A Proposed Model for an Artificial Intelligence Algorithm to Improve Pharmaceutical Industry

Hossam Abdelrahman Al-Ansary[1]

[1]Information Technology and Computer Sciences -Cairo University, Égypt

Publication Date: 2025/02/26

Abstract: **An Artificial Intelligence (AI) is revolutionizing the pharmaceutical industry by enhancing and streamlining numerous processes, ranging from drug discovery to clinical trial optimization. AI technologies help accelerate processes, reduce costs, and improve effectiveness, enabling pharmaceutical companies to achieve long-term competitive advantages. This study aims to advance the pharmaceutical industry by designing a software solution that utilizes a genetic algorithm based on artificial intelligence. In this research, drug components are represented as genes within a set of chromosomes, allowing for the optimization of pharmaceutical molecule design by exploring vast spaces of possible chemical compounds. The proposed algorithm identifies the most effective molecules while minimizing potential side effects, significantly accelerating the drug discovery process by reducing the time and cost required to produce new or advanced formulations. This innovative approach holds the potential to transform drug development and improve outcomes in the pharmaceutical sector.**

Keywords: *Artificial Intelligence, Genetic Algorithm, Evolutionary Algorithm, Software Engineering, Knowledge Base.*

How to Cite: Hossam Abdelrahman Al-Ansary. (2025). A Proposed Model for an Artificial Intelligence Algorithm to Improve Pharmaceutical Industry. *International Journal of Innovative Science and Research Technology*, 10(2), 696-708. https://doi.org/10.5281/zenodo.14930503.

I. INTRODUCTION

Artificial Intelligence (AI) is a field of computer science aimed at developing systems capable of performing tasks that typically require human intelligence. These tasks include learning, reasoning, problem-solving, perception, understanding natural language, and interacting with the environment. .[1]

The emergence of artificial intelligence (AI) has sparked a significant transformation in the pharmaceutical sector, driving a paradigm shift across multiple areas such as drug discovery, formulation development, manufacturing, quality control, and post-market surveillance. This review provides a comprehensive analysis of the diverse impacts of AI-driven technologies on the entire pharmaceutical life cycle. It explores the use of genetic algorithms, data analytics, and predictive modeling to streamline drug discovery, improve formulation optimization, and boost manufacturing efficiency.[2]

A Knowledge Base System (KBS) in the pharmaceutical industry is a centralized repository of structured and unstructured data, designed to store, organize, and retrieve critical information. It serves as a vital tool for decision-making, regulatory compliance, innovation, and operational efficiency. [3]

A Knowledge Base System is a cornerstone of modern pharmaceutical operations, enabling innovation, compliance, and efficiency across the drug development lifecycle. Its integration with advanced technologies like AI and big data analytics is transforming the industry, paving the way for faster, safer, and more effective therapies. [4]

The Research and Development (R&D) Department in pharmaceutical industries is pivotal in driving innovation and ensuring the development of safe, effective therapeutics. Its functions span the entire drug lifecycle, from discovery to post-market monitoring.[5] The pharmaceutical manufacturing process is intricate and demands high precision and rigorous oversight to guarantee the quality, efficacy, and safety of drugs. Below are the essential steps involved in drug manufacturing:

- **Drug Discovery:** Identification of chemical or biological compounds with potential therapeutic effects for specific diseases.
- **Preclinical Testing:** Evaluation of compounds in laboratory settings and animal models to assess efficacy and safety. [6]

ISSN No:-2456-2165 https://doi.org/10.5281/zenodo.14930503

- *Clinical Trials: Human testing conducted in three phases:*

- Phase I: Focuses on safety and determining the appropriate dosage.
- Phase II: Assesses efficacy and identifies potential side effects.
- Phase III: Confirms efficacy and safety in a larger patient population.

- *Formulation and Design*

- Formulation Development: Determination of the drug's final form (e.g., tablets, capsules, injections) and the inclusion of excipients such as preservatives and coloring agents. [7]
- Optimization: Ensures the formulation's chemical and biological stability.

- *Manufacturing*

- Active Pharmaceutical Ingredient (API) Production: Synthesis of the active ingredient through chemical reactions or biotechnological methods.
- Primary Manufacturing: Combining the active ingredient with excipients to produce the final drug product.
- Packaging: Packaging the drug in appropriate containers (e.g., bottles, blister packs) with accurate labeling. [8]

- *Quality Control*

- Chemical and Physical Testing: Ensures each batch meets specified standards.
- Biological Testing: Verifies efficacy and safety.
- Documentation: Maintains detailed records of all processes and tests to ensure traceability and regulatory compliance.

- *Regulatory Approval*

- Submission of Applications: Applications are filed with regulatory bodies (e.g., FDA, EMA) for marketing approval.
- Inspections: Facilities are audited to ensure adherence to regulatory standards.

- *Distribution and Marketing*

- Distribution: The drug is supplied to pharmacies, hospitals, and other healthcare providers.
- Marketing: Promotional activities are conducted in compliance with advertising regulations.

- *Post-Marketing Surveillance*

- Pharmacovigilance: Ongoing monitoring of side effects after the drug is released to the market.
- Updates: Medical information is revised based on new findings.

- *Safe Disposal*

- Waste Management: Proper disposal of chemical and biological waste in line with environmental regulations.

- *Continuous Improvement*

- Process Optimization: Ongoing refinement of manufacturing processes to enhance efficiency and reduce costs.
- New Product Development: Exploration of new therapeutic applications for existing active ingredients.

Pharmaceutical manufacturing relies on the collaboration of chemists, pharmacists, engineers, and quality assurance experts to ensure the production of safe and effective medications for patients. This research focuses on the R&D Department for Process Optimization, which involves the ongoing refinement of manufacturing processes to enhance efficiency and reduce costs by using genetic algorithm. Where The research and development team conduct numerous attempts to refine the formulation, adjust the proportions and values of the preparation, and carry out experiments until the best possible outcome is achieved. [9]

II. EVOLUTIONARY ALGORITHMS

An evolutionary algorithm is a type of artificial intelligence-based computational tool designed to solve complex problems by simulating processes inspired by biological evolution. It mimics natural behaviors such as reproduction, mutation, and recombination to iteratively improve solutions over time. Evolutionary algorithms operate through a Darwinian-like process of natural selection, where weaker solutions are eliminated, and stronger, more viable options are retained and refined in subsequent iterations. The ultimate goal is to arrive at optimal or near-optimal solutions that achieve the desired outcomes. [10]

