**META-HEURISTIC OPTIMIZATION ALGORITHMS BASED FEATURE SELECTION FOR CLINICAL BREAST CANCER DIAGNOSIS**

ABSTRACT :

The paper offers a crossbreed streamlining algorithm combining harmony search (HS) and simulated annealing (SA) known as harmony search and simulated annealing (HS-SA) for precise and accurate breast malignancy. Additionally, an improved wavelet-based contourlet transform (WBCT) system for feature extraction explores to get the highlights of the region of interest (ROI), permitting performance improvement over other standard methodologies. In the mined feature space, the projected HS-SA algorithm intends to diminish the feature di- mensions and congregate at the unprecedented feature set. The SVM classifier backed with diverse kernel functions is used for classification, which is fed by the chosen features, and its exhibition contrasts with the conventional machine learning classification and optimization techniques. The actualized computer-aided diagnosis (CAD) learning mechanism is challenged by evaluating its findings. It examines two different breast mammographic datasets i) benchmark BCDR-F03 dataset and ii) local mammographic dataset. Trial re- productions, empirical outcomes, and measurable examinations likewise indicate that the proposed model is practical and advantageous for the arrangement of malignant breast growth. The findings show that the proposed CAD framework (HS-SA + kernel SVM) is better than different characterization accuracy procedures (with an accuracy of 99.89% for the local mammographic dataset and 99.76% for benchmark BCDR-F03 dataset, AUC of 99.41% for the local mammographic dataset and 99.21% for reference BCDR-F03 dataset), while keeping the feature space limited to just seven feature subsets and computational prerequisites as low as is prudent.

INTRODUCTION :

Body cells change exponentially when they are influenced by ma- lignant growth in breast cancer, inevitably turning into a protuberance or mass of tumor. Most of the diseases of breast cancer arise, in partic- ular, in the segments of milk-creating organs, called lobules, and chan- nels linked to lobules, and finally to the nipple. The breast includes greasy, connective, and lymphatic tissues.

Comparing to 22% of the new cases every year, breast disease is the most serious of all types of malignant growth, as it has occurred at a rapid pace. There are more than 200 kinds of cancer, but the pace and velocity by which breast cancer affects women worldwide is incomparable. Among US women in 2017, an expected 2,52,710 new instances of obtrusive breast malig- nancy, 63,410 new instances of breast carcinoma in situ, and 40,610 breast disease passing were witnessed as per the International Agency for Research on Cancer Disease (IARC) report. Breast disease rep- resents 23% of the all-out malignant growth cases and 14% of the malignant growth demise in both developed and developing nations. The estimate is that more than 1.6 million new cases of breast cancer have occurred among women worldwide in 2010 alone.In 2011, almost 1.7 million people were targeted for breast disease. Measure- ments say that in the USA, 527 new instances of breast disease were analyzed every day, and 110 individuals died each day. In 2016–2017 insights, as per the American Cancer Society (ACS), many new instances of breast malignancy were expected in the US. It incorporates 18.3% of all malignant growth types in Egypt. In the event that a woman lives at the age of 85, there is one in eight possibilities (12%) for her to develop breast disease at some point in her life, because the risk of malignant breast growth increases significantly, paying little attention to her family ancestry as a woman ages.Indian females are most provoked by breast cancer by being the number one malignant growth among them, which compels the likewise inclination to its risks for a developing country. Factual research has shown a mortality of 12.7 per 100,000 women in parallel to an age adjusted rate as high as 25.8 per 100,000 women. In 2015–2016, women were influenced by malignant breast growths, and the passing rate was 50 per cent as communicated by the findings from the Indian Council of Medical Research (ICMR). Pakistan has to face an excess of 40,000 deaths annually because of it. After coronary disease and mis- haps, malignant growth is the third cause of death in Iran, accounting for 24.6 percent of all cancers. The average age of a woman who has breast cancer is 49.6 years.

Malignancy is the primary source of death around the world, representing 8.2 million in 2012. It is normal that the number of cases of annual disease will rise from 14 million in 2012 to 22 million in the next two decades.Malignant growth control and far-reaching avoidance plan are hence vital.Interim, more accentuation ought to be put on the early inference of maladies to add to the patient’s lifetime.The timely seizure of breast cancer has significance for endurance, especially in low-wage countries where assets are extremely con- strained, and the diagnosis of late-stage illness makes the burden more troublesome.

Mammography screening of asymptotic women is a proven non-invasive technique for decreasing mortality from breast malignant by as much as 30%. Woman above 40, should undergo mammography screening once a year as recommended by the ACS. Quite often, mammogram discovers something that resembles malignancy, that re- sults in false positive (FP). Also, radiologists may miss up to 30% of breast malignant growths contingent upon the thickness of the breast. In light of World health organization’s (WHO) report, around 33% of disease is treated through early analysis unfurling tthe occurrence of breast cancer can be stopped provided early diagnosis becomes a reality. Improving malignancy anticipates a variety of techniques, that aims to enable the individuals to follow successful strategies for the prevention of disease.All these compelling factors have convinced a great deal of research over the last decades, focusing on the advancement of computational frameworks to assist the doctor in deciphering radiological pictures. These CAD frameworks have increased space in current medications, filled in as a data hotspot for authorities, and expanded the pace of correct discovery in the recognizable evidence of genuine infections, such as breast malignancy.

Notwithstanding, examinations found in writing utilize similar strategies and setups for both thick and non-thick masses, while those systems could be progressively fitting for a particular sort of thickness. Magnetic resource imaging (MRI), self and clinical breast checks, ultrasound, and mammography are some of the screening strategies utilized for malignant breast growt. The most accurate and straightforward system for distinguishing breast cancer is in favour of mammography. Film mammography is superseded by advanced digital mammography, where outstanding mechanized hard- ware is used to record patient breast images and for additional treat- ments such as detection and classification. Microcalcifications and masses are the most well-known anomaly that stimuli breast malignant growth.The healthcare data analytics shifted the conventional healthcare services to recent evidence and cure based soon after the birth of ma- chine learning and pattern recognition fields. The analysis of biomedical data (biomedical images like MRI, CT scan, PET, US, mammography, and biomedical signals like EEG, ECG, EMG) has led to the design of automated and smart CAD systems that help to diagnose early, accurate and precise diseases even before symptoms are visible externally. With the development of artificial intelligence techniques, the race of data- driven intelligent classification approaches have been applied for breast cancer diagnosis, such as Naive Bayesian , Neural Network , Support Vector Machine (SVM) , ensemble methods , K-means , fuzzy and rough set techniques , PSO , semi-supervised techniques , deep learning , transfer learning Active learning or other hybrid algorithms .

In 2011, B. Zhang et al. proposed a random Subspace cascade with rejection options for the classification of microscopic biopsy images. A. E. Hassanien and T. Kim in 2012 familiarized an amalgam method that cartels the rewards of fuzzy sets, pulse coupled neural networks (PCNNs), and support vector machine, in conjunction with wavelet-based feature for best cancer classification. In 2013, V. Balanica et al. presented four new methods for mining the specu- lation feature of a perceived breast lesion on mammography. The au- thors in Ref. et al. displayed a fast-orthogonal search (FOS) that delivers competent iterative way to compute step by step regression, and can select features with a predictive value from a set of kinetic and texture candidate features computed from dynamic contrast-enhanced magnetic resonance images. L. Taifi et al.in 2014 presented a preprocessing method, grounded on homomorphic filtering and wavelet, for the removal of irregularities in mammographic images. W. Sun et al. planned a three-stage Semi-Supervised Learning (SSL) method for refining presentation of computerized breast cancer analysis with undiagnosed data. In 2015, N. P. P ́erez et al. offered a new feature selection method (named uFilter) that advances the Man- n–Whitney U test for tumbling dimensionality and ranking features in binary classification problems. In 2015, authors et al. proposed a novel local energy-based shape histogram (LESH) as the feature set for the appreciation of irregularities in mammograms. In 2016, H. Kong et al. proposed Jointly Sparse Discriminant Analysis (JSDA) to sightsee the main factors in breast cancer to enlighten the accuracy in diagnosis and prediction. In 2017, W. Sun et al. developed a graph-based semi-supervised learning (SSL) scheme using a deep con- volutional neural network (CNN) for breast cancer diagnosis. In 2018, S. Liu et al. utilized a Bayesian network (BN) modelling approach for breast cancer. In 2019, L. Tsochatzidis et al. investigated Deep convolutional neural networks (CNNs) in the context of computer-aided diagnosis (CADx) of breast cancer. Q. Xu et al. in 2019 developed a CAD based on a CNN network that aims to classify breast mass lesions in optical tomographic images. T. A. Shaikh et al. in 2020, proposed a LUPI-based CAD framework for breast cancer using privileged infor- mation that enhances the performance of a single-modal imaging-based CAD for breast cancer by relocating PI.

This research aims to diagnose breast cancer based on the charac- teristics of the extricated tumor. The classifiers are a vital ingredient in data mining methods whose quality of performance is fully reliant on the feature extraction and selection. The presentation of the estimator, either in terms of learning speed, generalization dimensions or straight forwardness, progressed the right choice of the features founded on classification and clustering methods, numerous approaches is applied for breast cancer diagnosis in the recent literature. However, the bulk rise in the amount of available data (both features and records) from recent years has constrained the old methodologies, and thus, the meta- heuristics approaches employ feature selection and dipping the number of features grew its usage in the data mining field. The current investi- gation condenses the prevailing feature space to diminish the compu- tational cost for SVM training and preserve a parallel diagnosis accuracy. To excerpt beneficial information and diagnose the tumour, a hybrid of HS-SA search and support vector machine (K-SVM) algorithms is developed. The HS-SA algorithm is exploited to distinguish the unseen designs of the benign and malignant tumours distinctly and discover the best features for clinical breast cancer diagnosis. Seven subsets of ab- stract tumour features are mined from the original 21 features subsets for the training phase. Despite the fact that the K-SVM lessens dimen- sionality of input feature space, the high prediction accuracy is sus- tained. From the computation time perspective, the proposed strategy diminishes the training time altogether by diminishing the number of input features. To the best of our knowledge, the proposed amalgam- ation of HS-SA-SVM hybrid strategy with optimal feature subset is novel and treated as a new effort in the after-mentioned direction.

EXISTING SYSTEM :

This section presents the detailed description of the feature selection problem with the mathematical model and the definitions, concepts and the classifications of metaheuristic algorithms.

