Optimizing ICU Hospitalization Prediction Models for COVID-19 Patients Using Pattern Discovery and Machine Learning

Mohsen Tajgardan
Faculty of Electrical and Computer Engineering, Qom University of Technology, Qom, Iran
tajgardan.m@qut.ac.ir

Mahboubeh Shamsi
Faculty of Electrical and Computer Engineering, Qom University of Technology, Qom, Iran
Shamsi@qut.ac.ir

Reza Khoshkangini
Sustainable Digitalisation Research Centre, Department of Computer Science and Media Technology, Faculty of Technology and Society, Malmo University,s Sweden
reza.khoshkangini@mau.se

Abdolreza Rasouli Kenari
Faculty of Electrical and Computer Engineering, Qom University of Technology, Qom, Iran
rasouli@qut.ac.ir

Abstract - The COVID-19 pandemic has underscored the critical challenges faced by healthcare systems worldwide, particularly in meeting the escalating demand for resources such as ICU beds, specialized care, and medical equipment. This shortfall has resulted in significant loss of life, highlighting the urgent need for accurate and timely diagnosis to optimize patient outcomes and reduce healthcare costs. In response to these challenges, our research focuses on developing a machine learning system capable of predicting whether patients will require ICU admission or can be managed remotely at home during peak periods of demand. Leveraging a novel two-dimensional reduction approach that combines evolutionary algorithms, Pattern Discovery, and machine learning techniques, we aim to streamline patient-collected data to train predictive models capable of forecasting ICU needs and remote care requirements. By providing healthcare systems with the ability to anticipate patient needs during critical phases of the pandemic, our predictive model empowers healthcare providers to allocate resources more effectively, optimize patient care delivery, and mitigate the impact of healthcare crises. The results of our experimental evaluation demonstrate the promising potential of our approach in addressing the pressing challenges posed by the COVID-19 pandemic and similar public health emergencies.

Index Terms - Pattern Discovery, Machine Learning, Developmental Algorithms, Feature Selection, Sample Selection.

INTRODUCTION

The World Health Organization (WHO) classified SARSCoV-2 (Covid-19) as an epidemic on March 11. However, healthcare systems faced shortages and deficiencies in responding to patients' needs for ICU beds, specialists, and personal protective equipment. The increasing demand for ICU beds created a curve surpassing the growing capacity of hospitals. Flattening the curves during each peak of the epidemic involves classifying patients into those who can recover remotely through home care and those requiring intensive care in hospitals with specialized equipment and personnel. Given the critical nature of patient classification, reducing ineffective features could be a solution to increase classification efficiency and accuracy. Attempting to place patients within the normal range of characteristics corresponding to their health status helps avoid the inclination of features towards the special care class, ultimately preserving patients' lives. Considering the impact factor of each selected characteristic aids doctors in decision-making. Implementing the PCA algorithm in the objective function has proven effective in our experiments, although the challenge lies in optimizing its input parameter value. This value must be less than the number of selected features. If

optimization algorithms simultaneously produce these two values and fixing a constant value for the component in the PCA algorithm restricts the optimization algorithm from reducing the selected features to less than this value.

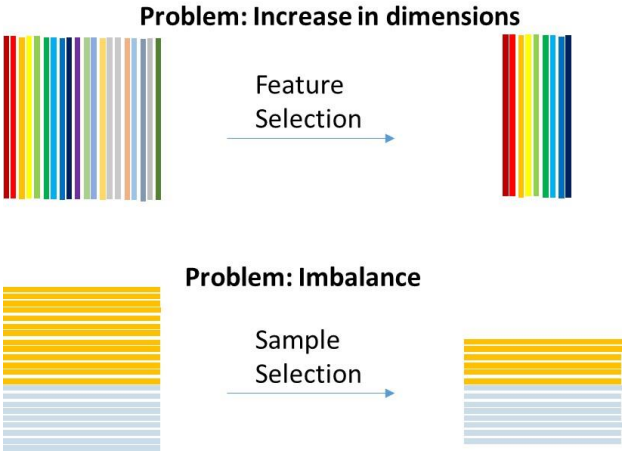


FIGURE 1. ADDRESSING DATASET CHALLENGES WITH FSIS ALGORITHM: A PROPOSED SOLUTION.

In Figure 2, two main challenges in data reduction are observed: feature abundance and imbalance in the target class among samples. Feature abundance is reduced through feature selection, while class imbalance is addressed by selecting samples. However, due to the large number of samples, it is usually not feasible to select them within the ranges determined by the genetic algorithm. In the second phase of the Feature Selection and Instance Selection (FSIS) method, which focuses on sample selection, the pattern for each feature is established using the normal ranges identified by the genetic algorithm. Subsequently, noisy data points that fall outside these patterns are eliminated from the training dataset. This process leads to an increase in the model's accuracy.

The first challenge, feature abundance, refers to having a large number of features in the dataset. This issue has been addressed by utilizing feature selection through the best evolutionary cover algorithm. Here, four evolutionary cover algorithms have been introduced, and their results have been compared, suggesting an algorithm with the minimum number of features and the highest performance as a recommendation. In the evolutionary cover algorithm, the PCA algorithm has also been used for dimensionality reduction and facilitating classification. The PCA algorithm is widely recognized as an effective method for dimensionality reduction and facilitating classification. However, if a constant value for the PCA component parameter exceeds the number of selected features, the PCA algorithm encounters difficulties. In other words, we have constrained the genetic algorithm to select a subset of features larger than this value. The use of evolutionary algorithms for feature selection, classification parameter optimization, and PCA has been implemented. It is essential for the parameter value in the PCA algorithm to be smaller than the number of selected features. This issue has been resolved by assigning the smallest number from the PCA parameter and the selected feature count, addressing the simultaneous generation problem of these two values.

The second challenge involves class imbalance in the target class, which reduces performance. To address this issue, sample selection (undersampling) has been considered. However, since the number of samples is usually much greater than the number of features, selecting samples similar to feature selection imposes a significant computational burden on the evolutionary algorithm, which inherently is time-consuming. The proposed approach in this study is as follows: The evolutionary algorithm selects the best value range for each feature with the target class as zero. This range is the normal range or pattern, and the accuracy of the classifier is the criterion for selecting this pattern. Samples within the normal range with the class labeled as zero are placed in the zero set, and samples outside the normal range with the class labeled as one are placed in the one set to remove noisy samples. This process is performed for all features, and sets are filled, but only one instance of each duplicate sample is considered. Finally, these two sets are balanced through undersampling. By combining these two sets, a training dataset is created. Then, the classifier is trained on the training data, and the prediction result of the classifier is determined as the value of the objective function. Preserving values within the normal range in some features, especially important cases in specific industries and professions, ensures the preservation of the target class.

TABLE 1. CLARIFYING KEY TERMINOLOGY USED IN OUR STUDY.

Words	**Definition**
Dateset 1	COVID-19 - Clinical Data to assess diagnosis [1]
Dateset 2	Diagnosis of COVID-19 and its clinical spectrum [2]
Dataset 3	200+ Financial Indicators of US stocks (2014-2015) [3]
Dataset 4	frican Country Recession Dataset (2000 to 2017) [4]
RBF	Is a parameter in SVM
C	Is a parameter in SVM
γ	Is a parameter in SVM
Components	Is a parameter in PCA
AUC	Is a metric

In Table 1, we elaborate on and describe the signs and abbreviations used in this article. This includes dataset names, input parameters of algorithms, and symbols employed in formulas.

Blood sugar (Feature)	Blood sugar disease(target class)
567	(Diabetes) 1
189	(Diabetes) 1
100	(Healthy) 0
87	(Healthy) 0
80	(Healthy) 0
75	(Healthy) 0
69	(Decreased blood sugar) 1

FIGURE 2. : IDEAL NORMAL RANGE DEFINITION IN BLOOD SUGAR TEST.