- *Benefits of Evolutionary Algorithms*

Evolutionary algorithms offer several advantages, making them highly effective for solving a wide range of problems:

- *Increased Flexibility:*

Evolutionary algorithms can be adapted and modified to address highly complex problems across various domains. Their flexible framework allows them to meet specific objectives and tackle challenges that traditional methods may struggle with. [11]

- *Better Optimization:*

These algorithms consider a vast "population" of potential solutions, enabling them to explore a wide search space. Unlike traditional methods that may be limited to a narrow set of solutions, evolutionary algorithms are not restricted to a single approach, increasing the likelihood of finding optimal or near-optimal solutions. [12]

ISSN No:-2456-2165 https://doi.org/10.5281/zenodo.14930503

- *Unlimited Solutions:*
  Unlike classical optimization techniques that often focus on maintaining a single best solution, evolutionary algorithms can generate and present multiple potential solutions to a problem. This diversity allows decision-makers to choose from a range of viable options based on specific criteria or constraints. [13]

III. GENETIC ALGORITHMS AND EVOLUTIONARY ALGORITHMS

Genetic Algorithms (GAs) are among the most prominent and widely used algorithms within the broader category of Evolutionary Algorithms (EAs). Inspired by the principles of biological evolution, GAs utilize mechanisms such as selection, crossover, and mutation to continuously improve and optimize solutions. They are particularly effective for problems involving continuous improvement, optimization, and search in complex, multidimensional spaces. [14]

Evolutionary algorithms, including genetic algorithms, provide a powerful and flexible approach to problem-solving by leveraging the principles of natural selection and biological evolution. Their ability to explore vast solution spaces, adapt to complex problems, and generate multiple potential solutions makes them invaluable tools in fields ranging from engineering and logistics to artificial intelligence and data science. [15]

IV. GENETIC ALGORITHMS

Genetic Algorithms (GAs) are optimization techniques inspired by natural evolutionary processes, such as natural selection, mutation, and crossover (reproduction). These algorithms are used to solve complex or nonlinear optimization problems, where traditional methods are either inefficient or difficult to apply. [16]

A. Components of Genetic Algorithms:

➢ *Chromosome:*

- Represents a potential solution to the problem.
- Typically represented as a string of genes (numbers, characters, or other data).

➢ *Population:*

- A group of potential solutions (chromosomes).
- Initially generated randomly.

➢ *Fitness Function:*

- Evaluates the quality of each chromosome (solution).
- Depends on the specific problem being solved.

➢ *Selection:*

- Choosing the fittest chromosomes to produce offspring (new generations).
- Usually done using methods like "Roulette Wheel Selection" or "Rank Selection."

➢ *Crossover:*

- A process where two chromosomes (parents) are combined to produce a new chromosome (offspring).
- One or more points in the chromosome are selected, and genes are exchanged between the parents.

➢ *Mutation:*

- A random change in one or more genes of the chromosome.
- Helps maintain genetic diversity and avoid getting stuck in local optima.

➢ *Replacement:*

- Replacing the least fit individuals in the population with new offspring.
- Replacement can be complete or partial.

➢ *Mechanism of Genetic Algorithms:*

- **Initialization:** Create an initial population of solutions randomly.
- **Evaluation:** Calculate the fitness value for each individual in the population using the fitness function.
- **Selection:** Select the fittest individuals to produce offspring.
- **Crossover:** Apply the crossover process to create new individuals.
- **Mutation:** Apply mutations to the new individuals to increase diversity.
- **Replacement:** Replace old individuals with new ones.
- **Iteration:** Repeat steps 2 to 6 until a stopping condition is met (e.g., a specific number of generations or reaching a satisfactory solution).

Genetic algorithms are used in various fields such as design optimization, machine learning, robotics, and many other applications that require finding optimal solutions in large and complex search spaces. [17]

B. Genetic Algorithm Process

In genetic evolutionary optimization algorithms, techniques are used to transform a problem from its real domain into the domain of evolutionary algorithms by generating several alternative solutions. The goal is to achieve a result that is closer to the optimal solution. The evolutionary process begins with a set of random solutions, known as the population. [6] These solutions, referred to as individuals, are encoded based on the specific problem at hand. The quality of each individual is then evaluated by computing its fitness value within the initial population.

ISSN No:-2456-2165 https://doi.org/10.5281/zenodo.14930503

Following this evaluation, the current population evolves into a new population through the application of three fundamental operators, as illustrated in Fig 1. [18]

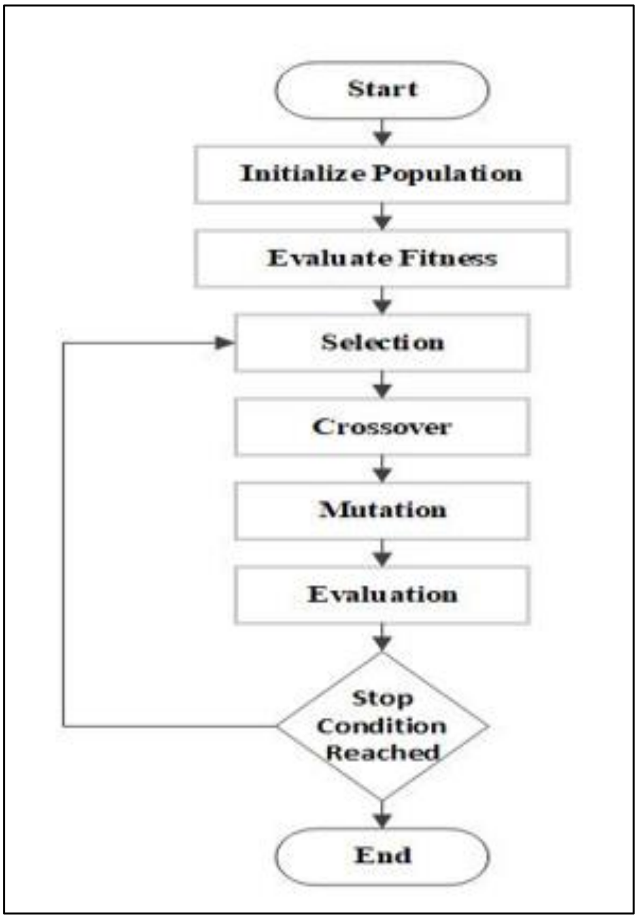

Fig 1: Genetic Algorithm Flowchart

➢ *Selection of Individuals for Reproduction:*

Individuals are chosen for reproduction using specific selection mechanisms, such as roulette wheel selection, tournament selection, or rank-based selection. These mechanisms prioritize individuals with higher fitness values, increasing their chances of contributing to the next generation. [19]

➢ *Creation of Offspring:*

New offspring are generated by applying crossover and mutation operators.