1. FEATURE SELECTION

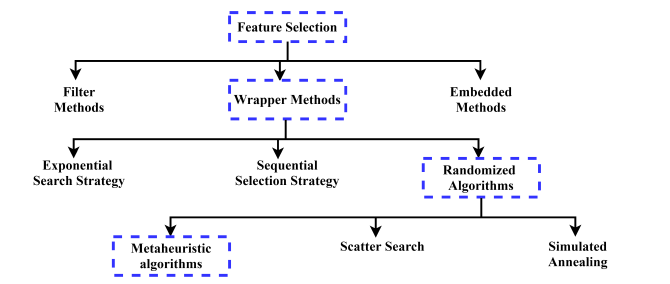
Feature selection deals with inappropriate, irrelevant, or unnecessary features. It is a process that extracts the best features from the datasets . Feature selection is one of the most critical and challenging problems in machine learning. The various applications of the feature selection problem can be demonstrated in different fields. There are some applications such as biomedical problems (to find the best gene from candidate gene) ; text mining (to find the best terms word or phrases) ; image analysis (to select the best visual contents pixels, colour) etc.

Mathematically, a feature selection problem can be formulated in the following way:

Assume a dataset 0S 0 contains 0d 0 number of features. Then the working mechanism of feature selection problem is to select relevant features among 0d 0 features. Given dataset S = {f1, f2, f3, . . . , fd } The objective is to select the best subsets of features from S. Extract Subset D = {f1, f2, f3, . . . , fn} where, n < d and f1, f2, f3, . . . , fn represents the features/attributes of any dataset. Figure 2 depicts the working mechanism of the feature selection process. From the figure, it can be observed that there are five main components of the feature selection process, i.e. original dataset, selection of feature subset, evaluation of feature subset, selection criterion and validation.

Several feature selection methods are developed to obtain the best subset of features. Generally, the techniques are classified into three categories filter, wrapper and embedded methods . Filter methods are independent of learning or classification algorithm. It always focuses on the general characteristics of the data . Wrapper methods always include the classification algorithm and interact with the classifier. These are computationally expensive methods than the filter and also provide more accurate results as compared to filter methods. Embedded methods are a combination of filters and wrapper methods. In embedded methods, the feature selection is a part of the training process and training process held with the classifier. Moreover, the embedded methods use learning algorithm in its process, they will be considered in wrapper approaches category.

Wrapper approaches present better results in comparison with filter methods, but they are slower than filters methods. Wrapper methods depend on the modelling algorithm in which every subset is generated and then evaluated. Subset generation in wrapper methods is based on the different search strategy. Jovic et al. differentiates search techniques into three categories; exponential, sequential and randomized selection strategy. In the exponential method, the number of evaluated features increases exponentially with the size of features. Although this method shows accurate results, it is not practically possible to apply because of the high computational cost. The examples for exponential search strategy are exhaustive search, branch and bound method . Sequential algorithms include or remove features sequentially. Once a feature is included or removed in the selected subset, it can not be further changed that leads to local optima. Some sequential algorithms are linear forward selection, floating forward or backward selection, best first etc. Randomized algorithms include randomness to explore the search space, which saves the algorithms from trapping into local optima. Randomized algorithms are commonly known as population-based approaches for example



simulated annealing, random generation, metaheuristic algorithms etc. . We do not present a detailed description of every method of the feature selection process. The detailed explanation of each method can be found in . The flow chart of categorization of methods for solving feature selection is shown in Figure . In the figure, the dashed line box represents the methodology of this paper which describes how we reach to metaheuristic algorithms.

1. METAHEURISTIC ALGORITHMS :

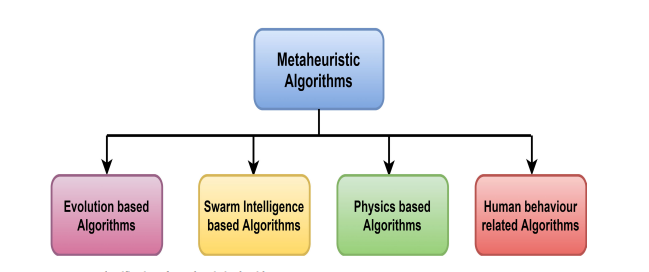
Metaheuristic algorithms are optimization methods that obtain the optimal (near-optimal) solution of optimization problems. These algorithms are derivative-free techniques and, have simplicity, flexibility and capability to avoid local optima . The behaviour of metaheuristic algorithms are stochastic; they start their optimization process by generating random solutions. It does not require to calculate the derivative of search space like in gradient search techniques. The metaheuristic algorithms are flexible and straightforward due to the simple concept and easy implementation. The algorithms can be modified easily according to the particular problem. The main property of metaheuristic algorithms is that they have a remarkable ability to prevent the algorithms from premature convergence. Due to the stochastic behaviour of algorithms, the techniques work as a black box and avoid local optima and explore the search space efficiently and effectively. The algorithms make a tradeoff between its two main essential aspects exploration and exploitation . In the exploration phase, the algorithms investigate the promising search space thoroughly, and exploitation comes for the local search of promising area(s) that are found in the exploration phase. They are successfully applied to various engineering and sciences problems, e.g. in electrical engineering (to find the optimal solution for power generation), industrial fields (scheduling jobs, transportation, vehicle routing problem, facility location problem), in civil engineering (to design the bridges, buildings), communication (radar design, networking), data mining (classification, prediction, clustering, system modelling) etc. Metaheuristic algorithms classify into the following two main categories;

1. Single solution based metaheuristic algorithms: These techniques start their optimization process with one solution, and their solution is updated during the iterations. It may lead to trapping into local optima and also does not explore the search space thoroughly.
2. Population (multiple) solution based metaheuristic algorithms: Initially, these algorithms generate a population of solutions and start their optimization process. The population of solutions update with the number of generations/iterations. The algorithms are beneficial for avoiding local optima as multiple solutions assist each other and have a great exploration of search space. They also have the quality of jump towards the promising part of search space. Therefore, population-based algorithms use in solving most of the real-world problems

Researchers pay great attention to metaheuristic algorithms because of their characteristics. Several algorithms have been designed and solved different types of problems. Based on their behaviour, the metaheuristic algorithms can be divided into four categories; evolution-based, swarm intelligencebased, physics-based and human-related algorithms . The categorization of the algorithms is depicted in Figure

1. **Evolution based algorithms:**

It is inspired from the natural evolution and start their process with randomly generated population of solutions. In these type of algorithms, the best solutions are put togther to create new individuals. The new individuals are formed using mutation, crossover and select the best solution. The most popular algorithm in this category is Genetic algorithm (GA) that is based on Darwin evolution technique . There are other algorithms such as evolution strategy genetic programming , tabu search , differential evolution etc.



**2.Swarm intelligence-based algorithms:**

These algorithms are inspired by the social behaviours of insects, animals, fishes or birds etc. The popular technique is Particle Swarm Optimization (PSO) developed by Kennedy and Eberhart [31]. It is inspired by the behaviour of a group of birds that fly throughout the search space and find their best location (position). Ant Colony optimization [32], Honey bee swarm optimization algorithm [33], monkey optimization [34] etc are the examples of swarm intelligence algorithms.

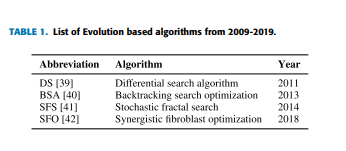
**3.Physics based algorithms:**

These are inspired by the rules of physics in the universe. Simulated annealing Harmony search etc come under physics-based algorithms.

**(4) Human behaviour related algorithms:**

These techniques are purely inspired by human behaviour. Every human being has its way of doing activities that affect its performance. It motivates researchers to develop the algorithms. The popular algorithms are Teaching learning-based optimization algorithm (TLBO) League Championship algorithm etc.

It is worth mentioning here that there are many metaheuristic algorithms developed from 1966 to till now. In this paper, we present the literature of those algorithms which are developed or proposed since 2009 to 2019 (ten-year span). According to the category, the list of metaheuristic algorithms are presented in Table 1, 2, 3, 4. The first column of the tables present the abbreviation; the second column gives the name of the algorithms, which is followed by the developed year. These algorithms have been applied to solve many real-world applications but, this paper is restricted to present the application in feature selection problems. Therefore, in the following sections, the algorithms which have been developed in ten years span and applied to feature selection problems are discussed.



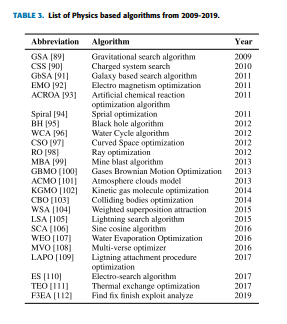
III. METAHEURISTIC ON FEATURE SELECTION :

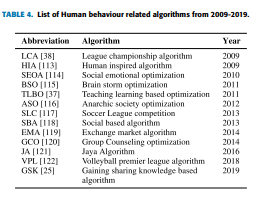
It describes the metaheuristic algorithm, which has been used in solving the feature selection problem. Binary vectors representations are considered to obtain the relevant feature. In the designed algorithm, a solution vector is represented by (10101100 . . . ..) this implies that 1 means that a particular feature is selected and 0 means that feature is not selected in the subset. Hence, this section investigates all binary variants of metaheuristic algorithms in detail. The first section describes the evolution-based algorithms; the second describes the swarm intelligence based algorithms, third demonstrates the physics-based algorithms, and the fourth one is for the human-related algorithm. And the last section is for the hybrid algorithms, which are a combination of two or more metaheuristic algorithms that have been used for classification problems.

1. **EVOLUTION BASED ALGORITHMS**

From Table 1, it can be seen that there are very few algorithms are developed in evolution based category from 2009- 2019. Gan and Duan proposed a chaotic differential search algorithm for image processing and it has been combined with lateral inhibition to edge extraction and image enhancement. Negahbani et al. used differential search algorithm for the diagnosis of coronary artery disease with fuzzy c-means that was used as a classifier. The performance of the proposed approach has been evaluated using accuracy, sensitivity and specificity measures. Zhang et al. proposed binary backtracking algorithm for wind speed forecasting in which extreme learning machine was employed for feature selection. Binary backtracking algorithm was developed using a sigmoidal function that transforms the continuous variables to binary variables. To identify the Leukemia cancer symptoms, Dhal et al. implemented the stochastic fractal search algorithm to provide optimal identification. The developed algorithm was compared with other classical methods and achieved high accuracy. Besides, a binary stochastic fractal search was developeto classify the galaxy colour images with extreme machine learning

. 





3 SYSTEM REQUIREMENTS**:**

## HARDWARE REQUIREMENTS:

* System : MINIMUM i3.
* Hard Disk : 40 GB.
* Ram : 4 GB.

