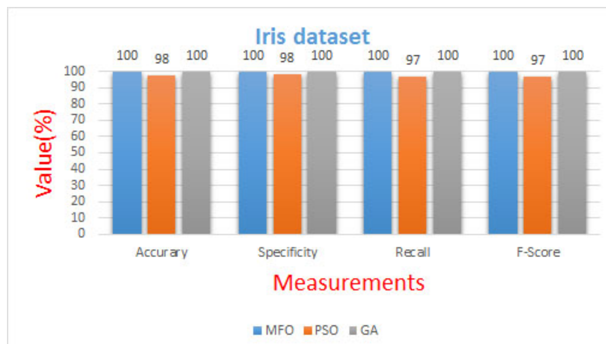


a: Whole dataset before feature selection

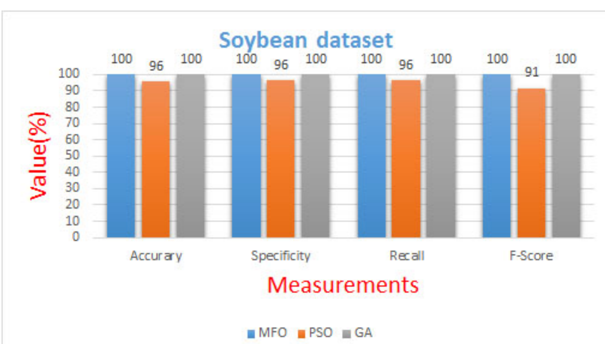
b: Adults dataset



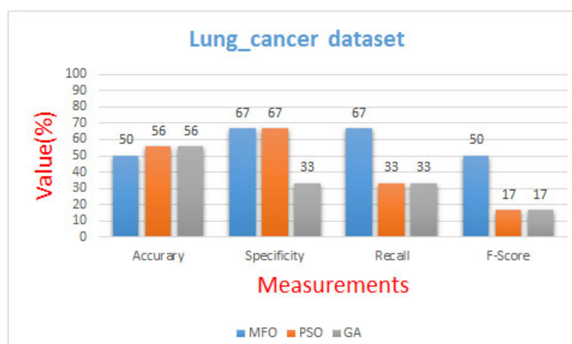
c: Zoo dataset



d: Iris dataset



e: Soybean dataset



f: Lung cancer dataset

Figure 3: Comparison between the results before and after employing the MFO, PSO, and GA based features selection algorithms using different datasets in terms of Accuracy, Precision, Recall and F-Score

dataset. As it can be observed from figures (3), the best obtained results produced from the new MFO feature selection algorithm that for most of the used dataset Table 4 listed computational time in seconds regarding the optimization algorithms for the feature selection task.

Table 4: computational time in seconds regarding the optimization algorithms for the feature selection task

Dataset	**MFORSFS**	**PSO**	**GA**
Adult	45	19	81
Iris	173	52	271
Zoo	76	29	160
Soybean-small	52	18	105
Lung	56	24	103
Monks	1.6383 e^3	945	4.5012 e^3

4.2. Evaluating MFO-based tomato Diseases Detection Approach

To assess the performance of the proposed MFO-based tomato diseases detection approach, firstly a real dataset of diseased tomato leaves were collected. Then, a set of features describing the diseased tomato leaves were extracted. These features were in a $m \times n$ matrix, where $m = 200$ is the number of used leaves and $n = 402$ is the number of features that describe each leaf. Three sub-scenarios were also designed here. Firstly, a simple classifier, KNN, was used a fitness function of MFO and its results were compared to the SVM-based ones. Secondly, a traditional feature selection, i.e., mRMR, was used to select the best features and the classification results were reported and compared with our proposed method. Thirdly, three features selection algorithms (MOF+rough set, PSO+rough sets, and GA+rough sets) were applied in the feature selection phase to choose the best one.

4.2.1. SVM-based vs KNN-based Fitness Function

Both of the SVM and KNN classifiers were used in the evaluation of the quality of the MFORSFS methods. Two kernel functions (RBF, and Polynomial) of the SVM were used and KNN with k=1,3,5, and 7 were also used. A comparison were also conducted between the two classifiers and the results are summarized in Table (5), and (6).