- **Crossover:** Combines genetic information from two parents to produce one or more offspring. The probability of crossover is determined based on the application.
- **Mutation:** Introduces random changes to the offspring's genetic information to maintain diversity and explore new solutions. The probability of mutation is also selected based on the application.

➢ *Computation of the New Generation:*

The new generation of the population is formed by replacing less fit individuals with the newly created offspring. This ensures that the population evolves toward better solutions over time [20]. This iterative process continues until one of the following stopping conditions is met:

- The maximum number of generations is reached.
- An optimal or satisfactory solution is found.

The algorithm terminates at this point, providing the best solution discovered during the evolutionary process.

C. GA Algorithm

```
Choose an initial population of chromosomes;
while termination condition not satisfied do
  repeat
    if crossover condition satisfied then
    {select parent chromosomes;
    choose crossover parameters;
    perform crossover};
    if mutation condition satisfied then
    {choose mutation points;
    perform mutation};
    evaluate fitness of offspring
  until sufficient offspring created;
select new population;
endwhile
```

Fig 2: Genetic Algorithm

➢ *Start:*

Begin by creating an initial population of potential solutions. This population consists of n randomly generated individuals, each representing a possible solution to the problem. [21]

➢ *Fitness Evaluation:*

Evaluate the fitness value f(x) of each individual x in the population. The fitness function measures how well each solution performs relative to the problem's objectives. Individuals with higher fitness values are considered better solutions.

➢ *New Population Creation:*

Repeat the following steps to generate a new population until the desired population size is achieved:

➢ *Selection (Reproduction):*

Select the "best" individuals from the current population to serve as parents for the next generation. The definition of "best" depends on the specific problem and is typically based on fitness values. Selection plays a crucial role in maintaining diversity within the population and preventing premature convergence, where the algorithm settles on a suboptimal solution. Choosing an appropriate selection technique is a critical step in ensuring the algorithm's effectiveness. [7]

➢ *Crossover (Recombination):*

Combine the genetic information of selected parents to produce offspring. This process mimics biological reproduction and introduces new combinations of traits into the population. [22]

➢ *Mutation:*

Introduce random changes to the offspring's genetic information. Mutation helps maintain genetic diversity and enables the algorithm to explore new areas of the search space that might not be reachable through crossover alone.

➢ *Replacement:*

Replace the old population with the new population of offspring, ensuring that the population size remains constant.

This iterative process continues until a termination condition is met, such as reaching a maximum number of generations or achieving a satisfactory solution. [23]

D. Genetic Algorithm Parameters

The parameters of the Genetic Algorithm play a crucial role in determining its performance and the quality of the solutions it produces. These parameters include the following:

➢ *Population Size:*

Selecting an appropriate population size is a critical decision. If the population size is too small, the search space becomes limited, increasing the risk of converging to a local optimum rather than the global one. Conversely, if the population size is too large, the search area expands significantly, leading to increased computational load and slower processing. Therefore, it is essential to choose a reasonable population size that balances exploration and efficiency. [24]

➢ *Crossover Rate:*

The crossover rate determines the frequency at which crossover operations occur between chromosomes within a single generation. This rate typically ranges between 0% and 100%. Setting the crossover rate is a delicate task, as it directly influences the algorithm's ability to explore new solutions. An inappropriate crossover rate can either limit diversity or introduce excessive randomness, both of which can negatively impact the algorithm's performance. [25]

➢ *Mutation Rate:*

The mutation rate specifies the proportion of genes that undergo mutation in a generation, also ranging between 0% and 100%. Like the crossover rate, the mutation rate requires careful tuning. While a higher mutation rate can introduce diversity and prevent premature convergence, an excessively high rate can disrupt good solutions. Conversely, a very low mutation rate may limit the algorithm's ability to escape local optima. Both increases and decreases in the mutation and crossover rates can have either positive or negative effects on the algorithm's outcomes. [26]

➢ *Number of Generations:*

This parameter defines the number of cycles the algorithm will execute before termination. In some cases, a few hundred generations may suffice to find a satisfactory solution, while more complex problems may require significantly more iterations. The optimal number of generations depends on the problem's complexity and the desired solution quality.

All these parameters are vital because they collectively determine the efficiency, accuracy, and overall success of the Genetic Algorithm in finding high-quality solutions. Properly tuning these parameters ensures a balance between exploration (searching new areas) and exploitation (refining existing solutions), ultimately leading to better results. [27]

E. Biological Chromosomes in GA Algorithm

➢ *The key points can be summarized as follows:*

- Chromosomes serve as the storage units for genetic information.
- Each chromosome is composed of DNA.
- Genes, which are embedded within the chromosomes, carry specific instructions.
- These genes are responsible for coding proteins.
- Every gene occupies a distinct and unique location on the chromosome.

The Chromosome in a genetic algorithm represents the set of possible combinations within the search space. It is commonly represented as a binary string of 0s and 1s as showing in Fig 3. In this paper, the Chromosome consists of the following parts of the drug, Active Ingredient (API), Inactive Ingredients (IAI), Fillers (FI) Increase the volume of the drug, Binders (BI) Ensure cohesion of the ingredients, (Colorants) Provide color, Preservatives (PR) Prevent bacterial contamination, Flavors (FL) Improve taste and Lubricants (LU) Facilitate the manufacturing process. [28]

Effectively constructing a population of API, IAI, FI, BI PR, FL and LU. For API considered two possibilities, represented by a binary string of 1 bit (19 = 2). For IAI, we considered three possible techniques, also represented by a binary string of 3 bits (24 = 8). For FI, BI, PR, Fl and UL we considered twelve different possibilities, which required 4 bits to represent (24 = 16). With this Chromosome representation, the goal of the genetic algorithm is to find the Chromosome that maximizes a fitness function. We only worked with combinations within the proposed range. For example, the range of the attribute selection is between 1 to 5. This means that combinations with the attribute selection set of 6 to 8 are not valid.