## SOFTWARE REQUIREMENTS:

* **Operating System:** Windows 8
* **Coding Language**: Python 3.7

# SYSTEM STUDY

# FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

## ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

## TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

## SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

# 4.SYSTEM DESIGN

## 4.1 UML DIAGRAMS :

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

## GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

## USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



# CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

****

## SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

****

## ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



## IMPLEMENTATION:

## MODULES:

1. Admin:

admin can login to application by using username and password as admin and then can add new food details and store in Blockchain and can view orders from customers

1. User:

user can signup with the application and can browse food and book the food for delivery

# SOFTWARE ENVIRONMENT

## What is Python :

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

* + [Machine Learning](https://www.geeksforgeeks.org/machine-learning/)
  + GUI Applications (like Kivy, Tkinter, PyQt etc. )
  + Web frameworks like Django (used by YouTube, Instagram, Dropbox)
  + Image processing (like Opencv, Pillow)
  + Web scraping (like Scrapy, BeautifulSoup, Selenium)
  + Test frameworks
  + Multimedia

## Advantages of Python :-

Let’s see how Python dominates over other languages.

## 1. Extensive Libraries

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don’t have to write the complete code for that manually.

## 2. Extensible

As we have seen earlier, Python can be**extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

## 3. Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities**to our code in the other language.

## 4. Improved Productivity

The language’s simplicity and extensive libraries render programmers**more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

## 5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

When working with Java, you may have to create a class to print **‘Hello World’**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and**code.** This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

## 7. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory.** This further aids the readability of the code.

## 8. Object-Oriented

This language supports both the **procedural and object-oriented**programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

## 9. Free and Open-Source

Like we said earlier, Python is **freely available.** But not only can you[**download Python**](https://data-flair.training/blogs/install-python-windows/) for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

#### 10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn’t the same with Python. Here, you need to**code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system-dependent features.

## 11. Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages.

Any doubts till now in the advantages of Python? Mention in the comment section.

# Advantages of Python Over Other Languages :

## 1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don’t have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

## 2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

**The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.**

## 3. Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and [**machine learning**](https://data-flair.training/blogs/machine-learning-tutorials-home/), automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

## Disadvantages of Python

So far, we’ve seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let’s now see the downsides of choosing Python over another language.

#### 1. Speed Limitations

We have seen that Python code is executed line by line. But since [Python](https://www.python.org/) is interpreted, it often results in **slow execution**. This, however, isn’t a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

#### 2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the **client-side**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbonnelle**.

The reason it is not so famous despite the existence of Brython is that it isn’t that secure.

#### 3. Design Restrictions

As you know, Python is **dynamically-typed**. This means that you don’t need to declare the type of variable while writing the code. It uses **duck-typing**. But wait, what’s that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can**raise run-time errors**.

#### 4. Underdeveloped Database Access Layers

Compared to more widely used technologies like **JDBC (Java DataBase Connectivity)** and **ODBC (Open DataBase Connectivity)**, Python’s database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

#### 5. Simple

No, we’re not kidding. Python’s simplicity can indeed be a problem. Take my example. I don’t do Java, I’m more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

## History of Python : -

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python.Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners1, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

## What is Machine Learning : -

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models tunable parameters that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain.Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

## Categories Of Machine Leaning :-

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into classification tasks and regression tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

## Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven’t surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, “to make decisions, based on data, with efficiency and scale”.

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can’t do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

## Challenges in Machines Learning :-

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

**Quality of data** − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

**Time-Consuming task** − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

**Lack of specialist persons** − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

**No clear objective for formulating business problems** − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

**Issue of overfitting & underfitting** − If the model is overfitting or underfitting, it cannot be represented well for the problem.

**Curse of dimensionality** − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

**Difficulty in deployment** − Complexity of the ML model makes it quite difficult to be deployed in real life.

## Applications of Machines Learning :-

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML −

* Emotion analysis
* Sentiment analysis
* Error detection and prevention
* Weather forecasting and prediction
* Stock market analysis and forecasting
* Speech synthesis
* Speech recognition
* Customer segmentation
* Object recognition
* Fraud detection
* Fraud prevention
* Recommendation of products to customer in online shopping

# How to Start Learning Machine Learning?

Arthur Samuel coined the term **“Machine Learning”** in 1959 and defined it as a **“Field of study that gives computers the capability to learn without being explicitly programmed”.**

And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to [Indeed](http://blog.indeed.com/2019/03/14/best-jobs-2019/), Machine Learning Engineer Is The Best Job of 2019 with a 344% growth and an average base salary of **$146,085** per year.

But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it? So this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let’s get started!!!

### How to start learning ML?

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

### Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don’t know these, never fear! You don’t need a Ph.D. degree in these topics to get started but you do need a basic understanding.

#### (a) Learn Linear Algebra and Multivariate Calculus

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

#### (b) Learn Statistics

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise that you need to learn it!!!  
Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

#### (c) Learn Python

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is [Python](https://www.geeksforgeeks.org/python-programming-language/)! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as [Keras](https://keras.io/" \t "_blank), [TensorFlow](https://www.tensorflow.org/" \t "_blank), [Scikit-learn](https://scikit-learn.org/stable/" \t "_blank), etc.

So if you want to learn ML, it’s best if you learn Python! You can do that using various online resources and courses such as [**Fork Python**](https://practice.geeksforgeeks.org/courses/fork-python) available Free on GeeksforGeeks.

### Step 2 – Learn Various ML Concepts

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It’s best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

#### (a) Terminologies of Machine Learning

* **Model –**A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.
* **Feature –**A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.
* **Target (Label) –**A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.
* **Training –**The idea is to give a set of inputs(features) and it’s expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.
* **Prediction –**Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

#### (b) Types of Machine Learning

* **Supervised Learning –**This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.
* **Unsupervised Learning –**This involves using unlabelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.
* **Semi-supervised Learning –**This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.
* **Reinforcement Learning –**This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

### Advantages of Machine learning :-

#### 1. Easily identifies trends and patterns -

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

#### 2. No human intervention needed (automation)

With ML, you don’t need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

#### 3. Continuous Improvement

As [**ML algorithms**](https://data-flair.training/blogs/machine-learning-algorithms/) gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

#### 4. Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

#### 5. Wide Applications

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

### Disadvantages of Machine Learning :-

#### 1. Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

#### 2. Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

#### 3. Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

#### 4. High error-susceptibility

[**Machine Learning**](https://en.wikipedia.org/wiki/Machine_learning) is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

**Python Development Steps : -**

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system.  
Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked.Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting unicode.Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it."Some changes in Python 7.3:

* Print is now a function
* Views and iterators instead of lists
* The rules for ordering comparisons have been simplified. E.g. a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
* There is only one integer type left, i.e. int. long is int as well.
* The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behaviour.
* Text Vs. Data Instead Of Unicode Vs. 8-bit

**Purpose :-**

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

**Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

**Modules Used in Project :-**

**Tensorflow**

TensorFlow is a [free](https://en.wikipedia.org/wiki/Free_software) and [open-source](https://en.wikipedia.org/wiki/Open-source_software) [software library for dataflow and differentiable programming](https://en.wikipedia.org/wiki/Library_(computing)) across a range of tasks. It is a symbolic math library, and is also used for [machine learning](https://en.wikipedia.org/wiki/Machine_learning) applications such as [neural networks](https://en.wikipedia.org/wiki/Neural_networks). It is used for both research and production at [Google](https://en.wikipedia.org/wiki/Google).‍

TensorFlow was developed by the [Google Brain](https://en.wikipedia.org/wiki/Google_Brain) team for internal Google use. It was released under the [Apache 2.0](https://en.wikipedia.org/wiki/Apache_License) [open-source license](https://en.wikipedia.org/wiki/Open-source_license) on November 9, 2015.

**Numpy**

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

* A powerful N-dimensional array object
* Sophisticated (broadcasting) functions
* Tools for integrating C/C++ and Fortran code
* Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

**Pandas**

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

**Matplotlib**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and [IPython](http://ipython.org/) shells, the [Jupyter](http://jupyter.org/) Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the [sample plots](https://matplotlib.org/tutorials/introductory/sample_plots.html) and [thumbnail gallery](https://matplotlib.org/gallery/index.html).

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

**Scikit – learn**

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. **Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

**Install Python Step-by-Step in Windows and Mac :**

Python a versatile programming language doesn’t come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

## How to Install Python on Windows and Mac :

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

**Note:** The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your **System Requirements**. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a **Windows 64-bit operating system**. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. [Download the Python Cheatsheet here.](https://myelearninghub.com/python-cheat-sheet/)The steps on how to install Python on Windows 10, 8 and 7 are **divided into 4 parts** to help understand better.

### Download the Correct version into the system

**Step 1:** Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: [**https://www.python.org**](https://www.python.org/)



Now, check for the latest and the correct version for your operating system.

**Step 2:** Click on the Download Tab.

****

**Step 3:** You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

****

**Step 4:** Scroll down the page until you find the Files option.

**Step 5:** Here you see a different version of python along with the operating system.



• To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.

•To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

**Note:** To know the changes or updates that are made in the version you can click on the Release Note Option.

### Installation of Python

**Step 1:** Go to Download and Open the downloaded python version to carry out the installation process.



**Step 2:** Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.



**Step 3:** Click on Install NOW After the installation is successful. Click on Close.



With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

**Note:** The installation process might take a couple of minutes.

### Verify the Python Installation

**Step 1:** Click on Start

**Step 2:** In the Windows Run Command, type “cmd”.



**Step 3:** Open the Command prompt option.

**Step 4:** Let us test whether the python is correctly installed. Type **python –V** and press Enter.



**Step 5:** You will get the answer as 3.7.4

**Note:** If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

### Check how the Python IDLE works

**Step 1:** Click on Start

**Step 2:** In the Windows Run command, type “python idle”.



**Step 3:** Click on IDLE (Python 3.7 64-bit) and launch the program

**Step 4:** To go ahead with working in IDLE you must first save the file. **Click on File > Click on Save**



**Step 5:** Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

**Step 6:** Now for e.g. **enter print**

**6.SYSTEM TEST**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### TYPES OF TESTS

**Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**Unit Testing**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

# Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

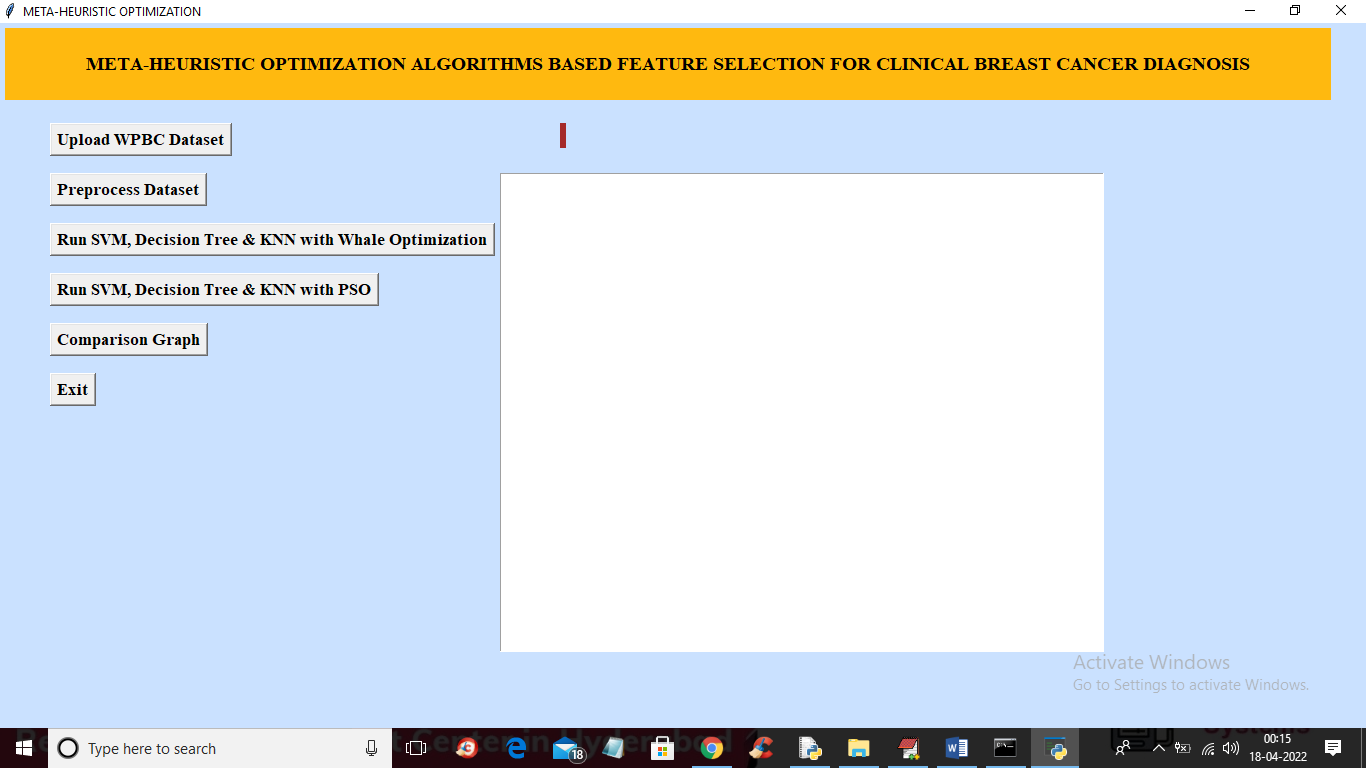
**Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

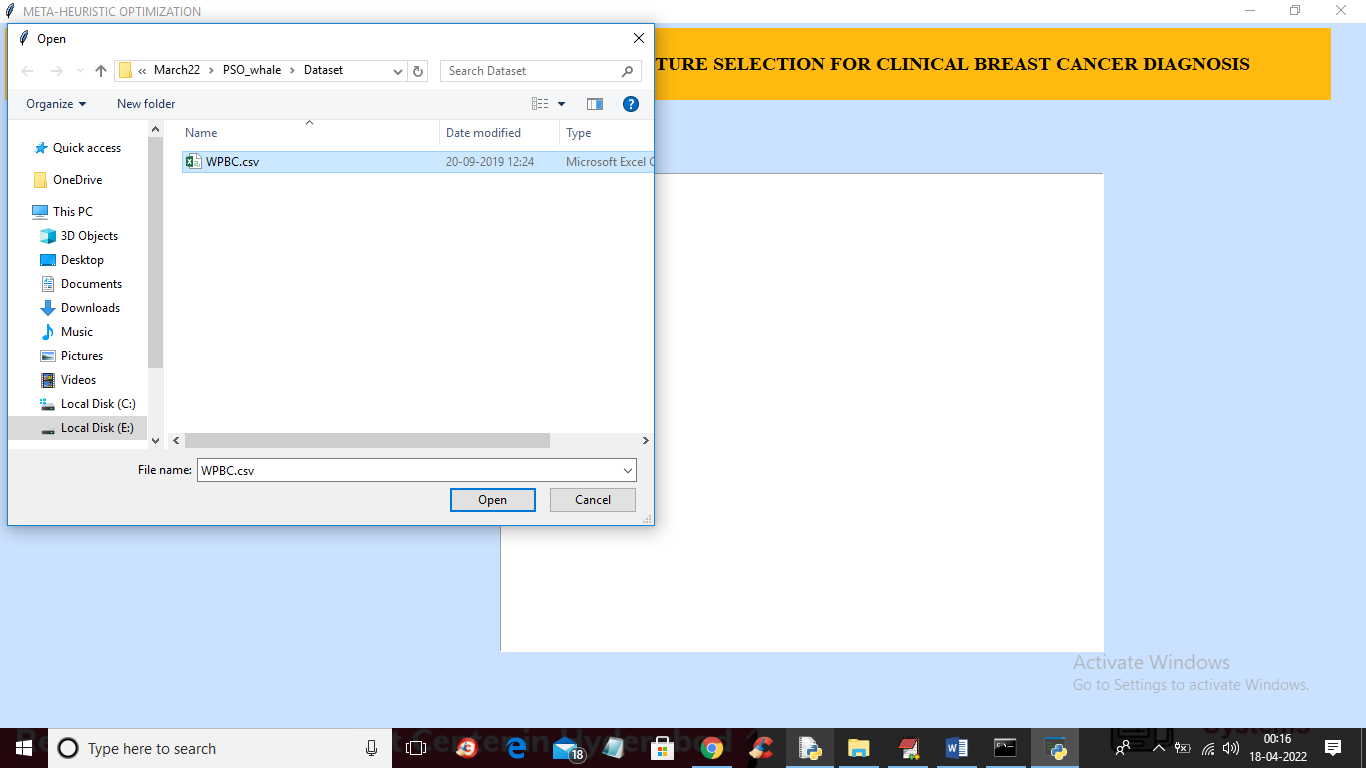
**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**7.SCREENSHOTS**

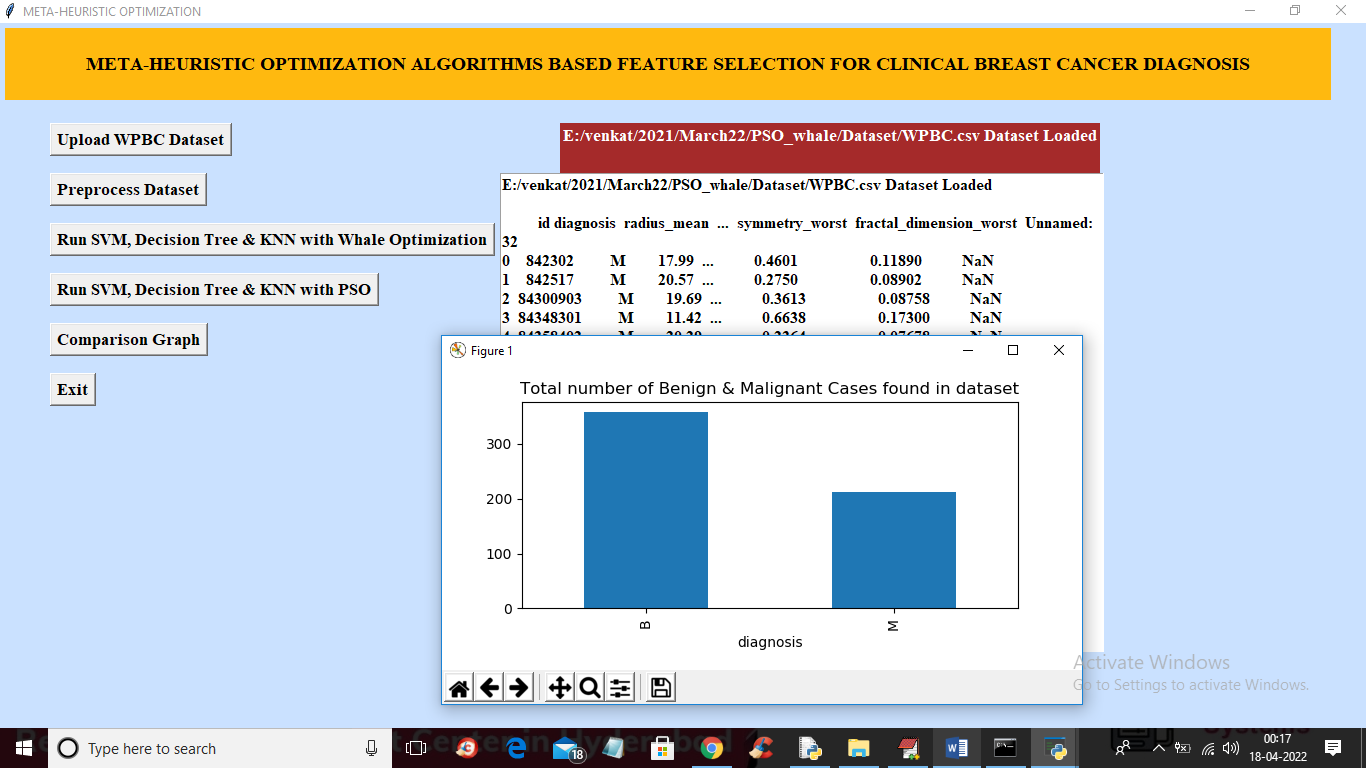
To run project double click on ‘run.bat’ file to get below screen