From Table (5), it can be noticed that when using the KNN as a classifier with k=5, the highest results 87%, in terms of accuracy, precision and recall, was obtained from features were selected using with MFORS when its parameters are KNN with k= 5

Table (5), it could be seen that the highest results, 91.5%, in terms of accuracy, precision and recall, was obtained using: SVM-Polynomial as a classifier from the feature selected by MFORSFS method with KNN is a fitness function and $k = 5$.

From Table (6) and (5), it can be noticed that SVM-based classification, applied to the MFORSFS-based features with KNN as fitness function, gave better results than that of the KNN-based ones. Where latter gave accuracy at 90.5% while the latter gave accuracy at 87

4.2.2. MFORSFS-based features vs mRMR-based features

A traditional feature selection, i.e., mRMR, was used to select the best features and the classification results were reported and compared with our proposed method. The mRMR experiments, four sets of features (first 50,100,150, 200) were evaluated and the results are summarized in Table (7). From this table, it can be noticed that the highest accuracy results 90.5%, was obtained from using the first 200 features ranked by mRMR when classified by the SVM-Polynomial.

Based on the obtained results and the results of our method in), it can be noticed that our method is better than mRMR-based results.

From Table (Table (5) and (Table (7), it can be noticed that the MFORS-based classification results (91.5%) is better than that of the mRMR-based

Table 5: Classification results using KNN classifier when the KNN (with different k values) and SVM-linear-Kernel were used as fitness function in the features selection phase

K value	Number of samples in each class	Feature selection method	Accuracy	Precision	Recall
1	(2*100,2*100)	MFORS-KNN with k=1	84%	84.2%	84%
3	(2*100,2*100)	MFORS-KNN with k=1	85%	85.1%	85%
5	(2*100,2*100)	MFORS-KNN with k=1	87%	87.1%	87%
7	(2*100,2*100)	MFORS-KNN with k=1	84%	84.2%	84%
1	(2*100,2*100)	MFORS-KNN with k=3	83%	83.2%	83%
3	(2*100,2*100)	MFORS-KNN with k=3	83%	83.1%	83%
5	(2*100,2*100)	MFORS-KNN with k=3	85%	85%	85%
7	(2*100,2*100)	MFORS-KNN with k=3	83%	84.1%	83%
1	(2*100,2*100)	MFORS-KNN with k=5	83%	83.1%	83%
3	(2*100,2*100)	MFORS-KNN with k=5	83.5%	83.5%	83.5%
5	(2*100,2*100)	MFORS-KNN with k=5	86%	86%	86%
7	(2*100,2*100)	MFORS-KNN with k=5	84%	84.1%	84%
1	(2*100,2*100)	MFORS-KNN with k=7	83.5%	83.7%	83.5%
3	(2*100,2*100)	MFORS-KNN with k=7	85%	85.1%	85%
5	(2*100,2*100)	MFORS-KNN with k=7	86.5%	86.5%	86.5%
7	(2*100,2*100)	MFORS-KNN with k=7	84.5%	84.7%	84.5%
1	(2*100,2*100)	MFORS-SVM with linear kernel	85.5%	85.8%	85.5%
3	(2*100,2*100)	MFORS- SVM with linear kernel	85%	85.1%	85%
5	(2*100,2*100)	MFORS- SVM with linear kernel	86.5%	86.6%	86.5%
7	(2*100,2*100)	MFORS- SVM with linear kernel	85%	85.1%	85%

results (90.5%, the highest results in (Table (7). Both these results are obtained using the same kernel functions (polynomial) of the SVM classifier. So, it could be claimed that our proposed method is better than the mRMR, the traditional

Table 6: Classification results using SVM classifier when KNN (with different k values) and SVM-linear-Kernel were used as fitness function in the features selection phase.

SVM kernel function	Number of samples in each class	Feature selection method	Accuracy	Precision	Recall
RBF	(2*100,2*100)	MFORS-KNN with K=1	82.5	84.7	82.5
polynomial	(2*100,2*100)	MFORS KNN with K=1	89	89	89
RBF	(2*100,2*100)	MFORS-KNN with K=3	82.5	84.7	82.5
polynomial	(2*100,2*100)	MFORS KNN with K=3	90	90	90
RBF	(2*100,2*100)	MFORS-KNN with K=5	83	85.8	83
polynomial	(2*100,2*100)	MFORS KNN with K=5	91.5	91.5	91.5
RBF	(2*100,2*100)	MFORS-KNN with K=7	83	85	83
polynomial	(2*100,2*100)	MFORS KNN with K=7	91	91.1	91
RBF	(2*100,2*100)	MFORS-SVM with linear function	81.5	83.3	81.5
polynomial	(2*100,2*100)	MFORS-SVM with linear function	90.5	90.5	90.5

feature selection method.