Chromosome A	1	0	0	1	1	0	1	1	0	0	1
Chromosome B	1	1	1	0	0	1	1	0	1	0	1

Fig 3: Binary String Example

ISSN No:-2456-2165 https://doi.org/10.5281/zenodo.14930503

V. PERFORMANCE CRITERIA

In the pharmaceutical industry, equations for calculating the values of drug components are used to determine the precise quantities of active ingredients and excipients (such as solvents, fillers, binders, etc.) required for drug manufacturing. These calculations depend on several factors, including the required dosage, the dosage form (tablets, capsules, syrups, etc.), and the properties of the materials used. Below are some basic equations and concepts used in calculating drug components: evaluate the average relative error across a large dataset of effort values, providing a measure of estimation accuracy. [29]

➢ *Calculating the Quantity of Active Pharmaceutical Ingredient (API)*

The active ingredient is the main component of the drug that provides the therapeutic effect. The required quantity is calculated based on the specified dosage and the number of units (e.g., number of tablets or capsules).

$$\textbf{Quantity of API} = \text{Dosage per unit} \times \text{Number of unit}$$

➢ *Calculating the Quantity of Excipients*

Excipients are inactive components added to improve the properties of the drug (e.g., stability, solubility, shape, etc.). They are calculated based on the percentage or total weight of the drug.

$$\text{Quantity of Excipient} = \text{Total weight of the drug} \times \left(\frac{\text{Percentage of excipient}}{100}\right)$$

➢ *Calculating Solution Concentration*

For liquid medications (e.g., syrups or injections), the concentration of the active ingredient in the solution is calculated.

$$\text{Concentration of API} = \frac{\text{Quantity of API}}{\text{Volume of solution}}$$

➢ *Calculating Dilution Factor*

When diluting a concentrated solution, the dilution equation is used:

$C_1 \times V_1 = C_2 \times V_2$

Where:

- $C_1 C_1$: Initial concentration.
- $V_1 V_1$: Initial volume.
- $C_2 C_2$: Final concentration.
- $V_2 V_2$: Final volume.

➢ *Calculating Molar Quantity*

Sometimes, the concentration of substances is expressed using molarity (number of moles per liter).

$$\text{Molarity} = \frac{\text{Number of moles}}{\text{Volume of solution (liters)}}$$

➢ *Calculating Loading Capacity in Capsules*

When preparing capsules, the quantity of the active ingredient and excipients is calculated based on the caps

$$\text{Quantity of API} = \text{Capsule capacity} \times \left(\frac{\text{Percentage of API}}{100}\right)$$

➢ *Calculating Overall Process Efficiency*

In pharmaceutical manufacturing, the overall process efficiency is calculated to assess losses or waste.

$$\text{Efficiency} = \left(\frac{\text{Actual quantity produced}}{\text{Theoretical quantity}}\right) \times 100$$

These equations provide a foundation for accurately calculating drug components, taking into account the dosage, pharmaceutical form, and the physical and chemical properties of the materials used.

VI. PROPOSED DESIGN

The proposed design to calculating the values of drug components is used to determine the precise quantities and accelerate the drug discovery process by reducing the time and cost required for produce a new or advanced formulation by creating a Multi-Population of Genetic Algorithm as showing in Fig 4. The drug is determined based on offspring produced through inbreeding and crossbreeding during each phase of the drug composition process. In every phase, candidates for the next generation's population are selected, with each population consisting of two chromosomes. [30] [31] The proposed algorithm can be outlined as follows:

- **Start Initialization of Population:** Begin by generating two populations of potential solutions, each comprising n individuals.
- **Population (A):** This population consists of three chromosomes. The first chromosome is termed the Active Pharmaceutical Ingredient (API) Chromosome, as illustrated in Table 5. It comprises several genes, with each gene representing the weight of API factors. The second chromosome is referred to as the Inactive Ingredients (IAI) Factors Chromosome, which also contains multiple genes. Each gene in this chromosome represents a scale of IAI factors. Similarly, the third chromosome is designated as the Fillers (FI) Chromosome, structured in the same manner as the IAI chromosome. [32]

ISSN No:-2456-2165 https://doi.org/10.5281/zenodo.14930503

- **Population (B):** This population comprises four chromosomes. The first chromosome is termed the Binders (BI) Chromosome, which contains several genes, with each gene representing a risk factor associated with BI scales. The second chromosome is referred to as the Preservatives (PR) Chromosome, which also includes multiple genes, each representing a variable of PR factors. Additionally, there are two other chromosomes: Flavors (FL) and Lubricants (LU), both structured similarly to the BI chromosome, with genes representing their respective factors. [33].

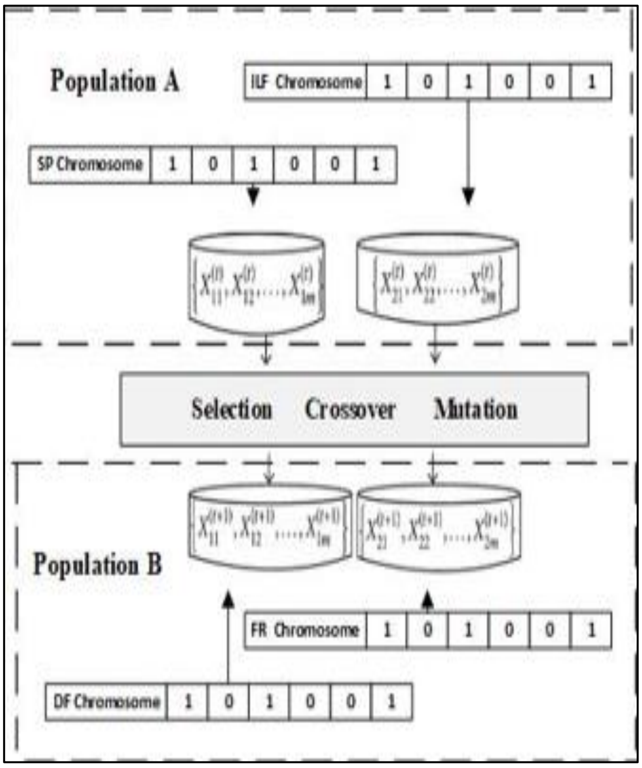

Fig 4: Multi Population of Genetic Algorithm

- **Fitness Evaluation:** Calculate the fitness value f(x)f(x) for each individual xx in Population (A). This involves assessing the performance or suitability of each solution based on predefined criteria. Once completed, repeat the same process to evaluate the fitness value for each individual in Population (B). This step ensures that the quality of potential solutions in both populations is quantified and compared effectively.

A. Active Pharmaceutical Ingredient (API) Chromosome:

The Active Pharmaceutical Ingredient (API) Chromosome is illustrated in Table 5. This chromosome can undergo multiple iterations based on the specific requirements of the system. The weights assigned to each API are predicted values.