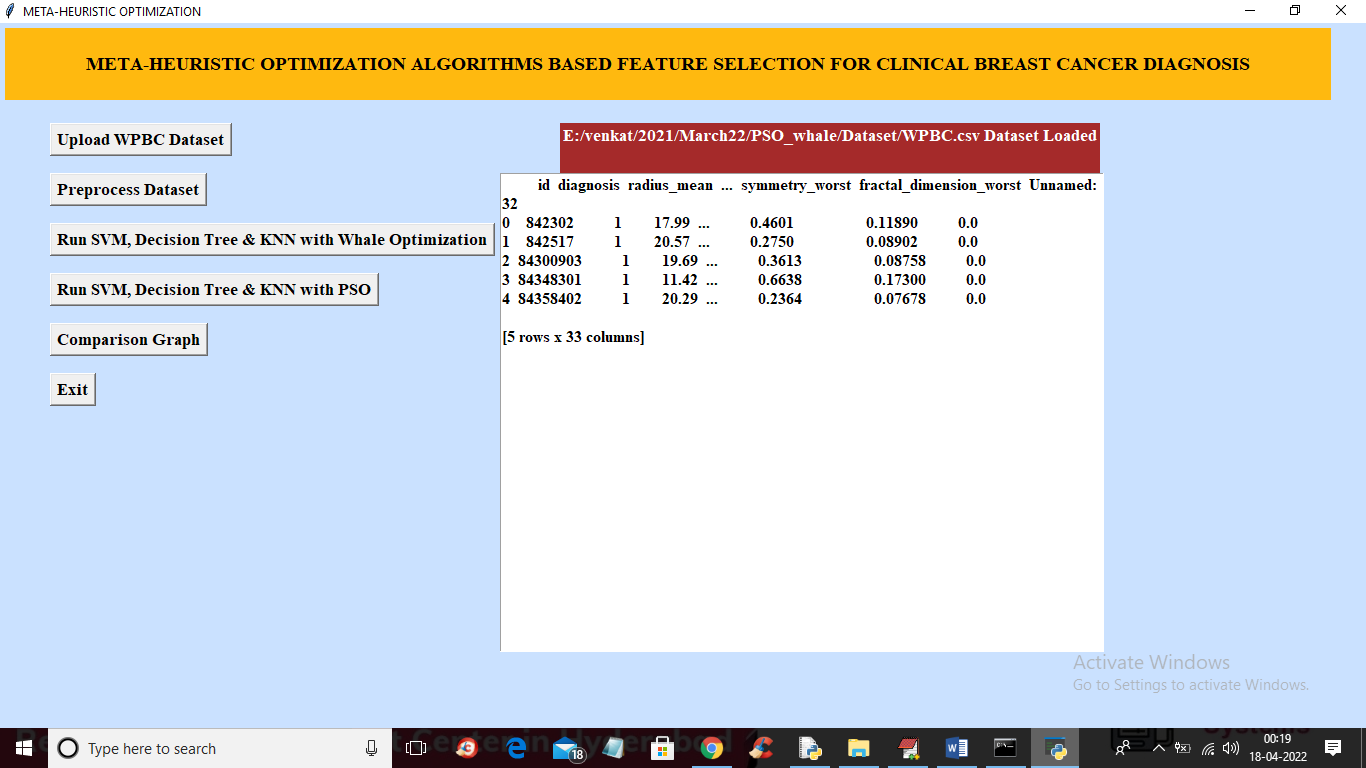
In above screen click on ‘Upload WPBC Dataset’ button to upload dataset and to get below screen



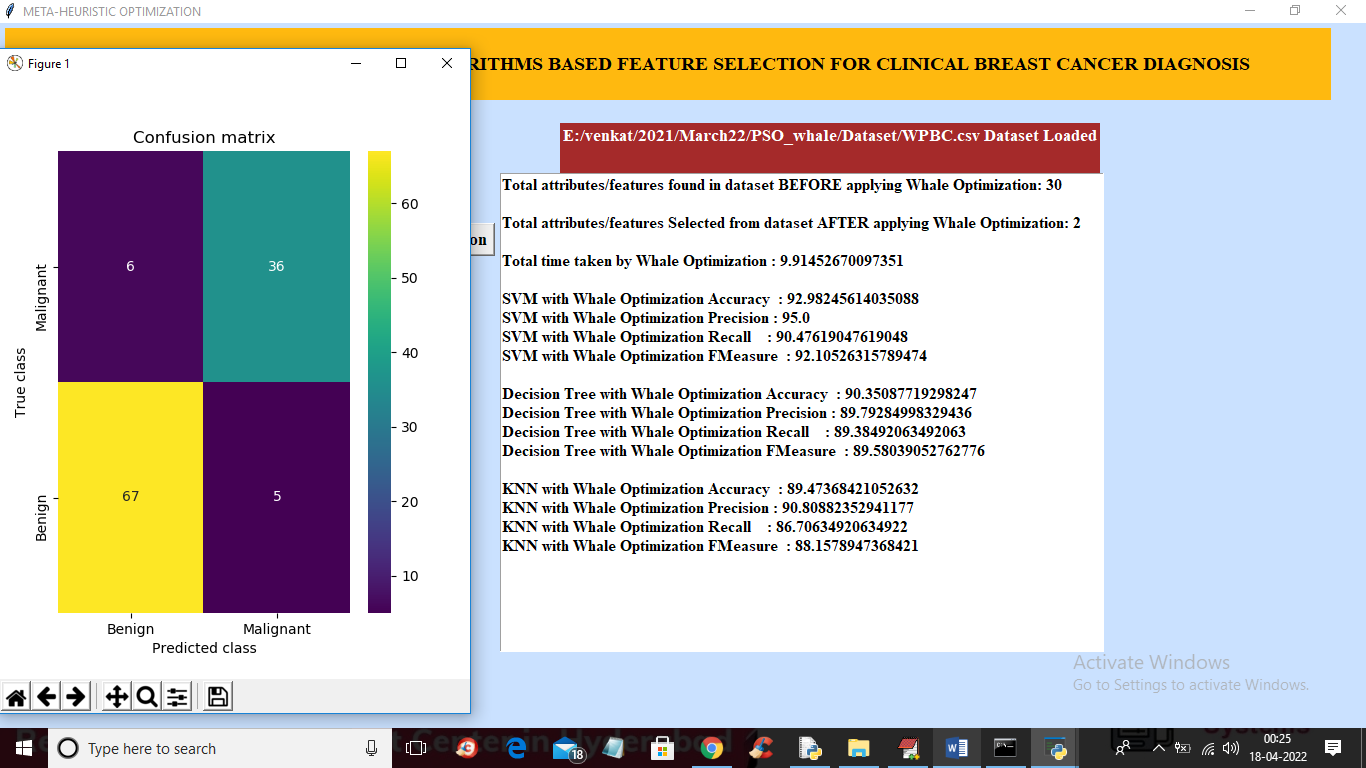
In above screen selecting and uploading dataset folder and then click on ‘Open’ button to load dataset and to get below screen



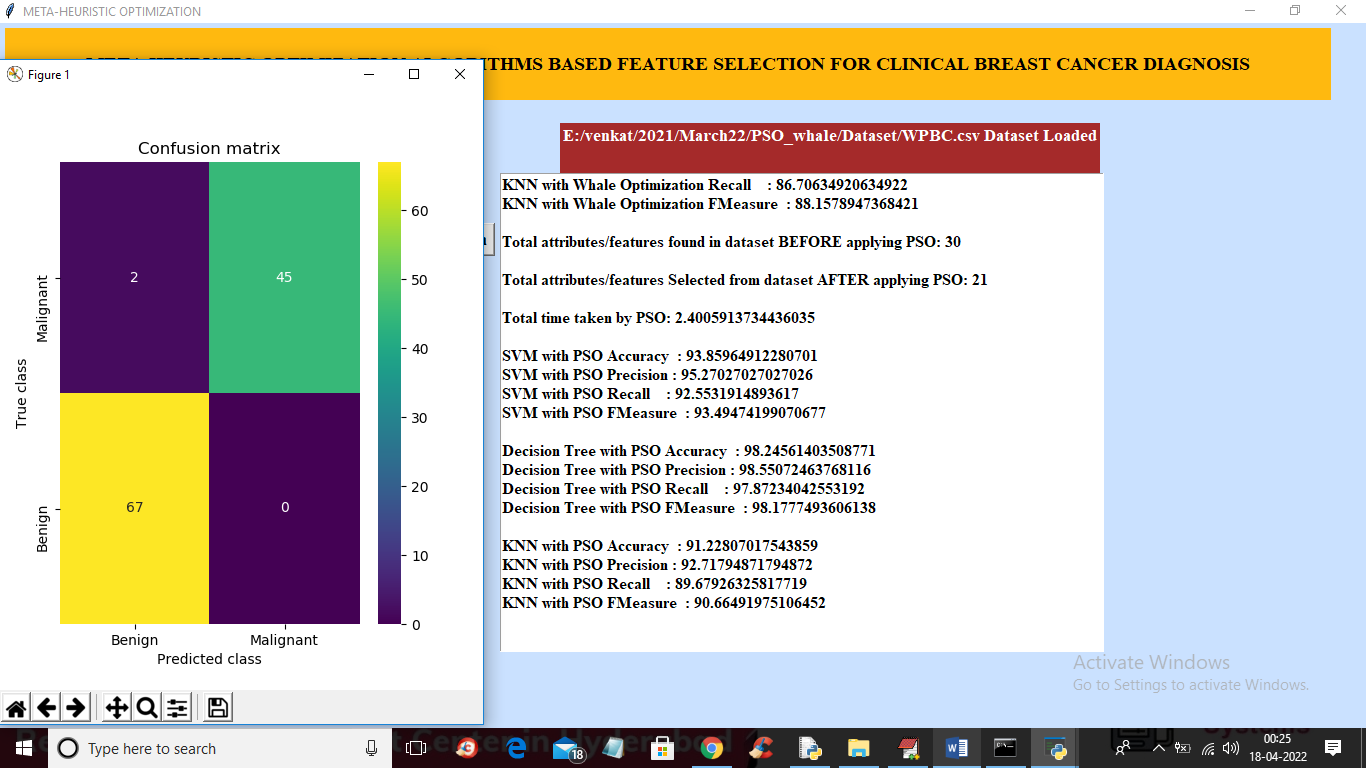
In above screen we can see dataset loaded and in dataset contains non-numeric values and missing NAN values and in above screen we can see graph showing number of benign and malignant cases found in dataset and now close above graph and then click on ‘Preprocess Dataset’ button to replace missing values and convert non-numeric data to numeric data and get below output