4.2.3. MFO-Rough-Set vs PSO-Rough-set vs GA-Rough-Set

PSO-Rough-set, GA-Rough-Set, The MFO-Rough-Set (our proposed method) feature selection algorithms were applied to select a number of features and to produce classification accuracy. This was done to compare the performance of our method in comparison with the related methods. Figure 4 illustrates this comparison between these algorithms in terms of Accuracy, Precision, Recall and F-Score. Figures (4: a) and (4: b) summarize the comparison results before and after employing the three features selection algorithms to original tomato's features (i.e., the 402 Gabor features). Also, figures (4: c) and (4: d) demon-

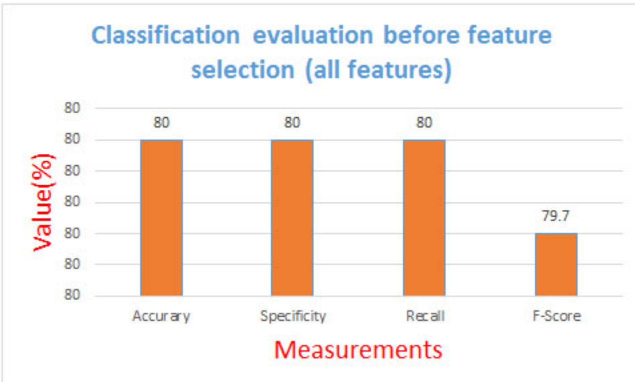
Table 7: Classification Results Using SVM classifier when using mRMR features (first 50, 100,150, and 200 features)

SVM kernel function	**Number of samples in each class**	**Feature selection method**	**Accuracy**	**Precision**	**Recall**
RBF	**(2*100,2*100)**	**mRMR (first 50 features)**	**86**	**87.2**	**86**
polynomial	**(2*100,2*100)**	**mRMR (first 50 features)**	**85**	**85.1**	**85**
RBF	**(2*100,2*100)**	**mRMR (first 100 features)**	**84.5**	**85.5**	**84.5**
polynomial	**(2*100,2*100)**	**mRMR (first 100 features)**	**89.5**	**89.5**	**89.5**
RBF	**(2*100,2*100)**	**mRMR (first 150 features)**	**83**	**84.7**	**83**
polynomial	**(2*100,2*100)**	**mRMR (first 150 features)**	**89.5**	**89.5**	**89.5**
RBF	**(2*100,2*100)**	**mRMR (first 200 features)**	**83.5**	**85.5**	**83.5**
polynomial	**(2*100,2*100)**	**mRMR (first 200 features)**	90.5	**90.5**	textbf90.5

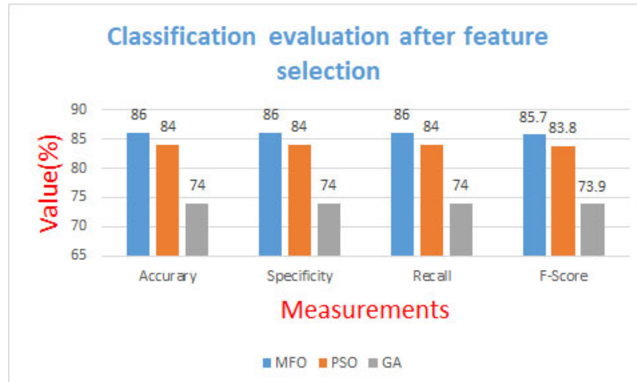
strates a comparison between the three methods in terms of the final reduct size and execution time, respectively.

From, Figure 4, it can be noticed that the MFO-based selection algorithm gave the best results for the classification evaluation, and in the execution time. Although, MFO-based method came the second in the reduct size (after the GA-based one), it gave the best in the classification performance and this is the most important in our case. The good performance of the MFO-based approach could be explained by the exploration power of the MFO and the the high performance of rough sets for the feature selection. Where the MFO algorithm uses the $\boldsymbol{t}$ parameter of the spiral equation 15. This parameter controls the direction of moth navigation around the flame, thus allowing each moth to fly around flame sand not necessarily in the space between them. Consequently, the exploration and exploitation of the search space can be guaranteed.