Our dataset encompasses data gathered from blood tests, patients' medical histories, and past medical records. Our goal is to train the system to identify, much like a blood test's normal range, which characteristics classify patients as healthy. For instance, elevated or decreased blood pressure might indicate potential risks. We aim to understand how these characteristics contribute to classifying patients into the intensive care unit (ICU) category. In Figure 2, when we arrange the data by blood sugar values, the zero-class values cluster together, forming a normal range between 70 and 110. Blood sugar values associated with class one are excluded, defining the normal range. Our criterion for each feature in the training data is for values with class zero to fall within the normal range, while values with class one fall outside this range. Subsequently, we have adjusted the training data based on this criterion, resulting in improved accuracy in evaluating the test data.

Objectives and achievements in this article:

- Flattening the curve in pandemics
- Identification of influential features in ICU admissions confirmed for COVID-19
- Selection of samples with normal range features (blood tests, etc.)
- Determination of the direction of features (blood tests, etc.) in their impact on transitioning to the high-risk class
- Ranking of features obtained from evolutionary cover algorithms

The remainder of the paper is structured as follows: Section II provides an overview of related studies. Section IV details the utilized dataset, followed by the problem formulation in Section V. The proposed approach is elucidated in Section VI. Experimental evaluation and results are presented in Section VII, followed by a discussion and summary of the work in Section VIII. Future work is outlined in Section IX.

RELATED WORK

In this study, evolutionary algorithms have been extensively used for both feature selection and pattern recognition. Reference articles [5], [6], [7] comprehensively explain the necessity and advantages of using these algorithms in solving complex problems. In the feature selection process, this study has focused on optimizing the number of features and increasing model accuracy, integrating these two objectives into a unified fitness function for optimization. Additionally, some articles, such as [8], define multi-objective evolutionary algorithms as algorithms capable of managing multiple fitness functions simultaneously. This approach enables the achievement of more optimal solutions. For example, in article [9], the similarity between features, calculated through a correlation matrix, has been considered as an additional objective alongside optimizing the number of features and increasing accuracy. This reduces redundancy among features and improves model performance. Furthermore, in article [10], optimizing the number of dataset samples and increasing the minority class count have been introduced as complementary objectives alongside the two main goals. These approaches demonstrate that the use of multi-objective evolutionary algorithms can simultaneously improve multiple aspects of the problem. In this study, four different methods for feature selection have been used, and the identified features have been improved. These methods were then compared, and the best one was employed for pattern recognition. In the following, a critical evaluation of these four feature selection techniques is provided.

I. GA-PCA

In the study by Behar et al. [11], the GA-PCA-DT algorithm is employed for breast cancer detection. The images are first preprocessed and filtered before being converted into feature sets. These features are then selected using the GA-DT algorithm, and subsequently reduced to 12 components through PCA. Notably, unlike our approach, PCA is not involved in feature selection but is instead applied afterward to reduce the feature set to 12 components as determined by PCA. Similarly, in Yang et al. [12], an intelligent variation extraction method is proposed, integrating the optimization algorithm GA into PCA, resulting in the GA-PCA technique. The outcomes of GA-PCA are compared with standard PCA, and the findings demonstrate a more significant impact of the GA optimization, highlighting improved results.

II. GA-SVM

Regarding the integration of GA and SVM approaches, Zhang et al. [13] conducted a study that employed a combination of multiple algorithms for the precise

identificat of corn types. This approach involved noise reduction techniques such as the Savitzky-Golay (SG) filter and Multiple Scattering Correction (MSC), alongside optimization algorithms like Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). Additional techniques, including Successive Projection Algorithms (SPA) and Competitive Adaptive Reweighted Sampling (CARS), were also utilized. By exploring and comparing various configurations of these algorithms, the study identified an optimal combination named MSC-(CARS-SPA)-GA-SVM. Similarly, another study [14] analyzed 10 UCI datasets, demonstrating notable improvements across all datasets. In the case of SVM with an RBF kernel, the performance is highly sensitive to the tuning of parameters c and γ. The ISMA algorithm was applied here to simultaneously reduce the feature set and optimize the SVM parameters, leading to enhanced results.

III. PSO-SVM

In the literature, an intriguing study is presented in [15], where the focus was on improving the accuracy of detecting changes in multi-temporal images. The method was evaluated across six datasets using various approaches, with the PSO-SVM method consistently outperforming others. Parameters C1 and C2, which represent the change coefficients for the global best particle and the best particle at each stage, were optimized through trial and error on three datasets. Both parameters, set to 2, produced optimal results across all datasets. Similarly, the study in [16] utilized 12 datasets from the UCI repository, yielding favorable outcomes. The feature reduction process was conducted in two stages to obtain a more relevant and informative subset of features. Initially, the REF feature selection method was used, followed by the application of a combination of SVM and DWPSO (a dynamically weighted PSO optimization algorithm) in the second stage. This method not only enhanced accuracy but also reduced execution time, as PSO tends to slow down with larger feature sets. The first stage feature selection played a crucial role in accelerating the process.

IV. PSO-PCA

The PSO-PCA algorithm was utilized in [17] to improve the accuracy of leukemia detection. Leukemia is characterized by an abnormal increase in immature lymphocytes in the blood and bone marrow, which can be classified using imaging techniques. In this study, features were extracted from images using a convolutional neural network (CNN). However, unrelated features can reduce classification accuracy and increase execution time. By integrating the PSO optimization algorithm with PCA, the method successfully selected more informative features and sped up the classification process. Similarly, Ahmed et al. [18] focused on enhancing the accuracy of COVID-19 detection from lung images. After splitting the data into training, testing, and validation sets, a 2D-CNN was used to extract features. Due to the large number of irrelevant and non-informative features, dimensionality reduction techniques such as PSO and PCA were applied. The classifiers tested included Linear SVM, k-Nearest Neighbor, and Naive Bayes, with SVM achieving the highest accuracy among them.

BACKGROUND

In this section, we offer explanations for some of the algorithms utilized in this paper.

I. Optimizer Algorithms

Optimization algorithms strategically select influential features based on the classification algorithm's optimal performance. This selection process improves the speed and overall efficiency of the classifier by eliminating ineffective data.

1) Genetic Algorithm (GA): Genetic algorithms are a type of search algorithm, but they have several distinguishing characteristics:

- Genetic algorithms do notdirectly manipulate the raw data values of the problem. Instead, they operate on a coded representation of the dataset. They explore a population of potential solutions encoded in a specific format to iteratively search for optimal or near-optimal solutions to the problem. This approach allows genetic algorithms to efficiently navigate the search space and discover solutions that may not be immediately evident when working with the raw data values directly. By applying genetic operators such as selection, crossover, and mutation to the coded representations, genetic algorithms iteratively evolve and refine the population toward better solutions. This abstraction enables genetic algorithms to address a wide range of optimization and search problems across various domains.
- When gradient information is not available, the objective function becomes the main tool for optimizing the problem. Without gradients, which show the direction of the steepest ascent or descent, optimization algorithms must rely only on evaluations of the objective function to guide their search for optimal solutions. These algorithms explore the parameter space by iteratively adjusting the input values and evaluating their corresponding objective function values to determine the direction that leads to improved performance. Although this approach may be more computationally intensive compared to gradient-based optimization methods, it remains effective for optimizing objective functions in scenarios where gradient information is unavailable or difficult to compute

Genetic algorithms (GAs) are a type of evolutionary algorithm [19], that are suitable for both constrained and unconstrained optimization tasks and are used across a wide range of domains [20], [21], [22], [23]. Unlike other optimization methods [24], [25], GAs operate on a coded representation of the problem dataset and search for a

population of potential solutions to identify optimal answers to the problem. In GA, a population of candidate solutions, represented as chromosomes, is continually generated. Over multiple generations, individuals from the current population are randomly selected as parents using GA operators, and these parents are used to produce a new population for the subsequent generation. The GA operators are outlined as follows:

- Encoding: The prevalent coding scheme is binary coding, wherein each chromosome ci comprises a vector of operators represented as binary values of 1 or 0. In this encoding, each individual feature f_i denotes whether it is present (f_i = 1) or not (f_i = 0) in that specific chromosome $C_{(i=1,\ldots,m)}$ [26].
- Generation/Initialization: The population is initialized after encoding. The initial population is created by randomly selecting individuals with labels 1 or 0. The first label indicates the inclusion of the individual predictor, while the second indicates that the predictor was not selected [26].
- Gene Selection: Various subsets of genes are selected from different training sets. The presence of each gene in the different selected gene subsets is recorded. The final gene subset is chosen from the genes with the lowest number of occurrences [27].
- Mutation: To preserve diversity from one generation to another, mutation occurs where some genes undergo mutation with low probability [26].
- Crossover: After evaluating fitness and selecting the best two chromosomes as parents, a certain portion of the genes from each parent chromosome is merged to create two children

Table 2 illustrates the terms and their corresponding definitions utilized in these experiments and Table 3 shows the full definition of the GA terms.

The structure of genetic algorithms is illustrated in Figure 3. In modeling the problem, features are translated into a gene format represented as a chromosome. In the genetic search, there is a cycle with a condition (criterion) that must be satisfied. After initialization, the fitness function is applied in each round to evaluate the performance of chromosomes/population (each solution/answer). The best chromosome is selected for the next generation, with crossover, mutation, and selection as GA operators as part of the process. These steps are repeated until the criterion is met; in our experiment, we set the criterion to the maximum generation. The best chromosome is then chosen as the best solution/response. This text describes the structure and operation of genetic algorithms in solving problems, detailing each step involved in the process.

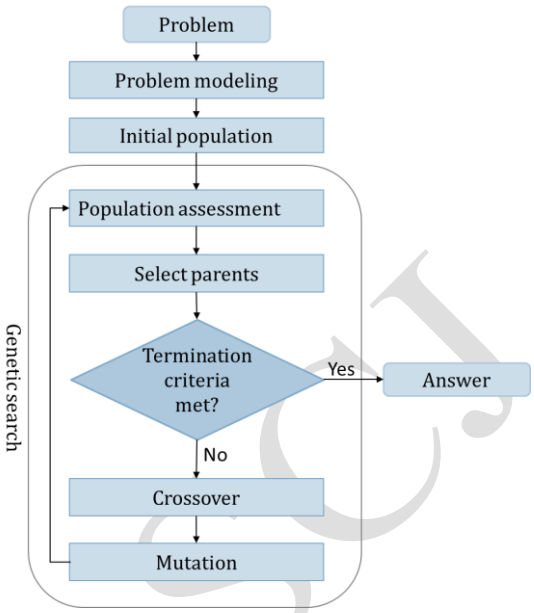


FIGURE 3. : STRUCTURE OF GENETIC ALGORITHMS.

TABLE 2. DEFINITIONS OF TERMS UTILIZED IN GENETIC ALGORITHMS.

GA Term	**Definition**
Chromosome	The overall solution to the problem
Gene	The smallest unit of a chromosome
Locus	The position of a gene on a chromosome
Alleles	The possible values that a gene can take
Phenotype	The decoded representation of the chromosome
Genotype	The encoded representation of the chromosome

TABLE 3. EQUIVALENCE TABLE OF BIOLOGICAL CONCEPTS AND GENETIC ALGORITHM ELEMENTS.

Natural evolution	**GA**
environment	Problem environment to solve
population	A set of alternative answers
produce	Repeat step
Man	Volunteer Answer
parents	Selected answers
Adaptation size	Fit value

2) Binary Particle Swarm Optimization algorithm: The binary PSO algorithm draws inspiration from the collective movement of animals, particularly birds, who select their landing spots based on safety and opportunity. Each bird's decision-making is influenced by its personal experience (pBest) and observations of other birds' movements (gBest), akin to social knowledge. In the binary PSO algorithm, these birds are represented as particles randomly positioned in the problem space. Throughout each iteration, particles adjust their positions to find more suitable locations based on the objective function, similar to genetic algorithms. While genetic algorithms are typically applied to continuous problems, we adapt the binary PSO algorithm for discrete problems in our context. Here, binary values (0 and 1) indicate the presence or absence of features. The goal of optimization problems is to minimize a variable represented by a vector $\boldsymbol{P} = [\boldsymbol{p_1}, \boldsymbol{p_2}, \boldsymbol{p_3}, \dots, \boldsymbol{p_n}]$, determined by the objective function formula, where n denotes the number of problem-specific variables.

$$P_i^t = [p_{i1}, p_{i2}, p_{i3}, \dots, p_{in}]^T \quad (1)$$

Equation (1) represents the position vector, while Equation (2) represents the velocity vector for each iteration of particle i. The notation P_{ij} represents the vector $Position_i$.

$$S_i^t = [s_{i1}, s_{i2}, s_{i3}, \dots, s_{in}]^T \quad (2)$$

Equation (3) demonstrates the effect of the internal multiplication of w by the velocity vector on the particle's position in the subsequent step. Increasing the value of w reduces the search velocity and consequently the distance traveled in the search space. However, this adjustment may result in a more precise solution being obtained at the subsequent position.

$$Object1_{ij} = wS_{ij}^t \quad (3)$$

Equation (4) represents $Object_2$, which relies on the personal experience and self-perception of the particle. If the individual experience differs slightly from the current state, it directs the particle towards a new position, with a constant coefficient $c_1$denoting its influence. The introduction of a random variable $Random_1$prevents the parameters from converging prematurely.

$$Object2_{ij} = c_1 Random_1^t \left(p_{ij}^{lBest} - p_{ij}^t\right) \quad (4)$$

Equation (5) corresponds to the best social experience. It involves exchanging individual experiences. If the current position of the particle deviates from the best social experience, it moves towards a new position influenced by a factor of c_2. The presence of a random variable $Random_2$ prevents premature parameter convergence.

$$Object3_{ij} = c_2 Random_2^t \left(p_j^{gBest} - p_{ij}^t\right) \quad (5)$$

Equation (6), all three objects affect the speed of the next step.

$$S_{ij}^{t+1} = Object1_{ij} + Object2_{ij} + Object3_{ij} \quad (6)$$

P_{ij}^{t+1} in Equation (8) represents the next position, which is determined by the sigmoid function applied to the speed. The sigmoid function, as depicted in Equation (7), normalizes the speed values between zero and one. If the likelihood is low, the position tends towards one; otherwise, it tends towards zero. Then we use Equation (8) to discretize Equation (7).

$$Sigmoid\left(S_{ij}^{t+1}\right) = \frac{1}{1 + e^{-S_{ij}^{t+1}}} \quad (7)$$

$$p_{ij}^{t+1} = \begin{cases} 1, & \text{if rand()} \le \text{Sigmoid}\left(S_{ij}^{t+1}\right) \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

The Binary PSO algorithm is versatile, suitable for a wide range of continuous problems, and its binary variant is particularly useful for discrete problems. It has demonstrated effectiveness in providing high-quality solutions across various optimization problems.

3) Whale Optimization Algorithm (WOA): WOA, a novel metaheuristic algorithm introduced by Mirjalili, draws inspiration from nature, specifically mimicking the hunting behavior of humpback whales. It operates based on a hunting strategy that involves bubble-net feeding [28].

Humpback whales exhibit a preference for hunting aggregations of krill or small fish near the water's surface. This behavior is characterized by the formation of discernible bubbles arranged in a circular or "9"-shaped path, indicating the presence of prey [28].