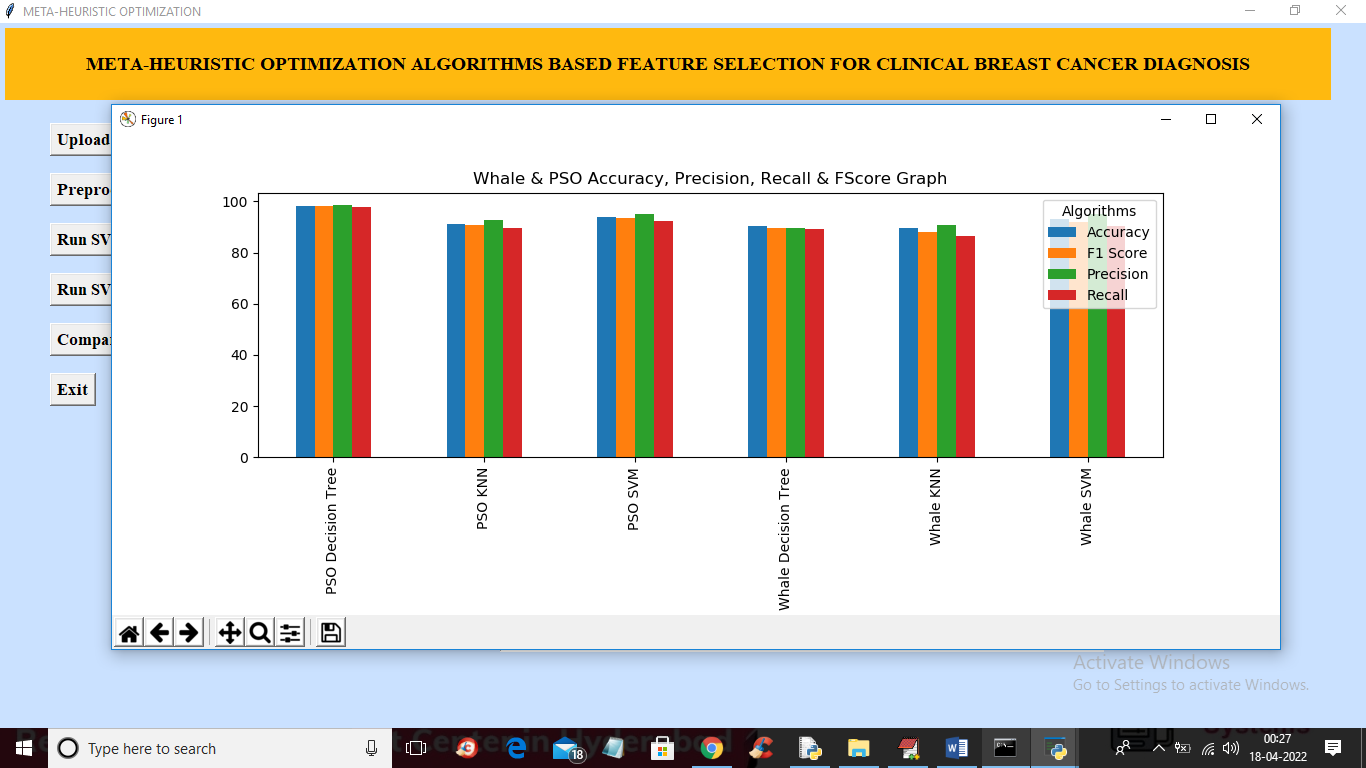
In above screen we can see all values are converted to numeric and now click on ‘Run SVM, Decision Tree & KNN with Whale Optimization’ button to apply whale optimization and train all ML algorithms to get below output



In above screen in first two lines we can see dataset contains 30 attributes and after applying whale we got 2 important attributes and then we can see accuracy of each algorithm on selected features and we can see execution time also and in above screen we can see prediction confusion matrix graph where application predict 67 records as benign correctly and only 6 records are incorrectly predicted and now close above graph and then click on ‘Run SVM, Decision Tree & KNN with PSO’ button to select features with PSO and train all algorithms to get below output

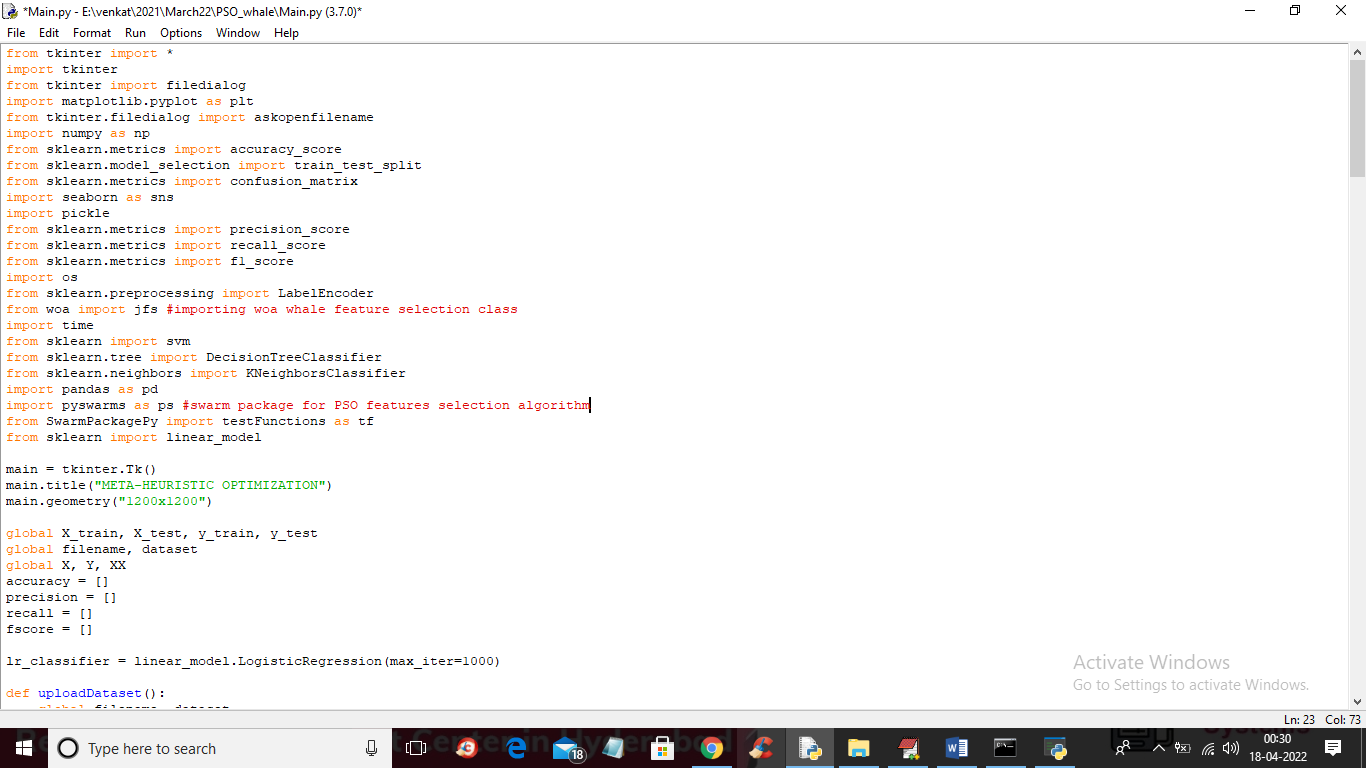


In above screen we can see PSO selected 21 attributes out of 30 and we can see accuracy of each algorithm on selected features and in confusion matrix we can see with PSO 67 records are correctly predicted as Benign and only 2 records are incorrectly predicted and now close above graph and then click on ‘Comparison Graph’ button to get below graph



In above graph x-axis represents algorithm names and y-axis represents accuracy and other values and in above graph each different colour bar represents different metrics like accuracy, precision and etc. In above graph we can see PSO with decision tree give better performance.

In below screen you can see we are using WOA package for whale optimization and PYSWARM package for PSO



In above screen read red colour comments to know about features selection classes

CONCLUSION :

This paper presents a comprehensive survey on metaheuristic algorithms that are developed from 2009 to the 2019 year and their binary variants, which have been applied to feature selection problem. A detailed description and mathematical model of feature selection problem are given that could help researchers to understand the problem properly. Moreover, the techniques of solving feature selection problems are presented. Additionally, metaheuristic algorithms are considered in solving the feature selection problem. Therefore, basic definition, importance and the classification of metaheuristic algorithms are given. The evolutionbased, swarm-based, physics-based category, human rlated algorithms have been developed and applied to feature selection problems.

However, metaheuristic algorithms have some following drawbacks:

• They suffer from slow convergence rate due to random generation movement.

• They explore the search space without knowing the search direction.

• They can trap into local optima, or they have some premature convergence.

• The values of the parameters used in the metaheuristic algorithms have to be adjusted, this may also lead to pre-mature convergence.

Besides, the limitation of the metaheuristic algorithms, the modified and enhanced version of the algorithms were developed which are successfully applied to the feature selection problems. Also, a categorization is presented based on the behaviour of algorithms; evolution-based, swarm-based, physics-related and human behaviour related algorithms. This paper benefits in such a way that a list of metaheuristic algorithms is presented based on their classification. It also benefits for the application point of view as it consists of a case study. The case study presents the eight benchmark datasets and the optimal feature subsets are found by implementing different metaheuristic algorithms.

evolution and human-related category, but there are several algorithms have been designed in the swarm and physics-related algorithms. It implies that there is a scope to develop or propose new metaheuristic algorithms in these categories. This paper mainly focuses on solving the feature selection problem using binary variants of metaheuristic algorithms. Hence, extensive literature is presented in every class of metaheuristic algorithms. All binary variants of all reviewed algorithms regarding feature selection problems are pointed. In swarm-based category, all binary variants of Cuckoo search, Bat algorithm, Firefly algorithm, flower pollination algorithm, Krill herd algorithm, Grey wolf optimizer, Ant lion optimizer, Dragonfly algorithm, Whale optimization algorithm, Grasshopper optimization algorithm, Salp swarm algorithm are reviewed with the key factor of solving feature selection problem. Moreover, hybrid approaches are also reviewed in the process of solving the feature selection problem.

It can be concluded that there is some area(s) which are less explored, such as spam detection, theft detection and weather prediction. However, lots of research has been done on the well-known datasets of UCI repository and in medical diagnosis (cancer classification), intrusion detection systems, text classification, multimedia etc. Hence, researchers should pay great attention to explore this area with metaheuristic algorithms. Moreover, there are some algorithms in the literature for which binary variants are not developed yet such as PFA, CGS, TCO, ES, HSO, WSA, BMO, OptBees, TGSR, EVOA, VCS, EPC, GbSA, CSO, WEO, LCA, EMA, VPL. These algorithms benefit classification after developing their binary version. From the literature, it can be observed that the researcher has to face many challenges to obtain the best feature subset of the considered classification problem. A good choice of classifier has a significant impact of the quality of obtained solution such KNN classifier is the most used classifier in getting the best subset with well-known datasets of UCI repository. After that, SVM classifier used to classify in different applications such as medical diagnosis, pattern recognition, image analysis etc. There are some other classifiers which are less used in terms of classification. Hence, this another gap to use different classifiers in classification problem and compared with most used ones. Finally, researchers will get the benefit of this study as they could find all the key factors in solving the feature selection problem using metaheuristic algorithms under one roof.

REFERENCES :

[1] B. Xue, M. Zhang, W. N. Browne, and X. Yao, ‘‘A survey on evolutionary computation approaches to feature selection,’’ IEEE Trans. Evol. Comput., vol. 20, no. 4, pp. 606–626, Aug. 2016.

[2] P. Y. Lee, W. P. Loh, and J. F. Chin, ‘‘Feature selection in multimedia: The state-of-the-art review,’’ Image Vis. Comput., vol. 67, pp. 29–42, Nov. 2017.

[3] B. Remeseiro and V. Bolon-Canedo, ‘‘A review of feature selection methods in medical applications,’’ Comput. Biol. Med., vol. 112, Sep. 2019, Art. no. 103375.

[4] M. Sharma and P. Kaur, ‘‘A comprehensive analysis of nature-inspired meta-heuristic techniques for feature selection problem,’’ Arch. Comput. Methods Eng., pp. 1–25, Feb. 2020, doi: 10.1007/s11831-020-09412-6.

[5] M. Z. Asghar, A. Khan, S. Ahmad, and F. M. Kundi, ‘‘A review of feature extraction in sentiment analysis,’’ J. Basic Appl. Sci. Res., vol. 4, no. 3, pp. 181–186, 2014.