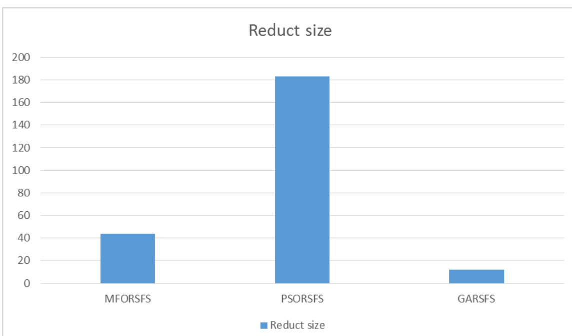
Although the database was manually built in this study, an automatic process could be achieved as in the following scenario. A mobile app could be de-



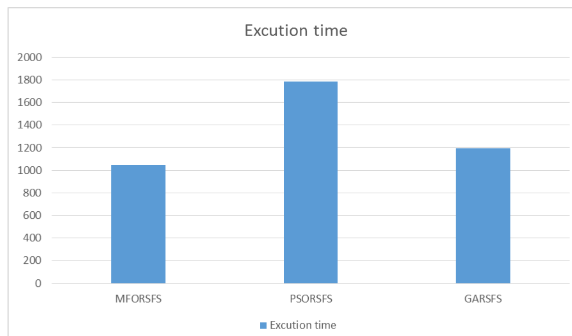
a: Classification accuracy before feature selection



b: Classification accuracy after feature selection



c: No. of best features



d: Computational time in seconds

Figure 4: Visualization for the results of MFORSFS-based tomato diseases detection approach

volved and deployed to trained farmers who can take picture of infected tomato leaf and send it to a server. On this server, the proposed algorithm could be implemented to achieve the disease detection task and then reply to the mobile app (i.e., to the farmer) with the disease name/type or no disease

5. Conclusions and Future Work

In this paper, a new approach for tomato diseases detection called MFO-based rough set tomato diseases detection approach was introduced. In this approach, a now algorithm for feature selection (i.e. MFORSFS) was proposed, implementedm, and evaluated. This approach is a combination of the MFO and the rough set and used in the dimension reduction phase of the tomato diseases detection approach. Firstly, the MFORSFS was tested on well defined 6 datasets obtained from the UCI machine learning data repository and it was found that MFO-based approach outperformed PSO and GA-based ones. The MFORSFS was then employed the tomato disease detection approach to reduce the number of features to the ones that can effectively describe each leaf of the diseased tomatoes. The MFORSFS algorithm was compared against feature selection based on PSO and GA. It was found that MFORSFS gave much better performance, robustness and faster convergence. In the future, our approach could be improved by applying other parameters selection algorithms for best parameter values selection.

References

[1] Aghdam, M., Ghasem-Aghaee, N., 2008. Application of ant colony optimization for feature selection in text categorization. 2008 IEEE Congress. URL http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=4631182

[2] Ali, A., Couceiro, M., Anter, A., 2014. Evaluating an evolutionary particle swarm optimization for fast fuzzy c-means clustering on liver ct images. Computer Vision and.
URL https://www.google.com/books?hl=en{&}lr={&}id=NBKXBQAAQBAJ{&}oi=fnd{&}pg=PA1{&}dq=evaluating+an+evolutionary+particle+swarm+optimization+for+fast+fuzzy+c-means+Clustering+on+Liver+CT+Images{&}ots=ThKHdrFeyB{&}sig=IqzUnQ25BuIQOo9fb5M8WKlGIVc

[3] Arlot, S., Celisse, A., 2010. A survey of cross-validation procedures for model selection. Statistics surveys 4, 40–79.
URL http://projecteuclid.org/euclid.ssu/1268143839

[4] Asraf, H. M., Nooritawati, M., Rizam, M. S., 2012. A Comparative Study in

Kernel-Based Support Vector Machine of Oil Palm Leaves Nutrient Disease. Procedia Engineering 41, 1353–1359.