II. SVM Algorithm

Supervised machine learning algorithms like Support Vector Machine (SVM) serve as tools for both regression and classification tasks. SVM operates as a pattern recognition technique rooted in statistical learning theory, aiming to minimize structural risk. By ensuring effective classification, SVM enhances the generalizability of the learning system through the maximization of classification margins. A key advantage of SVM lies in its ability to mitigate issues like overfitting and the curse of dimensionality, thereby circumventing computational complexity and local optimization. This feature makes SVM particularly valuable for addressing challenges associated with limited sample sizes, high-dimensional data, and nonlinear relationships [29].

III. PCA Algorithm

Principal Component Analysis (PCA) is a technique commonly used for dimensionality reduction. As the name suggests, PCA aims to identify essential components within a dataset, allowing us to focus on a subset of features that provide the most meaningful information. By extracting these crucial features, PCA helps streamline the analysis process and allows for a more efficient examination of the data.

DATA REPRESENTATION AND PREPARATION

In this section, we introduce two datasets related to COVID-19 in the medical field and two non-medical datasets to evaluate the application of the proposed method in other domains:

I. COVID-19 - Clinical Data to assess diagnosis:

- Description: This dataset comprises anonymized data obtained from Hospital Sírio-Libanês in São Paulo and Brasilia.
- Purpose: The dataset is utilized to assess the diagnosis of COVID-19.

Data has been cleaned and scaled by column according to Min Max Scaler to fit between -1 and 1. The orginal feature consists of 54 attributes that consist of the following groups:

- Patient demographic information (03)
- Patient previous grouped diseases (09)
- Blood results (36)
- Vital signs (06)

To which the features extracted by the following methods are added:

- mean
- median
- max
- min
- diff = max - min
- relative diff = diff/median

II. Diagnosis of COVID-19 and its clinical spectrum:

- Description: This dataset contains anonymized data collected from patients examined at the Hospital Israelita Albert Einstein in São Paulo, Brazil.
- Purpose: The dataset is employed to examine the diagnosis of COVID-19 and its clinical manifestations.

The datasets comprise samples collected for SARS-CoV-2 RT-PCR testing and additional laboratory analyses carried out during hospital visits.

Figure 4 illustrates the imbalance present in both datasets, highlighting the importance of selecting metrics that are robust to class imbalance. These metrics do not exhibit bias towards the majority class.

All data underwent anonymization procedures following the best international practices and recommendations. Furthermore, standardization was applied to all clinical data to ensure a mean of zero and a unit standard deviation.

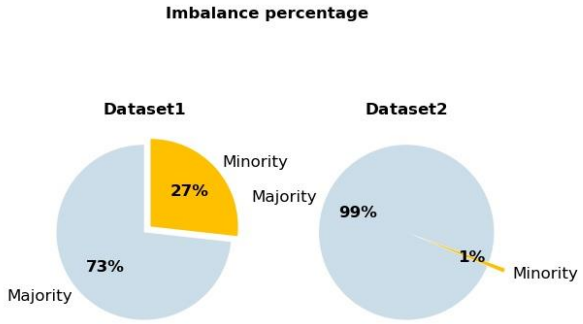


FIGURE 4. IMBALANCE PERCENTAGE IN DATASET1 AND DATASET2 .

III. 200+ Financial Indicators of US stocks (2014-2015):

This dataset compiles over 200 financial indicators for all stocks listed in the US stock market. The financial indicators are obtained from the Financial Modeling Prep API and are sourced from the 10-K filings released annually by publicly traded companies.

The last column in the dataset denotes the class of each stock, where:

- If the value of a stock increases during 2015, then class=1.
- If the value of a stock decreases during 2015, then class=0.

In essence, stocks belonging to class 1 are those that one should buy at the beginning of 2015 and sell at the end of 2015.

IV. African Country Recession Dataset (2000 to 2017):

The dataset comprises 49 feature variables and 1 target variable (referred to as the 'growthbucket' variable). It includes a total of 486 samples. Notably, 92.81% of the samples are categorized as "0" or "No Recession", while 7.82% are classified as "1" or "Recession". Consequently, the dataset exhibits class imbalance. This imbalance offers an opportunity to explore techniques for addressing such scenarios, such as Cost Sensitive Classification, Oversampling, and Undersampling. The dataset spans the years from 2000 to 2017 and covers 27 African countries, including Morocco, South Africa, Tanzania, Rwanda, Eswatini, Togo, Burkina Faso, Angola, Tunisia, Nigeria, Kenya, Burundi, Benin, Namibia, Central African

Republic, Sudan, Gabon, Niger, Sierra Leone, Lesotho, Mauritania, Senegal, Mauritius, Botswana, Cameroon, Zimbabwe, and Mozambique.

The dataset is meticulously curated to address the inquiry: "Which factors wield the greatest influence on, or serve as the most significant indicators of, recessions in Africa?"

PROBLEM FORMULATION

The demand for ICU beds often surpasses the expanding capacity of hospitals. By categorizing patients into those who can recuperate remotely with home care and those who necessitate intensive care in the ICU with specialized equipment and expertise, we aim to flatten the curves during each peak of the epidemic. Given that our classification directly impacts patients' lives, ensuring the accuracy of our classification is of paramount importance. Factors that diminish classification accuracy include:

- The multiplicity of dimensions not only reduces accuracy but also increases the complexity of classification, leading to higher resource consumption and longer execution times.
- Imbalance in the target class negatively impacts classification by causing certain metrics to lean towards the majority class, affecting the overall accuracy of the classification.

PROPOSED APPROACH

The data we collected includes blood test results, patient histories, demographics, and more. Our primary goal is to establish a normal range, similar to that of standard blood tests. The system should be capable of identifying the characteristic range for healthy individuals and selecting samples that fall within it. Additionally, it should detect potentially dangerous conditions, such as high or low blood pressure, that could warrant admission to an intensive care unit (ICU). Given that evolutionary algorithms lack ranking coverage, this article focuses on assessing the importance of each selected feature.

As part of the data preprocessing, we implemented a reduction step. Initially, we performed feature selection, followed by filtering samples based on the identified normal range. As illustrated in Figure 5, the first phase involved selecting features, which is explained in the opening section of this chapter. The second phase involved filtering samples according to the normal range. In the final phase, we provide a detailed description and analysis of the selected features.

I. Phase1: Feature Selection and parameter optimization

In feature selection, the GA-PCA method yielded the best results by pursuing dual objectives within the objective function: feature reduction and maintaining or improving performance. Leveraging the genetic algorithm, we optimized parameters for both the SVM and PCA algorithms alongside feature selection. However, a limitation arose concerning the PCA algorithm parameter, specifically the number of output components. This parameter cannot exceed the number of inputs, thereby constraining feature selection within the genetic algorithm. Consequently, the number of selected features might be smaller than the PCA parameter value, as both values are generated together in the chromosome and cannot be independently manipulated. To address this issue, we resolved to choose the smaller value between the number of features and the PCA parameter, subsequently optimizing this parameter accordingly.

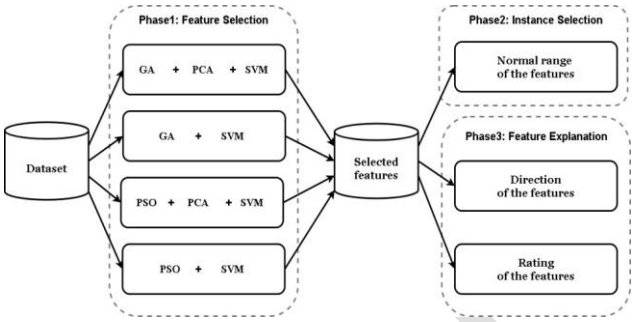


FIGURE 5. THE CONCEPTUAL VIEW OF THE PROPOSED APPROACH IS ILLUSTRATED IN THREE DISTINCT PHASES. WE COMPARED THE FOUR METHODS SEPARATELY TO ENSURE PRECISE FEATURE SELECTION IN THE SUBSEQUENT PHASES, WHICH LEADS TO IMPROVED OVERALL RESULTS. BASED ON THIS COMPARISON, WE SELECTED THE BEST FEATURE SELECTION METHOD FOR THE FOLLOWING PHASES. GA-PCA WAS CHOSEN, AS IT DEMONSTRATED SUPERIOR PERFORMANCE COMPARED TO THE OTHER THREE METHODS .