Table 1: API Chromosome

kilogram	**Gram**	**Pound**	**Ounce**	**Milligrams**
VSS	SS	MS	LS	VLS
1	2	3	5	8
0001	0010	0011	0101	1000

B. Inactive Ingredients (IAI) Chromosome

Table 6 shows (IAI) Chromosome scaling factor for which represents the level of understanding each binder, that include percentage of (VSS), percentage of Components (SS), percentage of Components (MS), and New Components (LS).

Table 2: (IAI) Chromosome

Stabilizers	**Vehicles**	**Retardants**	**Anti-caking Agents**
%	%	%	%
1	2	3	4
0001	0010	0011	0100

Table 7 shows the ((FI) Chromosome, which also includes multiple genes, each representing a variable of PR factors.

Table 3: (FI) Chromosome

VSS	**SS**	**MS**	**LS**
%	%	%	%
2	3	4	8
0010	0011	0100	1000

C. Binders (BI) Chromosome

Table 8 shows the (BI) Chromosomes refers to the fac-tors that could lead to such as hardness, disintegration time, and drug release rate. These factors are unpredictable and unexpected.

Table 4: Binders (BI) Chromosome

	Normal	**High**	**Very High**	**Extra-High**
Adhesion	0001	0010	0011	0100
Solubility	0001	0010	0011	0100
Chemical Stability	0001	0010	0011	0100
Biocompatibility	0001	0010	0011	0100
Flowability	0001	0010	0011	0100
Compressibility	0001	0010	0011	0100
Disintegration Time	0001	0010	0011	0100

ISSN No:-2456-2165 https://doi.org/10.5281/zenodo.14930503

The other chromosomes (PR, Fl, and LU) follow a process identical to that of the BI chromosome.

After the transformation, all chromosomes (API, IAI, FI, BI, PR, FL, and LU) will be represented as genes within their respective chromosomes. These genes will then undergo standard genetic algorithm operations—Selection, Crossover, and Mutation—independently for each population (A and B).

Assuming we have a new product project divided into sub-scenarios, we assign random values to the Active Pharmaceutical Ingredient (API) in Population (A), as shown in Table 9. These random values represent initial potential solutions for the API factors within the project. These values will serve as the starting point for the genetic algorithm operations—Selection, Crossover, and Mutation—applied to Population (A) to optimize the API configuration for the new product.

Table 5: Population A

Gen0	**Gen1**	**Gen2**	**Gen3**	**Gen4**	**Gen5**	
API1	API 2	API 3	API 4	API 5	API 6	API Project 1
2.4	0.7	8	-2	5	1.1	44.1
SS	SS	VLS	SS	VLS	SS	

As illustrated in Table 10, the six APIs, which collectively account for 44.1% of the total values in the first project, have the potential to improve the results. Each of these APIs will be represented as a gene in the chromosome of the genetic algorithm.

.

Table 6: Project 1 Chromosome

APIs Chromosomes) – Population A							**API'**	**F(C)**
P1	-0.1	2	2	-3	2	0.9	13.9	0.033
P2	3.1	4	0	2.4	4.8	0	69.2	0.04
P3	-2	3	-7	6	3	3	3	0.024

The **API'** = API1 Gen0 + API Gen1 + API 3 Gen2 API Gen3 + API 5 Gen4 + API 6 Gen5

The objective is to identify the optimal set of parameters (API1:API6) that accurately maps the given input to its corresponding output. This will be achieved by leveraging the genetic algorithm to evolve the chromosome and refine the parameter values for the best possible outcome.

API' = 4 API1 + 2 API2 + 7 API3 + 5 API4 + 11 API5 + API6

To calculate the fitness function using the following equation in the regression model:

$$F(c) = \frac{1}{error} = \frac{1}{|SP\text{-}SP'|}$$

API' = 4 API1 + 2 API2 + 7 API3 + 5 API4 + 11 API5 + API6
API' = 4 x 2.4 – 2 x 0.7 + 7 x 8 + 5 x -2 +11 x 5 + 1.1
API' = 110.3

$$F(c) = \frac{1}{error} = \frac{1}{|44.1 - 110.3|} = \frac{1}{66.2} = 0.015$$

Then calculate the fitness function for all chromosomes in the population (A) with the same previous steps as shown in Table 11.

Table 7: Fitness Function for all Chromosomes

APIs Chromosomes) – Population A						**API'**	**F(C)**
2.4	0.7	8	-2	5	1.1	110.3	0.015
-0.4	2.7	5	-0.1	7	0.1	100.1	0.018
-0.1	2	2	-3	2	0.9	13.9	0.033
4	7	12	6.1	1.4	-4	127.9	0.012
3.1	4	0	2.4	4.8	0	69.2	0.04
-2	3	-7	6	3	3	3	0.024

- **Selection** – First, all chromosomes are sorted in descending order based on their fitness values. This means the chromosome with the highest fitness value (closest to 1) will be ranked first, and the one with the lowest fitness value will be ranked last.
- Rank-Based - Selection After sorting, the top-performing chromosomes (those with the highest fitness values) are selected as parents for the next generation. These chromosomes are chosen from Table 12, which contains the fitness values for all individuals.

As shown in Table 12 the best parents from population A.

ISSN No:-2456-2165 https://doi.org/10.5281/zenodo.14930503

Table 8: Best Chromosomes

	API1	API2	API3	API4	API5	API6
Scenario 1	2.4	0.7	8	-2	5	1.1
Scenario 2	-0.4	2.7	5	-0.1	7	0.1
Scenario 3	-0.1	2	2	-3	2	0.9
Scenario 4	4	7	12	6.1	1.4	-4
Scenario 5	3.1	4	0	2.4	4.8	0
Scenario 6	-2	3	-7	6	3	3

- **Crossover -** in this operation will take two parents (chromosomes) to generate a new offspring by switching segments of the parent genes. It is more likely that the new offspring (children) as shown in fig 5 will contain a good part of their parents, and consequently perform better as compared to their ancestors.

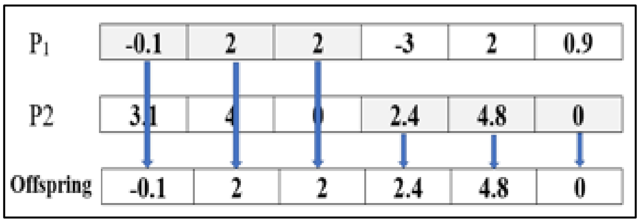

Fig 5: Crossover Operation

- The first crossover between two parents (P1, P2) as showing in fig 6 to generate new offspring.