[5] Asuncion, A., Newman, D., 2007. UCI machine learning repository. URL https://ergodicity.net/2013/07/

[6] Aziz, M. A. E., Hassanien, A. E., jul 2016. Modified cuckoo search algorithm with rough sets for feature selection. Neural Computing and Applications, 1–10.
URL http://link.springer.com/10.1007/s00521-016-2473-7

[7] Basiri, M., Ghasem-Aghaee, N., Aghdam, M., 2008. Using ant colony optimization-based selected features for predicting post-synaptic activity in proteins. European Conference on.
URL http://link.springer.com/10.1007/978-3-540-78757-0{_}2

[8] Bauer, S., Korc, F., Förstner, W., 2009. Investigation into the classification of diseases of sugar beet leaves using multispectral images. EJ van Henten, D. 9, 229—-238.
URL http://www.wageningenacademic.com/doi/pdf/10.3920/978-90-8686-664-9{#}page=230

[9] Bello, R., Nowe, A., Caballero, Y., Gómez, Y., Vrancx, P., 2005. A model based on ant colony system and rough set theory to feature selection. In: Proceedings of the 2005 conference on Genetic and evolutionary computation - GECCO '05. ACM Press, New York, New York, USA, p. 275.
URL http://portal.acm.org/citation.cfm?doid=1068009.1068053

[10] Camargo, A., Smith, J., 2009. An image-processing based algorithm to automatically identify plant disease visual symptoms. Biosystems Engineering 102 (1), 9–21.

[11] Chouchoulas, A., Shen, Q., oct 2001. Rough set-aided keyword reduction for text categorization. Applied Artificial Intelligence 15 (9), 843–873.

URL http://www.tandfonline.com/doi/abs/10.1080/088395101753210773

[12] Dash, M., Liu, H., 1997. Feature Selection for Classification. Intelligent Data Analysis 1 (3), 131–156.

[13] Emary, E., Yamany, W., Hassanien, A. E., dec 2014. New approach for feature selection based on rough set and bat algorithm. In: 2014 9th International Conference on Computer Engineering & Systems (ICCES). IEEE, pp. 346–353.
URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7030984

[14] Emary, E., Zawbaa, H. M., Grosan, C., Hassenian, A. E., 2015. Feature Subset Selection Approach by Gray-Wolf Optimization. Springer International Publishing, pp. 1–13.
URL http://link.springer.com/10.1007/978-3-319-13572-4{_}1

[15] Frank, K., Rich, C., Longcore, T., 2006. Effects of artificial night lighting on moths. Ecological consequences of.
URL https://www.google.com/books?hl=en{&}lr={&}id=dEEGtAtR1NcC{&}oi=fnd{&}pg=PA305{&}dq=++++++Effects+of+artificial+night+lighting+on+moths.{%}5Crq{%}5Crq{~}++++{&}ots=83{_}ad6b9eM{&}sig=cFViFOg4kNnN3QbTDgH8Ak2d2ps

[16] Geyer, L. H., DeWald, C. G., oct 1973. Feature lists and confusion matrices. Perception & Psychophysics 14 (3), 471–482.
URL http://www.springerlink.com/index/10.3758/BF03211185

[17] Grigorescu, S. E., Petkov, N., Kruizinga, P., 2002. Comparison of texture features based on gabor filters. IEEE Transactions on Image processing 11 (10), 1160–1167.

[18] Guo, Y., Wang, B., Zhao, X., Xie, X., Lin, L., Zhou, Q., aug 2010. Feature selection based on Rough set and modified genetic algorithm for intrusion

detection. In: 2010 5th International Conference on Computer Science & Education. IEEE, pp. 1441–1446.
URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5593765

[19] Komorowski, J., Øhrn, A., 1999. Modelling prognostic power of cardiac tests using rough sets. Artificial Intelligence in Medicine 15 (2), 167–191.

[20] Liu, L., Zhang, W., Shu, S., Jin, X., 2013. Image recognition of wheat disease based on rbf support vector machine. In: Proceedings of International Conference on Advanced Computer Science and Electronics Information (ICACSEI 2013), Supported by the Key Technology Projects of Anhui Province, China. No. 1201a0301008.

[21] Lu, J., Zhao, T., Zhang, Y., 2008. Feature selection based-on genetic algorithm for image annotation. Knowledge-Based Systems.
URL http://www.sciencedirect.com/science/article/pii/S095070510800097X

[22] Lutu, P., Engelbrecht, A., 2010. A decision rule-based method for feature selection in predictive data mining. Expert Systems with Applications.
URL http://www.sciencedirect.com/science/article/pii/S0957417409005831

[23] Mirjalili, S., 2015. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. Knowledge-Based Systems 89, 228–249.