1) Pseudocode algorithm:

Algorithm 1. Feature Selection with optimize parameters

Input:
DataSet can be from dataset1 or dataset2.
Output:
SF is Selected Features,
AUC is the performance of selected features.
Let Train = split(DataSet, train_size = 0.80)
Let Test = split(DataSet, test_size = 0.20)
Let dimension = Number of feature from Train
Let ps = Population Size
Let MNI = Max Number of Iteration
Let pp = Elite Number
Let SF = GA(dimension, ps, MNI, pp)

Algorithm 1. Feature Selection with optimize parameters

Let AUC = f(SF)

Algorithm 2. GA-based Feature Selection

procedure GA(dimension, ps, MNI, pp)
Input:
dimension is Number of feature from Train,
PS is Population Size,
MNI is Max Number of Iteration,
PP is Elite Number.
Output:
Return best solution.
Definition:
Parent is Chromosomes selected from the previous iteration,
P1 is Parent1,
P2 is Parent2,
POP is Population list,
$P_{Mutation}$ is Probability of Mutation.
#Initialization (binary form)
For each i in PS
Let $POP_i.Dim$ = Random_List(dimension)
#Population Assessment
For each i in PS
Let $POP_i.Fit$ = Fitness($POP_i.Dim$)
#Select parent
Let POP = sort(POP, Fit)
Let POP = $POP_{1..PP}$
#Termination
For each j in MNI
For each i in (PP + 1 .. PS) **Step** 2
#Crossover
Let r1 = Random_Int(PP)
Let P1 = $POP_{r1}.Dim$
Let r2 = Random_Int(PP)
Let P2 = $POP_{r2}.Dim$
Let C1, C2 = Cross(P1, P2)
#Mutation
If Rand() <$P_{Mutation}$ **Then**
Let C1 = Mut(C1)
Let C2 = Mut(C2)
Let $POP_i.Dim$, $POP_{i+1}.Dim$ = C1, C2
Let $POP_i.Fit$ = Fitness(C1)

Algorithm 2. GA-based Feature Selection

Let $POP_{i+1}.Fit$ = Fitness(C2)
#Select parent
Let POP = sort(POP, Fit)
Let POP = $POP_{1..PP}$
Return $POP_1.Dim$

Algorithm 3. GA Fitness Function

procedure FITNESS(Feat_Set)
Input:
Feat_Set is Selected Features.
Output:
Return value of objective function.
Let W_1 = .01
Let W_2 = .99
#Decode chromosome
Let Len = Feat_Set.lenght
Let X_1 = decode($Feat\_Set_{1..8}$)
Let X_2 = decode($Feat\_Set_{8..15}$)
Let X_3 = decode($Feat\_Set_{15..23}$)
Let SF = $Feat\_Set_{23..Len}$
If X_1 <1 or X_2 <1 or X_3 <1 **Then**
Return 100000000
#Number of selected feature
Let $Object_1$ = sum(SF)
If $Object_1$ <= 0 **Then**
Return 100000000
#Performance measurement by classifier
Let $Object_2 = f(Feat\_Set)$
If $Object_1$ <= 0 **Then**
Return 100000000
Let $Object_2 = Object_2^{-1}$
Return $W_1 * Object_1 + W_2 * Object_2$

Algorithm 4. Classifier

procedure f(Feat_Set)
Input:
Feat_Set is Selected Features.
Output:
AUC is the performance of selected features.
#Decode chromosome
Let Len = Feat_Set.lenght
Let X_1 = decode($Feat\_Set_{1..8}$)

Algorithm 4. Classifier

Let X_2 = decode($Feat\_Set_{8..15}$)
Let X_3 = decode($Feat\_Set_{15..23}$)
Let SF = $Feat\_Set_{23..Len}$
#Select subset from chromosome
Let Train_subset = select(SF, Train)
Let Test_subset = select(SF, Test)
#PCA
Let n_components = min(SF, X_3)
Let pca = PCA(n_components)
Let Train_subset.x=pca.fit_transform(Train_subset.x)
Let Test_subset.x=pca.transform(Test_subset.x)
#SVM
Let Classifier= SVM(kernel='rbf', C=X_1, γ=X_2)
Classifier.fit(Train.x, Train.y)
Let proba = Classifier.predict_proba(Test.x)
#Metric
Let AUC = Metric(Test.y, proba)
Return AUC

Algorithm 1 takes a dataset as input and outputs the selected features along with their corresponding AUC values. It divides the dataset into training and test sets in an 80:20 ratio, respectively, and sets the parameters for the genetic algorithm. In line 5, the total number of features is specified. Line 6 sets the number of individuals in the initial population. Line 7 determines the number of generations for the genetic algorithm. Line 8 sets the number of elite individuals eligible for reproduction. Line 9 executes the genetic algorithm to obtain the selected features. Finally, the AUC value of the selected features is computed in the last line.

Algorithm 2 is a genetic algorithm that takes the number of features, initial population size, number of generations, and number of elite individuals as input. It outputs a binary string representing the selected features, where '1' indicates the presence of a feature and '0' indicates absence. At the outset, the population is randomly initialized in lines 6 and 7. The fitness of each chromosome is evaluated using the objective function in lines 1 and 9. Next, parents are selected based on their fitness, with the top-performing individuals considered elites. Elite parents are chosen first, followed by selection from the remaining individuals, as depicted in lines 12 and 13.

Subsequently, two parents are randomly chosen from the elite population, and their chromosomes are combined to generate two children, as shown in lines 18 to 22. Mutation is then applied to the children with a low probability in lines 24 to 26. The fitness of the children's chromosomes is calculated, and the best-performing ones are selected as parents for the next generation. This process continues until the specified number of generations is reached. Finally, the best chromosome is returned as the optimal solution.

In Algorithm 3, the objective function is defined. It takes a chromosome as input and returns the value of the objective function. Firstly, the chromosome is decoded to obtain the selected features, represented by '1's in the chromosome. The number of selected features is then counted. The algorithm has two objectives: reducing the number of features and maximizing performance. For the first objective, which aligns with the optimization policy of the genetic algorithm, the number of selected features is counted without modification and included in the output. For the second objective of maximizing performance, the inverse of the performance metric is calculated to ensure that higher values correspond to better performance. Both objectives are then weighted by their importance factors, multiplied, and summed to obtain the value of the objective function.

In Algorithm 4, the input is a chromosome, and its corresponding function is returned. The chromosome is decoded to determine which genes are responsible for constructing and optimizing the C and Gamma parameters in the Radial Basis Function Support Vector Machine (RBF-SVM), and which genes are responsible for constructing and optimizing the number of components for Principal Component Analysis (PCA). Before classification, the features are transformed into components using PCA, which simplifies the classification process and enhances accuracy. The classification is then performed using SVM, and the accuracy of the classification is assessed using the Area Under the Curve (AUC) metric.

2) Fitness: The fitness of each chromosome is assessed by the objective function, which takes into account the selection of features and parameters of the algorithms pertinent to the problem. The fitness value assigned to each chromosome is a positive numerical representation of its suitability. During the selection stage, the probability of selecting each chromosome is determined based on its fitness, ensuring a proportional representation according to the appropriate size of each chromosome.