[6] Y. Saeys, I. Inza, and P. Larrañaga, ‘‘A review of feature selection techniques in bioinformatics,’’ Bioinformatics, vol. 23, no. 19, pp. 2507–2517, Oct. 2007.

[7] V. Bolón-Canedo and A. Alonso-Betanzos, ‘‘Ensembles for feature selection: A review and future trends,’’ Inf. Fusion, vol. 52, pp. 1–12, Dec. 2019.

[8] H. Liu and L. Yu, ‘‘Toward integrating feature selection algorithms for classification and clustering,’’ IEEE Trans. Knowl. Data Eng., vol. 17, no. 4, pp. 491–502, Apr. 2005.

[9] S. Ahmed, M. Zhang, and L. Peng, ‘‘Enhanced feature selection for biomarker discovery in LC-MS data using GP,’’ in Proc. IEEE Congr. Evol. Comput., Jun. 2013, pp. 584–591.

[10] M. H. Aghdam, N. Ghasem-Aghaee, and M. E. Basiri, ‘‘Text feature selection using ant colony optimization,’’ Expert Syst. Appl., vol. 36, no. 3, pp. 6843–6853, Apr. 2009.

[11] A. Ghosh, A. Datta, and S. Ghosh, ‘‘Self-adaptive differential evolution for feature selection in hyperspectral image data,’’ Appl. Soft Comput., vol. 13, no. 4, pp. 1969–1977, Apr. 2013.

[12] M. Dash and H. Liu, ‘‘Feature selection for classification,’’ Intell. Data Anal., vol. 1, nos. 1–4, pp. 131–156, 1997.

[13] I. Guyon and A. Elisseeff, ‘‘An introduction to variable and feature selection,’’ J. Mach. Learn. Res., vol. 3, pp. 1157–1182, Mar. 2003.

[14] H. Liu, H. Motoda, R. Setiono, and Z. Zhao, ‘‘Feature selection: An ever evolving frontier in data mining,’’ in Feature Selection in Data Mining. Hyderabad, India, 2010, pp. 4–13.

[15] N. Hoque, D. K. Bhattacharyya, and J. K. Kalita, ‘‘MIFS-ND: A mutual information-based feature selection method,’’ Expert Syst. Appl., vol. 41, no. 14, pp. 6371–6385, Oct. 2014.

[16] Z. Xu, I. King, M. R.-T. Lyu, and R. Jin, ‘‘Discriminative semi-supervised feature selection via manifold regularization,’’ IEEE Trans. Neural Netw., vol. 21, no. 7, pp. 1033–1047, Jul. 2010.

[17] J. Tang, S. Alelyani, and H. Liu, ‘‘Feature selection for classification: A review,’’ in Data Classification: Algorithms and Applications. 2014, p. 37.

[18] A. Jović, K. Brkić, and N. Bogunović, ‘‘A review of feature selection methods with applications,’’ in Proc. 38th Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO), 2015, pp. 1200–1205.

[19] Z. Sun, G. Bebis, and R. Miller, ‘‘Object detection using feature subset selection,’’ Pattern Recognit., vol. 37, no. 11, pp. 2165–2176, Nov. 2004.

[20] A. K. Jain, R. P. W. Duin, and J. Mao, ‘‘Statistical pattern recognition: A review,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 1, pp. 4–37, Jan. 2000.

[21] H. Liu and H. Motoda, Feature Extraction, Construction Selection: A Data Mining Perspective, vol. 453. USA: Springer, 1998.

[22] S. Mirjalili, S. M. Mirjalili, and A. Lewis, ‘‘Grey wolf optimizer,’’ Adv. Eng. Softw., vol. 69, pp. 46–61, Mar. 2014.

[23] O. Olorunda and A. P. Engelbrecht, ‘‘Measuring exploration/exploitation in particle swarms using swarm diversity,’’ in Proc. IEEE Congr. Evol. Comput. (IEEE World Congr. Comput. Intell.), Jun. 2008, pp. 1128–1134.

[24] L. Lin and M. Gen, ‘‘Auto-tuning strategy for evolutionary algorithms: Balancing between exploration and exploitation,’’ Soft Comput., vol. 13, no. 2, pp. 157–168, Jan. 2009.

[25] A. W. Mohamed, A. A. Hadi, and A. K. Mohamed, ‘‘Gaining-sharing knowledge based algorithm for solving optimization problems: A novel nature-inspired algorithm,’’ Int. J. Mach. Learn. Cybern., vol. 11, pp. 1501–1529, 2020.

[26] J. H. Holland, ‘‘Genetic algorithms,’’ Sci. Amer., vol. 267, no. 1, pp. 66–73, 1992.

[27] I. Rechenberg, ‘‘Evolutionsstrategien,’’ in Simulationsmethoden der Medizin und Biologie. Berlin, Germany: Springer, 1978, pp. 83–114.

[28] J. H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence. Cambridge, MA, USA: MIT Press, 1992.

[29] F. Glover, ‘‘Future paths for integer programming and links to artificial intelligence,’’ Comput. Oper. Res., vol. 13, no. 5, pp. 533–549, Jan. 1986.

[30] R. Storn and K. Price, ‘‘Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces,’’ J. Global Optim., vol. 11, no. 4, pp. 341–359, 1997.

[31] J. Kennedy and R. Eberhart, ‘‘Particle swarm optimization,’’ in Proc. IEEE Int. Conf. Neural Netw. (ICNN), vol. 4, Nov./Dec. 1995, pp. 1942–1948.

[32] M. Dorigo, V. Maniezzo, and A. Colorni, ‘‘Ant system: Optimization by a colony of cooperating agents,’’ IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 26, no. 1, pp. 29–41, Feb. 1996.

[33] D. Karaboga, ‘‘An idea based on honey bee swarm for numerical optimization,’’ Dept. Comput. Eng., Erciyes Univ., Kayseri, Turkey, Tech. Rep. 06, 2005.

[34] R.-Q. Zhao and W.-S. Tang, ‘‘Monkey algorithm for global numerical optimization,’’ J. Uncertain Syst., vol. 2, no. 3, pp. 165–176, 2008.

[35] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, ‘‘Optimization by simulated annealing,’’ Science, vol. 220, no. 4598, pp. 671–680, 1983.

[36] Z. Woo Geem, J. Hoon Kim, and G. V. Loganathan, ‘‘A new heuristic optimization algorithm: Harmony search,’’ Simulation, vol. 76, no. 2, pp. 60–68, Feb. 2001.

[37] R. V. Rao, V. J. Savsani, and D. P. Vakharia, ‘‘Teaching–learning-based optimization: An optimization method for continuous non-linear large scale problems,’’ Inf. Sci., vol. 183, no. 1, pp. 1–15, Jan. 2012.

[38] A. H. Kashan, ‘‘League championship algorithm: A new algorithm for numerical function optimization,’’ in Proc. Int. Conf. Soft Comput. Pattern Recognit., 2009, pp. 43–48.

[39] P. Civicioglu, ‘‘Transforming geocentric Cartesian coordinates to geodetic coordinates by using differential search algorithm,’’ Comput. Geosci., vol. 46, pp. 229–247, Sep. 2012.

[40] P. Civicioglu, ‘‘Backtracking search optimization algorithm for numerical optimization problems,’’ Appl. Math. Comput., vol. 219, no. 15, pp. 8121–8144, Apr. 2013.

[41] H. Salimi, ‘‘Stochastic fractal search: A powerful metaheuristic algorithm,’’ Knowl.-Based Syst., vol. 75, pp. 1–18, Feb. 2015.

[42] T. T. Dhivyaprabha, P. Subashini, and M. Krishnaveni, ‘‘Synergistic fibroblast optimization: A novel nature-inspired computing algorithm,’’ Frontiers Inf. Technol. Electron. Eng., vol. 19, no. 7, pp. 815–833, Jul. 2018.

[43] F. Comellas and J. Martinez-Navarro, ‘‘Bumblebees: A multiagent combinatorial optimization algorithm inspired by social insect behaviour,’’ in Proc. 1st ACM/SIGEVO Summit Genet. Evol. Comput., 2009, pp. 811–814.

[44] U. Premaratne, J. Samarabandu, and T. Sidhu, ‘‘A new biologically inspired optimization algorithm,’’ in Proc. Int. Conf. Ind. Inf. Syst. (ICIIS), Dec. 2009, pp. 279–284.

[45] X.-S. Yang and S. Deb, ‘‘Cuckoo search via Lévy flights,’’ in Proc. World Congr. Nature Biol. Inspired Comput. (NaBIC), 2009, pp. 210–214.

[46] S. He, Q. H. Wu, and J. R. Saunders, ‘‘Group search optimizer: An optimization algorithm inspired by animal searching behavior,’’ IEEE Trans. Evol. Comput., vol. 13, no. 5, pp. 973–990, Oct. 2009.

[47] S. Iordache, ‘‘Consultant-guided search: A new Metaheuristic for combinatorial optimization problems,’’ in Proc. 12th Annu. Conf. Genetic Evol. Comput., 2010, pp. 225–232.

[48] X.-S. Yang, ‘‘A new metaheuristic bat-inspired algorithm,’’ in Nature Inspired Cooperative Strategies for Optimization. Berlin, Germany: Springer, 2010, pp. 65–74.

[49] R. Hedayatzadeh, F. Akhavan Salmassi, M. Keshtgari, R. Akbari, and K. Ziarati, ‘‘Termite colony optimization: A novel approach for optimizing continuous problems,’’ in Proc. 18th Iranian Conf. Electr. Eng., May 2010, pp. 553–558.

[50] R. Oftadeh, M. J. Mahjoob, and M. Shariatpanahi, ‘‘A novel metaheuristic optimization algorithm inspired by group hunting of animals: Hunting search,’’ Comput. Math. Appl., vol. 60, no. 7, pp. 2087–2098, Oct. 2010.

[51] X.-S. Yang and S. Deb, ‘‘Eagle strategy using lévy walk and firefly algorithms for stochastic optimization,’’ in Nature Inspired Cooperative Strategies for Optimization. Berlin, Germany: Springer, 2010, pp. 101–111.

[52] H. Chen, Y. Zhu, K. Hu, and X. He, ‘‘Hierarchical swarm model: A new approach to optimization,’’ Discrete Dyn. Nature Soc., vol. 2010, pp. 1–30, Jan. 2010.

[53] X-S. Yang, ‘‘Firefly algorithm, stochastic test functions and design optimisation,’’ Int. J. Bio-Inspired Comput., vol. 2, no. 2, pp. 78–84, 2010.

[54] W.-T. Pan, ‘‘A new fruit fly optimization algorithm: Taking the financial distress model as an example,’’ Knowl.-Based Syst., vol. 26, pp. 69–74, Feb. 2012.