[24] Mokhtar, U., Ali, M. A. S., Hassenian, A. E., Hefny, H., dec 2015. Tomato leaves diseases detection approach based on Support Vector Machines. In: 2015 11th International Computer Engineering Conference (ICENCO). IEEE, pp. 246–250.
URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7416356

[25] Naidu, R. A., Perry, E. M., Pierce, F. J., Mekuria, T., 2009. The potential of spectral reflectance technique for the detection of Grapevine leafroll-associated virus-3 in two red-berried wine grape cultivars. Computers and Electronics in Agriculture 66 (1), 38–45.

[26] Nakamura, R. Y. M., Pereira, L. A. M., Costa, K. A., Rodrigues, D., Papa, J. P., Yang, X.-S., aug 2012. BBA: A Binary Bat Algorithm for Feature Selection. In: 2012 25th SIBGRAPI Conference on Graphics, Patterns and Images. IEEE, pp. 291–297.
URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6382769

[27] Oliveira, L. S., Sabourin, R., Bortolozzi, F., Suen, C. Y., sep 2003. A Methodology for Feature Selection Using Multiobjective Genetic Algorithms for Handwritten Digit String Recognition. International Journal of Pattern Recognition and Artificial Intelligence 17 (06), 903–929.
URL http://www.worldscientific.com/doi/abs/10.1142/S021800140300271X

[28] Pawlak, Z., oct 1982. Rough sets. International Journal of Computer & Information Sciences 11 (5), 341–356.
URL http://link.springer.com/10.1007/BF01001956

[29] Phadikar, S., Sil, J., Das, A., 2012. Classification of Rice Leaf Diseases Based onMorphological Changes. International Journal of Information 2 (3), 460.
URL http://search.proquest.com/openview/72e1db851a71867d848c4f03e5dbaa2b/1?pq-origsite=gscholar

[30] Rodrigues, D., Pereira, L. A., Nakamura, R. Y., Costa, K. A., Yang, X.-S., Souza, A. N., Papa, J. P., 2014. A wrapper approach for feature selection based on Bat Algorithm and Optimum-Path Forest. Expert Systems with Applications 41 (5), 2250–2258.

[31] Salama, M. A., Hassanien, A. E., Fahmy, A. A., Kim, T.-h., 2011. Heart Sound Feature Reduction Approach for Improving the Heart Valve Diseases Identification. Springer Berlin Heidelberg, pp. 280–290.
URL http://link.springer.com/10.1007/978-3-642-27183-0{_}30

[32] Salama, M. A., Hassanien, A. E., Mostafa, A., dec 2016. The prediction of virus mutation using neural networks and rough set techniques. EURASIP Journal on Bioinformatics and Systems Biology 2016 (1), 10.
URL http://bsb.eurasipjournals.springeropen.com/articles/10.1186/s13637-016-0042-0

[33] Sayed, G. I., Soliman, M., Hassanien, A. E., 2016. Bio-inspired Swarm Techniques for Thermogram Breast Cancer Detection. Springer International Publishing, pp. 487–506.
URL http://link.springer.com/10.1007/978-3-319-33793-7{_}21

[34] Sweidan, A., El-Bendary, N., Hassanien, A., 2015. Water Quality Classification Approach based on Bio-inspired Gray Wolf Optimization. In 7th IEEE International Conference of Soft Computing and Pattern Recognition, 13 – 15.
URL http://scholar.cu.edu.eg/sites/default/files/abo/files/water{_}quality{_}classification{_}approach{_}based{_}on{_}bio-inspired{_}gray{_}wolf{_
pdf

[35] Wang, X., Yang, J., Teng, X., Xia, W., Jensen, R., 2007. Feature selection based on rough sets and particle swarm optimization.

[36] Weizheng, S., Yachun, W., Zhanliang, C., Hongda, W., 2008. Grading Method of Leaf Spot Disease Based on Image Processing. In: 2008 International Conference on Computer Science and Software Engineering. IEEE, pp. 491–494.
URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4723305

[37] Z., P., S., R., 1991. Theoretical aspects of reasoning about data. Nonvell, MA: Kluwer.

[38] Zawbaa, H. M., Hassanien, A. E., Emary, E., Yamany, W., Parv, B., dec 2015. Hybrid flower pollination algorithm with rough sets for feature selection. In: 2015 11th International Computer Engineering Conference (ICENCO). IEEE, pp. 278–283.
URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7416362

[39] Zivkovic, Z., van der Heijden, F., 2006. Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern recognition letters 27 (7), 773–780.