Figure 6 illustrates the structure of the objective function utilized in the GA-PCA algorithm. This algorithm optimizes the parameters of RBF-SVM and PCA, alongside feature selection. Initially, both the training and test datasets are transformed into components using PCA. Subsequently, SVM is trained using the training components, and predictions are made for the test data. The performance of the model is then evaluated using the AUC metric.

Equation (9) outlines the calculation of the fitness function, which comprises two distinct objectives.

$$Fitness = W_1 * Obj_1 + W_2 * Obj_2 \quad (9)$$

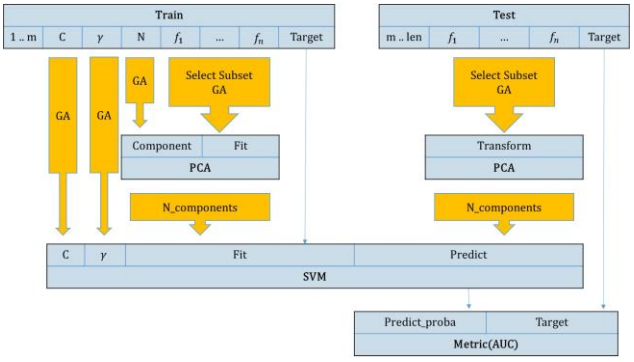


FIGURE 6. THE STRUCTURE OF THE GA-PCA FITNESS FUNCTION.

Where W_i represents the weight of each objective in the fitness function. The sum of these weights should equal 1, as indicated in Equation (10) below.

$$W_1 + W_2 = 1 \tag{10}$$

Equation (11) computes the AUC value, representing the performance of the classifier on validated COVID-19 data for predicting the need for ICU intensive care. A larger AUC value indicates better performance. In the context of the evolutionary algorithm's objective function, optimization involves minimizing the objective function. Therefore, the performance of the classifier is inversely related to the objective function, with its reduction represented by a negative power, which increases as the performance decreases.

$$Obj_1 = f(x)^{-1} \tag{11}$$

Equation (12) shows the number of selected features that should be reduced.

$$Obj_2 = \sum_{i=1}^{n} Feature_i \tag{12}$$

string of discrete variables, binary values, or continuous values. As depicted in Figure 7, f is a bit string of length n , where a value of 1 indicates the presence of a feature in that column, and 0 indicates absence. The penalty coefficient C and the parameter γ are related to the SVM parameters that require optimization. n components represents the PCA parameter, specifying the number of output columns.

2) Architecture: In Figure 8, you can observe the architecture of the GA-PCA approach. Prior to entering the SVM algorithm, the data undergoes dimensionality reduction through the PCA algorithm. Subsequently, the processed data is fed into the SVM algorithm. The performance of SVM is notably influenced by its core parameters, namely, the gamma γ and penalty coefficient C. Enhancing classification accuracy hinges on selecting appropriate parameters. Numerous methods exist for parameter optimization, among which genetic algorithms stand out. Feature subset selection is another critical aspect affecting classifier performance. Extraneous features contribute additional, often irrelevant information, which can inflate computational complexity and diminish classification accuracy. By managing both feature selection and parameter optimization, algorithms like genetic algorithms streamline this process.

II. Phase2: Instance selection with normal range

Limiting the normal range of features has numerous applications across various fields, including research, industries, and sciences. This practice involves defining the acceptable range for each feature relative to others, ensuring a certain criterion is consistently met.

In our research, it entails establishing a range of characteristics within which a patient does not require specialized care. For instance, in the heavy transportation industry, it could involve determining the optimal load capacity for vehicles to minimize wear and tear, ensuring they remain in optimal condition and avoid transitioning from a functional state to a breakdown condition. In our dataset, we have two classes: the first class represents

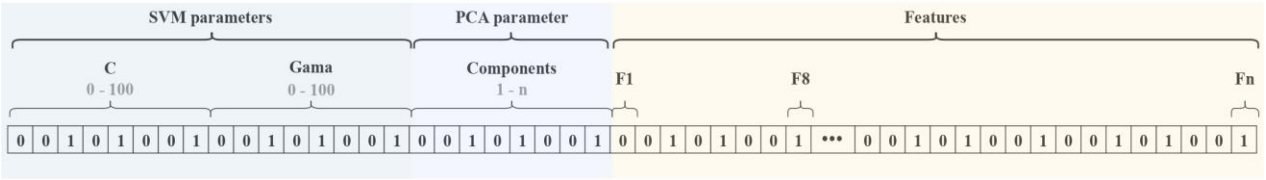


FIGURE 7. CHROMOSOME REPRESENTATION IN GA-PCA.

2) Chromosome: In a genetic algorithm, a chromosome (sometimes referred to as a genome) represents a proposed solution to the problem being solved by the algorithm. It consists of a set of parameters that define the solution. The chromosome is the practical representation of the solution in the implementation of the algorithm. Depending on the nature of the problem, a chromosome can be encoded as a

individuals in a healthy or non-critical condition, denoted by zero, while the second class indicates individuals who require hospitalization in the ICU, denoted by one.

During the data preprocessing stage, when separating samples for training, we follow a specific procedure. For each feature, we select samples from the zero class that fall within the normal range and exclude any zero-class

samples outside this range. Similarly, we exclude class one samples within the normal range and select those outside it. We ensure that the number of samples selected from both classes is equal to the size of the smaller set.

This approach aims to enhance the performance of the fitness function by focusing on samples that are most relevant to distinguishing between the two classes. Indeed, this operation is crucial for selected features because any adjustments made to a feature can significantly affect the determination of the ranges for other features.

For instance, let's consider the scenario of estimating the load range in cargo vehicles. If we take the number of wheels of the vehicle as a feature, whether the vehicle has one wheel or eighteen wheels would drastically impact our load range estimation.

Similarly, in our dataset, modifying the values or ranges of certain features can greatly influence the determination of normal ranges for other features. Therefore, it's essential to carefully preprocess and select features to ensure the accuracy and reliability of our analysis and predictions.

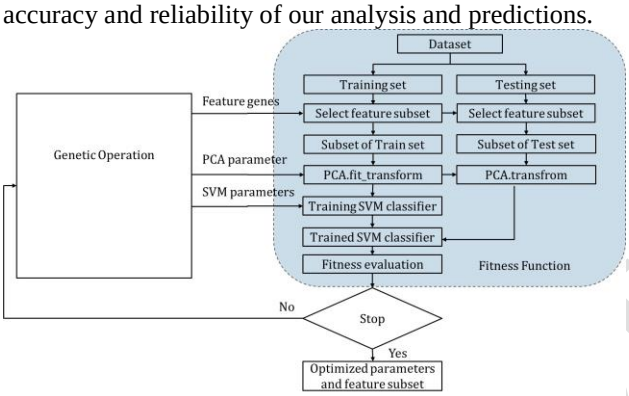


FIGURE 8. CONCEPTUAL OVERVIEW OF GA-PCA IN PHASE ONE.

This process is crucial for defining the normal ranges for features. As depicted in Figure 9, the genetic algorithm generates the normal ranges. Each feature is associated with genes responsible for determining its upper and lower bounds. To simplify computations, these bounds are specified as percentages and decoded within the target function.

which are balanced and merged through under-sampling to create the training dataset. This dataset is then used for classifying the test data. The optimal normal ranges for features are determined based on the highest criterion, such as the F1-score.

Feature1	Feature2	Feature3	Target
13	0.2	2000	0
23	0.4	2134	0
34	0.4	2234	1
45	**0.4**	**2278**	0
56	0.56	2278	1
78	**0.56**	**5000**	0
89	0.6	5100	0
90	0.7	6000	1
91	0.8	6000	1

FIGURE 10. INSTANCE SELECTION PROCESS BASED ON NORMAL RANGES GENERATED BY GENETICS.