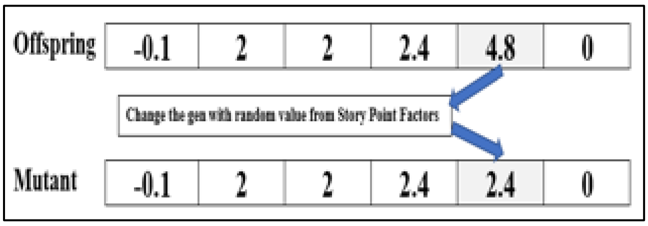

Fig 6: New Generate of Offspring

Then makes a crossover between other parents (P1, P2) with the same previous steps.

Mutation is a critical phase in the genetic algorithm, as it introduces diversity into the population and helps explore new solutions that might not be reached through selection and crossover alone. Here's how the mutation process works in your context:

➢ *Random Selection of API Factors:*

The mutation process involves randomly selecting values from the predefined API factors: VSS (Very Small Small), SS (Small Small), MS (Medium Small), LS (Large Small), and VLS (Very Large Small). These values are used to modify or switch specific genes within a single chromosome.

➢ *Gene Modification:*

During mutation, one or more genes (representing API parameters) in a chromosome are altered. For example, a gene representing API1 might be change from VSS to LS or any other random value from the available factors. This change is applied randomly to ensure diversity in the population.

➢ *Creating Enhanced Solutions:*

The mutation process generates new chromosomes that may provide improved or alternative solutions for the effort estimation problem. By introducing random changes, mutation helps the genetic algorithm escape local optima and explore a broader search space.

➢ *Controlled Mutation Rate:*

The mutation rate (probability of mutation) is typically kept low to avoid disrupting good solutions while still allowing for exploration. For example, a mutation rate of 1-5% is common.

➢ *Importance of Mutation:*

- Diversity: Mutation ensures that the population remains diverse, preventing the algorithm from converging too quickly to suboptimal solutions.
- Exploration: It allows the algorithm to explore new regions of the search space that might contain better solutions.
- Innovation: By introducing random changes, mutation can lead to innovative solutions that might not emerge through selection and crossover alone.

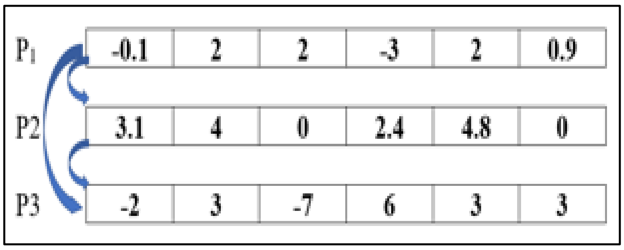

Fig 7: New Generate of Offspring

As shown in Fig 7 the describe the mutation of the chromosome by replacing the gene (4.8) with the gene (2.4) as a random value from story point factors.

Then makes the mutation for all chromosomes that generated form the crossover phase, with the same previous steps.

After complete, the mutation phase will become the first generation of the Population (A) as show the generation 1 in Table 13.

ISSN No:-2456-2165 https://doi.org/10.5281/zenodo.14930503

Table 9: Generation 1 from Population A

Generation 0 – Population A						
P1	-0.1	2	2	-3	2	0.9
P2	3.1	4	0	2.4	4.8	0
P3	-2	3	-7	6	3	3
Generation 1 – Population A						
P1	-0.1	2	2	2.4	4.8	0
P2	3.1	4	0	6	3	3
P3	-2	3	-7	-3	2	0.9

When the new generation generated from the population (A), all the phases of the genetic algorithm will be re-applied on this generation (Fitness value, Selection, Crossover, Mutation). The algorithm will be stopped when achieves the goal, after completing the mating pool of all chromosomes and completing the mutation phase for all genes in the population, that use random values of the story points factors, To ensure that all point stories are addressed to extract effort estimation for only story points Technique. can change the value of the gene of story point (API) chromosomes to binary sting if the gene contains a set of attributes.

In implementation Life Factors (IAI) technique, that the second chromosomes of the population (A) and the Population (B), that include two chromosomes (PR, UL and FI), will be appley the proposed genetic algorithm phases (Fitness value, Selection, Crossover, Mutation), with the same previous steps in all phases. The purpose of using the Multi-Genetic Algorithm to accelerate the algorithm, by dividing each of the two chromosomes into one population.

VII. PROPOSED SOFTWARE

This program was designed using the Agile software engineering methodology, with the collaboration of the Research and Development department, the program was designed by Python code to simulate a generic drug recipe, specifying ingredients, quantities, and steps. This code allows you to create or improve a drug recipe, add ingredients with quantities, specify preparation steps, and then display the entire recipe.

➢ *Python Software Code*

This program was designed using the Agile software engineering methodology, with the collaboration of the Research and Develop-ment department, the program was designed by Python code to simulate a generic drug recipe, specifying ingredients, quantities, and steps. This code allows you to create or improve a drug recipe, add ingredients with quantities, specify preparation steps, and then display the entire recipe, the Python code used to optimize drug formulations based on constraints such as efficacy, toxicity, and maximum weight. The code relies on the PuLP library to solve the optimization problem.

➢ *Recipe Code*

```
class DrugRecipe:
    def __init__(self, name):
        self.name = name
        self.ingredients = []
        self.steps = []

    def add_ingredient(self, ingredient, quantity, unit):
        self.ingredients.append({"ingredient": ingredient, "quantity": quantity,

    def add_step(self, step):
        self.steps.append(step)

    def display_recipe(self):
        print(f"Drug Recipe: {self.name}\n")
        print("Ingredients:")
        for item in self.ingredients:
            print(f"- {item['quantity']} {item['unit']} of {item['ingredient']}"
        print("\nPreparation Steps:")
        for i, step in enumerate(self.steps, 1):
            print(f"{i}. {step}")

# Example usage
drug = DrugRecipe("Pain Relief Syrup")
drug.add_ingredient("Paracetamol", 500, "mg")
drug.add_ingredient("Glycerin", 10, "ml")
drug.add_ingredient("Purified Water", 100, "ml")
drug.add_ingredient("Flavoring Agent", 5, "ml")

drug.add_step("Dissolve paracetamol in purified water under controlled heating."
drug.add_step("Add glycerin and mix thoroughly.")
drug.add_step("Incorporate the flavoring agent and mix well.")
drug.add_step("Filter the solution and fill in sterile bottles.")
drug.add_step("Label and store in a cool, dry place.")

drug.display_recipe()
```