[55] R. S. Parpinelli and H. S. Lopes, ‘‘An eco-inspired evolutionary algorithm applied to numerical optimization,’’ in Proc. 3rd World Congr. Nature Biol. Inspired Comput., Oct. 2011, pp. 466–471.

[56] T. Ting, K. L. Man, S.-U. Guan, M. Nayel, and K. Wan, ‘‘Weightless swarm algorithm (WSA) for dynamic optimization problems,’’ in Proc. IFIP Int. Conf. Netw. Parallel Comput. Berlin, Germany: Springer, 2012, pp. 508–515.

[57] X.-S. Yang, ‘‘Flower pollination algorithm for global optimization,’’ in Proc. Int. Conf. Unconv. Comput. Natural Comput. Berlin, Germany: Springer, 2012, pp. 240–249.

[58] A. Askarzadeh and A. Rezazadeh, ‘‘A new heuristic optimization algorithm for modeling of proton exchange membrane fuel cell: Bird mating optimizer,’’ Int. J. Energy Res., vol. 37, no. 10, pp. 1196–1204, Aug. 2013.

[59] P. Civicioglu, ‘‘Artificial cooperative search algorithm for numerical optimization problems,’’ Inf. Sci., vol. 229, pp. 58–76, Apr. 2013.

[60] A. H. Gandomi and A. H. Alavi, ‘‘Krill herd: A new bio-inspired optimization algorithm,’’ Commun. Nonlinear Sci. Numer. Simul., vol. 17, no. 12, pp. 4831–4845, Dec. 2012.

[61] H. Hernández and C. Blum, ‘‘Distributed graph coloring: An approach based on the calling behavior of Japanese tree frogs,’’ Swarm Intell., vol. 6, no. 2, pp. 117–150, Jun. 2012.

[62] R. D. Maia, L. N. de Castro, and W. M. Caminhas, ‘‘Bee colonies as model for multimodal continuous optimization: The OptBees algorithm,’’ in Proc. IEEE Congr. Evol. Comput., Jun. 2012, pp. 1–8.

[63] R. Tang, S. Fong, X.-S. Yang, and S. Deb, ‘‘Wolf search algorithm with ephemeral memory,’’ in Proc. 7th Int. Conf. Digit. Inf. Manage. (ICDIM), Aug. 2012, pp. 165–172.

[64] A. Mozaffari, A. Fathi, and S. Behzadipour, ‘‘The great salmon run: A novel bio-inspired algorithm for artificial system design and optimisation,’’ Int. J. Bio-Inspired Comput., vol. 4, no. 5, pp. 286–301, 2012.

[65] A. Kaveh and N. Farhoudi, ‘‘A new optimization method: Dolphin echolocation,’’ Adv. Eng. Softw., vol. 59, pp. 53–70, May 2013.

[66] M. Neshat, G. Sepidnam, and M. Sargolzaei, ‘‘Swallow swarm optimization algorithm: A new method to optimization,’’ Neural Comput. Appl., vol. 23, no. 2, pp. 429–454, Aug. 2013.

[67] C. Sur, S. Sharma, and A. Shukla, ‘‘Egyptian vulture optimization algorithm—A new nature inspired meta-heuristics for knapsack problem,’’ in Proc. 9th Int. Conf. Comput. Inf. Technol. (IC2IT). Berlin, Germany: Springer, 2013, pp. 227–237.

[68] X. Meng, Y. Liu, X. Gao, and H. Zhang, ‘‘A new bio-inspired algorithm: Chicken swarm optimization,’’ in Proc. Int. Conf. Swarm Intell. Cham, Switzerland: Springer, 2014, pp. 86–94.

[69] X. Li, J. Zhang, and M. Yin, ‘‘Animal migration optimization: An optimization algorithm inspired by animal migration behavior,’’ Neural Comput. Appl., vol. 24, nos. 7–8, pp. 1867–1877, Jun. 2014.

[70] O. Abedinia, N. Amjady, and A. Ghasemi, ‘‘A new metaheuristic algorithm based on shark smell optimization,’’ Complexity, vol. 21, no. 5, pp. 97–116, May 2016.

[71] S. Mirjalili, ‘‘The ant lion optimizer,’’ Adv. Eng. Softw., vol. 83, pp. 80–98, May 2015.

[72] X.-B. Meng, X. Z. Gao, L. Lu, Y. Liu, and H. Zhang, ‘‘A new bio-inspired optimisation algorithm: Bird swarm algorithm,’’ J. Experim. Theor. Artif. Intell., vol. 28, no. 4, pp. 673–687, Jul. 2016.

[73] M. D. Li, H. Zhao, X. W. Weng, and T. Han, ‘‘A novel nature-inspired algorithm for optimization: Virus colony search,’’ Adv. Eng. Softw., vol. 92, pp. 65–88, Feb. 2016.

[74] S. A. Uymaz, G. Tezel, and E. Yel, ‘‘Artificial algae algorithm (AAA) for nonlinear global optimization,’’ Appl. Soft Comput., vol. 31, pp. 153–171, Jun. 2015.

[75] S. Mirjalili, ‘‘Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems,’’ Neural Comput. Appl., vol. 27, no. 4, pp. 1053–1073, May 2016.

[76] W. Yong, W. Tao, Z. Cheng-Zhi, and H. Hua-Juan, ‘‘A new stochastic optimization approach—Dolphin swarm optimization algorithm,’’ Int. J. Comput. Intell. Appl., vol. 15, no. 2, 2016, Art. no. 1650011.

[77] A. Askarzadeh, ‘‘A novel Metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm,’’ Comput. Struct., vol. 169, pp. 1–12, Jun. 2016

[78] S. Mirjalili and A. Lewis, ‘‘The whale optimization algorithm,’’ Adv. Eng. Softw., vol. 95, pp. 51–67, May 2016.

[79] E. Jahani and M. Chizari, ‘‘Tackling global optimization problems with a novel algorithm–mouth brooding fish algorithm,’’ Appl. Soft Comput., vol. 62, pp. 987–1002, Jan. 2018.

[80] X. Qi, Y. Zhu, and H. Zhang, ‘‘A new meta-heuristic butterfly-inspired algorithm,’’ J. Comput. Sci., vol. 23, pp. 226–239, Nov. 2017.

[81] F. Fausto, E. Cuevas, A. Valdivia, and A. González, ‘‘A global optimization algorithm inspired in the behavior of selfish herds,’’ Biosystems, vol. 160, pp. 39–55, Oct. 2017.

[82] S. Saremi, S. Mirjalili, and A. Lewis, ‘‘Grasshopper optimisation algorithm: Theory and application,’’ Adv. Eng. Softw., vol. 105, pp. 30–47, Mar. 2017

[83] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, ‘‘Salp swarm algorithm: A bio-inspired optimizer for engineering design problems,’’ Adv. Eng. Softw., vol. 114, pp. 163–191, Dec. 2017.

[84] G. Dhiman and V. Kumar, ‘‘Spotted hyena optimizer: A novel bioinspired based Metaheuristic technique for engineering applications,’’ Adv. Eng. Softw., vol. 114, pp. 48–70, Dec. 2017.

[85] G. Dhiman and V. Kumar, ‘‘Emperor penguin optimizer: A bio-inspired algorithm for engineering problems,’’ Knowl.-Based Syst., vol. 159, pp. 20–50, Nov. 2018.

[86] M. Jain, V. Singh, and A. Rani, ‘‘A novel nature-inspired algorithm for optimization: Squirrel search algorithm,’’ Swarm Evol. Comput., vol. 44, pp. 148–175, Feb. 2019.

[87] S. Arora and S. Singh, ‘‘Butterfly optimization algorithm: A novel approach for global optimization,’’ Soft Comput., vol. 23, no. 3, pp. 715–734, Feb. 2019.

[88] S. Harifi, M. Khalilian, J. Mohammadzadeh, and S. Ebrahimnejad, ‘‘Emperor penguins colony: A new Metaheuristic algorithm for optimization,’’ Evol. Intell., vol. 12, no. 2, pp. 211–226, Jun. 2019.

[89] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, ‘‘GSA: A gravitational search algorithm,’’ Inf. Sci., vol. 179, no. 13, pp. 2232–2248, Jun. 2009.

[90] A. Kaveh and S. Talatahari, ‘‘A novel heuristic optimization method: Charged system search,’’ Acta Mechanica, vol. 213, nos. 3–4, pp. 267–289, Sep. 2010.

[91] H. Shah-Osseini, ‘‘Principal components analysis by the galaxy-based search algorithm: A novel metaheuristic for continuous optimisation,’’ Int. J. Comput. Sci. Eng., vol. 6, nos. 1–2, pp. 132–140, 2011.

[92] E. Cuevas, D. Oliva, D. Zaldivar, M. Pérez-Cisneros, and H. Sossa, ‘‘Circle detection using electro-magnetism optimization,’’ Inf. Sci., vol. 182, no. 1, pp. 40–55, Jan. 2012.

[93] B. Alatas, ‘‘ACROA: Artificial chemical reaction optimization algorithm for global optimization,’’ Expert Syst. Appl., vol. 38, no. 10, pp. 13170–13180, Sep. 2011.

[94] K. Tamura and K. Yasuda, ‘‘Spiral optimization—A new multipoint search method,’’ in Proc. IEEE Int. Conf. Syst., Man, Cybern., Oct. 2011, pp. 1759–1764.

[95] A. Hatamlou, ‘‘Black hole: A new heuristic optimization approach for data clustering,’’ Inf. Sci., vol. 222, pp. 175–184, Feb. 2013.

[96] H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi, ‘‘Water cycle algorithm—A novel Metaheuristic optimization method for solving constrained engineering optimization problems,’’ Comput. Struct., vols. 110–111, pp. 151–166, Nov. 2012.

[97] F. Farrahi Moghaddam, R. Farrahi Moghaddam, and M. Cheriet, ‘‘Curved space optimization: A random search based on general relativity theory,’’ 2012, arXiv:1208.2214. [Online]. Available: [http://arxiv.org/abs/1208. 2214](http://arxiv.org/abs/1208.%202214)

[98] A. Kaveh and M. Khayatazad, ‘‘A new meta-heuristic method: Ray optimization,’’ Comput. Struct., vols. 112–113, pp. 283–294, Dec. 2012.

[99] A. Sadollah, A. Bahreininejad, H. Eskandar, and M. Hamdi, ‘‘Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems,’’ Appl. Soft Comput., vol. 13, no. 5, pp. 2592–2612, May 2013.

[100] M. Abdechiri, M. R. Meybodi, and H. Bahrami, ‘‘Gases Brownian motion optimization: An algorithm for optimization (GBMO),’’ Appl. Soft Comput., vol. 13, no. 5, pp. 2932–2946, May 2013.