We divide the steps of sample selection with normal range in the objective function into 4 steps and describe them separately:

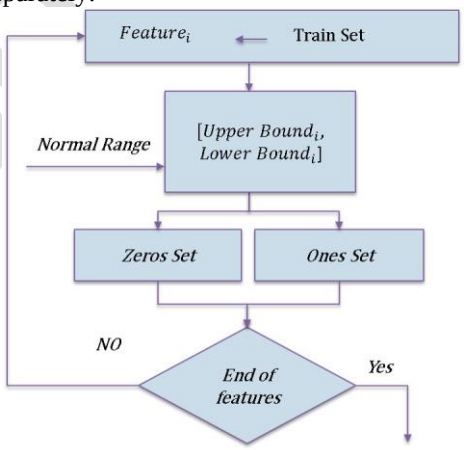


FIGURE 11. INSTANCE SELECTION PROCESS WITH NORMAL RANGES.

FIGURE 9. CHROMOSOME REPRESENTING NORMAL RANGE.

In the objective function, illustrated in Figure 10, samples within the normal range for each sorted feature (class zero) are segregated, while samples outside this range (class one) are separated. This results in two sets,

1) Step1: Building a set of zeros and ones: In Figure 11, we place a loop around the number of features, representing the iteration over each feature, and extract the normal range generated by the genetic algorithm for each one. We extract samples belonging to class zero whose

values fall within the normal range and include them in the set labeled as zeros. Similarly, we include samples belonging to class one whose values lie outside the normal range in the set labeled as ones. This process continues until we have iterated through all the features. Thus, we obtain two sets: one containing samples with class zero within the normal range and the other containing samples with class one outside the normal range.

2) Step2: Under-sampling: Exactly, in the previous step, the number of sets of zeros and ones may not be equal. To address this, we perform under-sampling to balance the two sets.

3) Step3: Building a train set: We merge the set of zeros and ones to create a new training dataset. This training set is correctly balanced in terms of the target class.

4) Step4: The value of the objective function: The XGBoost classifier is trained using the training set from the previous stage, and then tested using the reserved test dataset. The performance is evaluated using the F1-score criterion. The resulting score guides the genetic algorithm in selecting the best normal ranges based on the function value. This process iterates within the genetic algorithm until convergence, aiming to optimize the selection of normal ranges for the features.

III. Description of selected features

In this section, we will discuss the direction and ranking of the selected features from the initial stage.

1) Impact of feature direction in the target class: Determining the direction for features has numerous applications in medicine, industry, and various scientific fields. It indicates the trend in which each attribute can move relative to other attributes to transition from one target class to another. In this study, it illustrates the direction in which a feature tends to move as the patient's condition deteriorates.

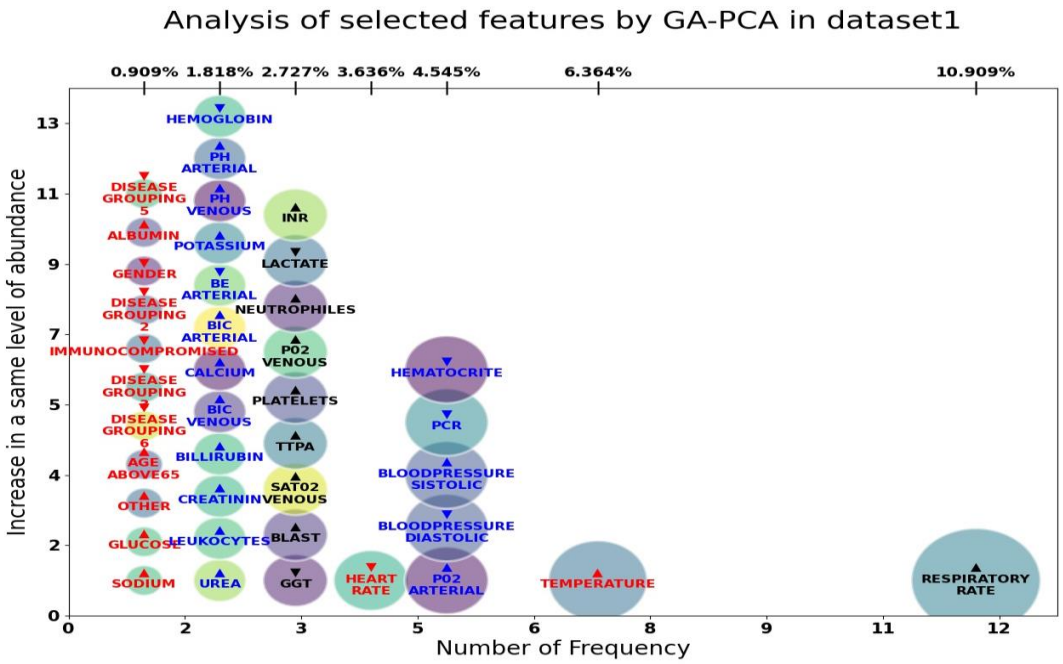


FIGURE 12. THE BUBBLE GRAPH DEPICTING THE RANK, DIRECTION, AND FREQUENCY OF SELECTED FEATURES ACROSS DIFFERENT FOLDS BY GA-PCA IN DATASET 1.

2) Ranking the selected features in the evolution algorithm and classifier: In feature selection using evolutionary algorithms, the importance of selected features is not inherently provided. To determine the importance of these features, we divide the data into different segments and feed them into the proposed algorithm. If a feature appears frequently across these segments, it indicates its significance. Thus, we count the frequency of each selected feature across all segments. The frequency of a selected feature across all segments indicates its importance. As shown in Equation (13), the ranking of a feature is calculated as the frequency of that feature divided by the sum of frequencies of all features. The results are visualized in Figure 12 using a bubble chart.

$$Ranking_i = \frac{f_i}{\sum_{j=1}^{n} f_j} \tag{13}$$

The direction for each sorted feature is determined by observing the highest number of class one samples in the upper or lower range. These two intervals start from the middle or average and extend to the top or bottom.

In Figure 12, the lower horizontal axis illustrates the frequency of each selected feature across different categories, while the upper horizontal axis represents the ranking percentage of each selected feature. The vertical axis indicates the number of features at each frequency level. Each bubble represents a selected feature, with its size indicating its importance. The direction of each feature is also depicted within its respective bubble. Features with larger circles on the right side are considered more important and stand out from the rest. Notably, features such as breathing rate and fever are among the initial selections made by the evolutionary algorithm.

EXPERIMENTAL EVALUATION AND RESULTS

I. Evaluation methods

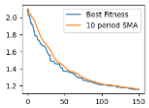
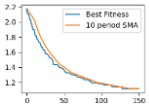
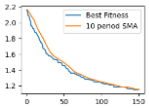
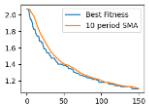
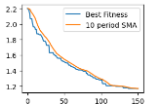
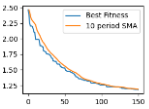
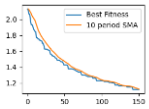
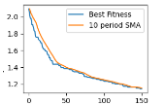
In Dataset 1, as depicted in Figure 4, there are two classes, labeled 0 and 1, which are imbalanced. Class 0 constitutes 73% of the data, while class 1 constitutes 27%.

On the other hand, in Dataset 2, as shown in the same figure, the imbalance is even more pronounced. Here, the ratio of class 0 is 99%, while the ratio of class 1 is only 1%.

Due to this class imbalance, the research has transitioned from using Accuracy to F1-Score as the evaluation metric. However, when calculating the F1-Score, division by zero issues may arise due to the limited number of samples in batch processing. Therefore, to address this, the Area Under the Curve (AUC) metric is utilized, obtained via batch processing to compute P-value and T-test.