ISSN No:-2456-2165 https://doi.org/10.5281/zenodo.14930503

➢ *Genetic Algorithm Code*

```
import random
from deap import base, creator, tools, algorithms

# Define the number of ingredients and value ranges for each ingredient
NUM_INGREDIENTS = 3
BOUNDS = [(0, 50), (0, 30), (0, 20)]  # Bounds for each ingredient in mg

def evaluate(individual):
    """ Evaluation function for drug formulation effectiveness """
    ingredient_A, ingredient_B, ingredient_C = individual
    efficacy = 5 * ingredient_A + 7 * ingredient_B + 4 * ingredient_C  # Effecti

    # Apply constraints
    penalty = 0
    if sum(individual) > 50:
        penalty += 1000  # Maximum total weight limit
    if 2 * ingredient_A + ingredient_B > 40:
        penalty += 1000  # Toxicity limit
    if ingredient_B + 3 * ingredient_C < 10:
        penalty += 1000  # Minimum beneficial compounds limit

    return efficacy - penalty,

# Set up the genetic algorithm environment
creator.create("FitnessMax", base.Fitness, weights=(1.0,))
creator.create("Individual", list, fitness=creator.FitnessMax)

toolbox = base.Toolbox()
toolbox.register("attr_float", random.uniform, 0, 50)
toolbox.register("individual", tools.initIterate, creator.Individual,
                 lambda: [random.uniform(*bounds) for bounds in BOUNDS])
toolbox.register("population", tools.initRepeat, list, toolbox.individual)

toolbox.register("mate", tools.cxBlend, alpha=0.5)
toolbox.register("mutate", tools.mutGaussian, mu=0, sigma=5, indpb=0.2)
toolbox.register("select", tools.selTournament, tournsize=3)
toolbox.register("evaluate", evaluate)
# Run the genetic algorithm
population = toolbox.population(n=50)
algorithms.eaSimple(population, toolbox, cxpb=0.7, mutpb=0.2, ngen=100, verbose=

# Print the best drug formulation
best_ind = tools.selBest(population, k=1)[0]
print("Best Drug Formulation:")
print(f"Ingredient A: {best_ind[0]:.2f} mg")
print(f"Ingredient B: {best_ind[1]:.2f} mg")
print(f"Ingredient C: {best_ind[2]:.2f} mg")
print(f"Effectiveness Score: {evaluate(best_ind)[0]:.2f}")
```

VIII. RESULT

The Genetic Algorithm parameters that were selected in the first test of the population (A), Which underwent a set of proposed algorithm operations showed the results when the program is run, the result is displayed, which includes the drug prescription, the improved drug components, as well as the preparation steps for the drug listed in the knowledge base as show in Fig-7 and Fig 8

```
Drug Recipe: Pain Relief Syrup

Ingredients:
- 500 mg of Paracetamol
- 10 ml of Glycerin
- 100 ml of Purified Water
- 5 ml of Flavoring Agent

Preparation Steps:
1. Dissolve paracetamol in purified water under controlled heating.
2. Add glycerin and mix thoroughly.
3. Incorporate the flavoring agent and mix well.
4. Filter the solution and fill in sterile bottles.
5. Label and store in a cool, dry place.
```

Fig 8: The Result of Recipe Code

ISSN No:-2456-2165 https://doi.org/10.5281/zenodo.14930503

```
96          42
97          32
98          34
99          43
100         34
Best Drug Formulation:
Ingredient A: 4.64 mg
Ingredient B: 30.20 mg
Ingredient C: 15.16 mg
Effectiveness Score: 295.20
```

Fig 9: The Result of Genetic Algorithm Code

➢ *Validation and Verification Methodology*

- Check Feasibility of Constraints- The constraints should not conflict, ensuring a feasible solution exists.
- Manually verify- The total weight does not exceed 50.Toxicity remains within limits. Minimum beneficial compounds are met.
- Run Multiple Test Cases Modify the constraints slightly and observe whether the model still finds an optimal solution. Example: Change up Bound values for ingredients to check edge cases.
- Compare with Manual Calculation - Solve the objective function manually for small values and compare results.
- Check Solver Status Ensure LpStatus[status] == "Optimal", indicating a successful optimization.

IX. CONCLUSION

This paper aimed to propose to enhance the pharmaceutical industry by leveraging artificial intelligence, specifically through a multi-gene genetic algorithm, to optimize drug formulation effectiveness. The proposed model considers critical constraints such as weight, toxicity, and beneficial components. The approach involves generating a set of candidate solutions, each characterized by modifiable properties (e.g., parents, chromosomes), these solutions are encoded in binary or decimal systems as strings of 0s and 1s, though other encoding methods are also feasible. and implemented this, a Python-based program has been developed to simulate the genetic algorithm, aiming to refine drug formulations while adhering to the specified constraints.

REFERENCES

[1]. K Huanbutta a, K Burapapadh , P Kraisit , P Sriamornsak , K Suwanpitak ,Artificial intelligence-driven pharmaceutical industry: A paradigm shift in drug discovery, formulation development, manufacturing, quality control, and post-market surveillance,European Journal of Pharmaceutical Sciences, 69:262–277, 2024

[2]. S. Bodade, N. Bajad Mam, S. Deshmukh, S, Mangesh, Artificial Intelligence in Pharmaceutical Industry, International Journal of Advanced Research in Science, Communication and Technology (IJARSCT),Volume 3, Issue 3, December 2023.

[3]. P. Sajja and R. Akerkar, Knowledge-Based Systems for Development, Advanced Knowledge Based Systems: Model Applications Research, vol. 1, pp. 1-11, 2010.

[4]. W. Lo and J. Choobineh, Knowledge-Based Systems as Database Design Tools: A Comparative Study, Intelligent Support Systems Technology: Knowledge Management, 2002.

[5]. Mak, K.-K.; Pichika, M.R. Artificial Intelligence in Drug Development: Present Status and Future Prospects. Drug Discov. Today, 24, 773–780, 2019.

[6]. Scannell, J.W.; Blanckley, A.; Boldon, H.; Warrington, B. Diagnosing the Decline in Pharmaceutical R&D Efficiency. Nat. Rev. Drug Discover., 11, 191–200, 2012.

[7]. Paul, D., Sanap, G., Shenoy, S., Kalyane, D., Kalia, K. and Tekade, R.K., . Artificial intelligence in drug discovery and development. Drug discovery today, 26(1), p.80,2021.

[8]. In Bioinformatics Tools for Pharmaceutical Drug Product Development; John Wiley & Sons, Ltd.: Hoboken, NJ, USA,pp. 1–18, 2023.

[9]. Lalitkumar K. Vora , Amol D. Gholap , Keshava Jetha , Raghu Raj Singh Thakur ,Hetvi K. Solanki and Vivek P. Chavda,Artificial Intelligence in Pharmaceutical Technology and Drug, School of Pharmacy, Queen's University Belfast, 97 Lisburn Road, Belfast BT9 7BL, UK© 2023 by the authors.Licensee MDPI.

[10]. Mafarja, M., Mirjalili, S., Aljarah, I., & Al-Zoubi, A. M. Hybridizing Genetic Algorithm with Advanced Metaheuristics for Optimization Problems. Swarm and Evolutionary Computation, 77, 101273. [DOI: 10.1016/j,2023.

[11]. Nguyen, T. T., Nguyen, M. T., & Vo, B. A Review on Genetic Algorithms and Their Applications in Optimization. Artificial Intelligence Review, 56(2),

ISSN No:-2456-2165 https://doi.org/10.5281/zenodo.14930503

1157-1190. [DOI: 10.1007/s10462-023-10412-5],2023.
[12]. Hossam Alansary," A Proposed Genetic Algorithm Model to Improve Effort Estimation in Agile Software Development" International Journal of Computer Science and Information Security (IJCSIS), Vol. 22, No.4,2024
[13]. Ochoa, G., & Veerapen, N. Genetic Algorithms: Advances, Trends, and Future Directions. Evolutionary Computation, 31(1), 1-27. [DOI: 10.1162/evco_a_00364] 2023.
[14]. Cai, Z., Sun, Y., & Chen, J. Multi-Objective Genetic Algorithms: A Comprehensive Survey. IEEE Transactions on Evolutionary Computation, 27(5), 1054-1075. [DOI: 10.1109/TEVC..3273856], 2023.
[15]. Abd Elaziz, M., Oliva, D., & Hinojosa, S. Genetic Algorithms in Machine Learning: Applications and Challenges. Expert Systems with Applications, 235, 120312. [DOI: 10.1016/j.eswa.120312], 2024.
[16]. Kar, A. K. A Review of Evolutionary Algorithms for Optimization Problems: Trends and Future Directions." Applied Soft Computing, 139, 110931. [DOI:10.1016/j.asoc..110931], 2023.
[17]. Mirjalili, S., & Heidari, A. A. Advances in Evolutionary Computation: State-of-the-Art and Emerging Trends. Swarm and Evolutionary Computation, 80, 101290. [DOI:10.1016/j.swevo.101290], 2023.
[18]. Wu, G., & Luo, Q. ,A Survey on Hybrid Evolutionary Algorithms for Large-Scale Optimization." Information Sciences, 646, 119204. [DOI:10.1016/j.ins.2023.119204], 2024.
[19]. Hossam Alansary," A Proposed Genetic Algorithm Model to Improve Effort Estimation in Agile Software Development" International Journal of Computer Science and Information Security (IJCSIS), Vol. 22, No.4,2024
[20]. Eldos, T. ,A, New Migration Model For Distributed Genetic Algorithms , Proceedings of the International Conference on Scientific Computing , , (CSC06), (pp. 128-134), 2006.
[21]. W Z Guo, L J Hao and W J. Di, Application and Development of Artificial Intelligence Technology for the Data Management and Analysis in Forestry[J], Artificial Intelligence Computational Intelligence. Aici. International Conferenc, vol. 4, no. 4, pp. 438-441, 2009.
[22]. M H Huang and R T. Rust, Artificial Intelligence in Service, Journal of Service Research, vol. 21, no. 2, pp. 155-172, 2018.
[23]. Verma and Archana, "A Survey on Image Contrast Enhancement Using Genetic Algorithm", International Journal of Scientific and Research Publications, vol. 2, no. 7, 1, pp. 2, July 2012.
[24]. Arkhipov DI, Wu D, Wu T, Regan AC, A parallel genetic algorithm framework for transportation planning and logistics management, IEEE Access 8:106506–106515, 2020.
[25]. Cheng H, Yang S, Multi-population genetic algorithms with immigrants scheme for dynamic shortest path routing problems in mobile ad hoc networks. Applications of evolutionary computation, Springer, In, pp 562–571, 2010.
[26]. Das AK, Pratihar DK, A direction-based exponential mutation operator for real-coded genetic algorithm. IEEE International Conference on Emerging Applications of Information Technology, 2018.
[27]. Dash SR, Dehuri S, Rayaguru S,Discovering interesting rules from biological data using parallel genetic algorithm, 3rd IEEE International Advance Computing Conference (IACC), Ghaziabad,, pp. 631–636, 2013.
[28]. de Ocampo ALP, Dadios EP , Energy cost optimization in irrigation system of smart farm by using genetic algorithm, 2017IEEE 9th international conference on humanoid. Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), Manila, pp 1–7, 2017.
[29]. Deb K, Pratap A, Agarwal S, Meyarivan T , A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197. 2002
[30]. Di Fatta G, Hoffmann F, Lo Re G, Urso A , A genetic algorithm for the design of a fuzzy controller for active queue management. IEEE Trans Syst Man Cybern Part C Appl Rev 33(3):313–324, 2003.
[31]. D-Manríquez, R Alvarado, B Zambrano, G Melendez, El Leal JC , An automatic document classifier system based on genetic algorithm and taxonomy. IEEE Access 6:21552–21559, 2018.
[32]. EkbataniFard GH, Monsefi R, Akbarzadeh-T M-R, Yaghmaee M et al. A multi-objective genetic algorithm based approach for energy efficient qos-routing in two-tiered wireless sensor net-works. In: wireless pervasive computing (ISWPC), 2010 5th IEEE international symposium on. IEEE, pp 80–85. 2010.
[33]. Shukla AK, Singh P, Vardhan M , A new hybrid feature subset selection framework based on binary genetic algorithm and information theory. International Journal of Computational Intelligence and Applications 18(3):1950020(1–10), 2019.

AUTHORS' PROFILES

Hossam Abdelrahman Al-Ansary.Ph.D. degree in an information system with Cairo University. His currently position is IT Director at Alandalous Pharmaceutical.