Table 4 outlines the parameters and results of the GA-PCA algorithm. Each row represents a specific category, identified by a number. Within each category, the algorithm optimizes the SVM parameters (C and Gamma) and the PCA component parameter. The outputs for each batch include the execution time, a convergence graph, and performance metrics, measured by the AUC. Batch

TABLE 4. EQUIVALENCE TABLE OF BIOLOGICAL CONCEPTS AND GENETIC ALGORITHM ELEMENTS.

Step	C	γ	components	AUC	Time	Convergence curve
1	99	7	172	0.99	385	
2	13	1	206	0.99	342	
3	23	32	14	1.0	346	
4	11	1	187	0.99	357	
5	38	18	81	0.99	414	
6	45	11	65	0.95	365	
7	15	5	96	0.99	405	
8	64	73	159	0.97	361	
Average				0.99	372	

execution is particularly useful for comparing Static Analysis results with other algorithms. It reduces the computational load, especially during the classifier's training phase, leading to significant time savings. Additionally, the convergence diagram of the objective function is smoothed by a Simple Moving Average (SMA) with a period of 10, offering a clearer view of convergence trends within each category.

parameters for SVM, as well as the component parameter for PCA, are optimized by the algorithm within each category. The execution time, convergence graph, and performance, indicated by the AUC, are the outputs obtained for each batch. Batch execution proves advantageous for comparing the results of Static Analysis with other algorithms. It effectively reduces the computational load, especially during the training phase of

TABLE 5. FEATURE SELECTION RESULTS FOR DATASET 1.

Step	# of features	Characteristic value
1	16	74, 75, 80, 87, 90, 107, 117, 129, 134, 147, 182, 195, 203, 206, 208, 227
2	12	9, 17, 76, 110, 132, 140, 153, 162, 212, 213, 215, 216
3	16	4, 6, 19, 33, 50, 75, 85, 106, 159, 197, 198, 203, 206, 207, 214, 215
4	10	12, 76, 89, 104, 151, 194, 195, 198, 215, 228
5	17	32, 37, 43, 47, 55, 67, 68, 81, 98, 110, 115, 133, 138, 144, 198, 203, 220
6	14	7, 10, 49, 66, 95, 109, 113, 132, 142, 180, 186, 195, 197, 206
7	12	3, 13, 41, 73, 90, 131, 162, 180, 209, 220, 222, 227
8	13	20, 34, 47, 48, 58, 84, 111, 117, 144, 166, 184, 198, 221
Average	13.75	'RESPIRATORY_RATE' is common feature to all Steps

TABLE 6. AUC RESULTS OF THE PROPOSED METHOD COMPARED TO OTHER FEATURE SELECTION METHODS IN DATASET 1.

Feature Selection Method	Fold1	Fold2	Fold3	Fold4	Fold5	Fold6	Fold7	Fold8	Average
1) Binary PSO-SVM	0.95	0.96	0.96	0.98	0.88	0.86	0.96	0.95	0.94
2) Binary PSO-PCA	0.97	0.94	0.96	0.99	0.94	0.93	0.95	0.94	0.95
3) WOA-SVM	0.92	0.95	0.87	0.91	0.87	0.75	0.90	0.90	0.88
4) WOA-PCA	0.93	0.93	0.94	0.89	0.91	0.87	0.92	0.93	0.92
5) GA-SVM	**0.99**	0.98	0.98	**0.99**	**0.99**	**0.96**	**0.99**	**0.97**	0.98
6) GA-PCA (Proposed)	**0.99**	**0.99**	**1.0**	**0.99**	**0.99**	0.95	**0.99**	**0.97**	**0.99**
7) Random Forest	0.95	0.79	0.91	0.82	0.82	0.76	0.94	0.855	0.86
8) XGBoost	0.92	0.87	0.87	0.83	0.86	0.78	0.89	0.88	0.87
9) AdaBoost	0.94	0.83	0.89	0.86	0.84	0.83	0.87	0.87	0.87
10) Gradient Boosting	0.93	0.88	0.89	0.90	0.84	0.79	0.97	0.85	0.89
11) Decision Tree	0.93	0.85	0.90	0.83	0.85	0.81	0.90	0.84	0.87
12) Variance (Filter)	0.90	0.85	0.75	0.76	0.72	0.75	0.88	0.87	0.81
13) Pearson Correlation (Filter)	0.86	0.69	0.79	0.69	0.68	0.83	0.87	0.81	0.78

Table 5 presents the parameters and outcomes of the GA-PCA algorithm. Each row corresponds to a specific category identified by a number. The C and Gamma

the classifier, resulting in significant time savings. Additionally, the convergence diagram of the objective function is complemented by the Simple Moving Average (SMA) diagram, calculated with a period of 10, providing

a smoother representation of the convergence trends within the same category.

In Table 6, each row corresponds to a category identified by a number, where the selected feature names and the number of selected features are listed.

II. Comparison of three optimization algorithm

In the Fitness function, we pursue two goals: one is to enhance performance, and the other is to limit the number of features. Therefore, to assess the result, we need to examine the combined outcome. Figure 13 compares the performance of six algorithms. In part (a), GA-PCA exhibits the best results with less distortion compared to other approaches. It shows comparable results to GA-SVM but with less distortion. In part (b), WOA-PCA displays slightly worse AUC but not significantly different from the best AUC. However, it exhibits considerable distortion compared to WOA-SVM, yet it remains close to AUC. In part (c), PSO-PCA shows visible distortion, but it still achieves a respectable AUC. Compared to PSO-SVM, there is some distortion, but it remains close to AUC.

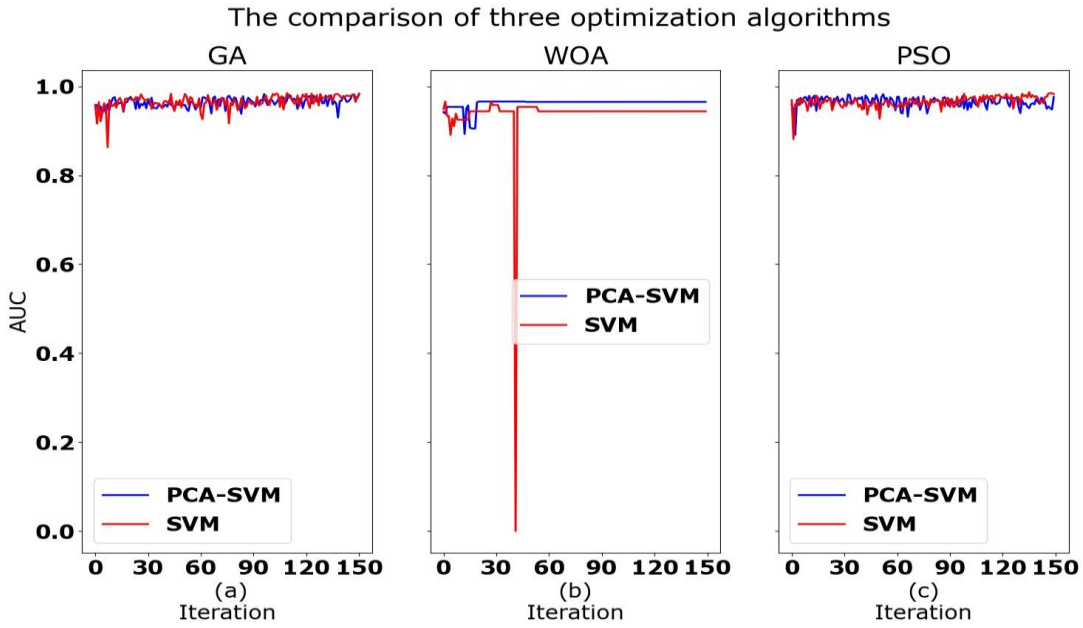


FIGURE 13. COMPARATIVE ANALYSIS OF THREE OPTIMIZATION ALGORITHMS BASED ON AUC SCORES FOR DATASET 1.

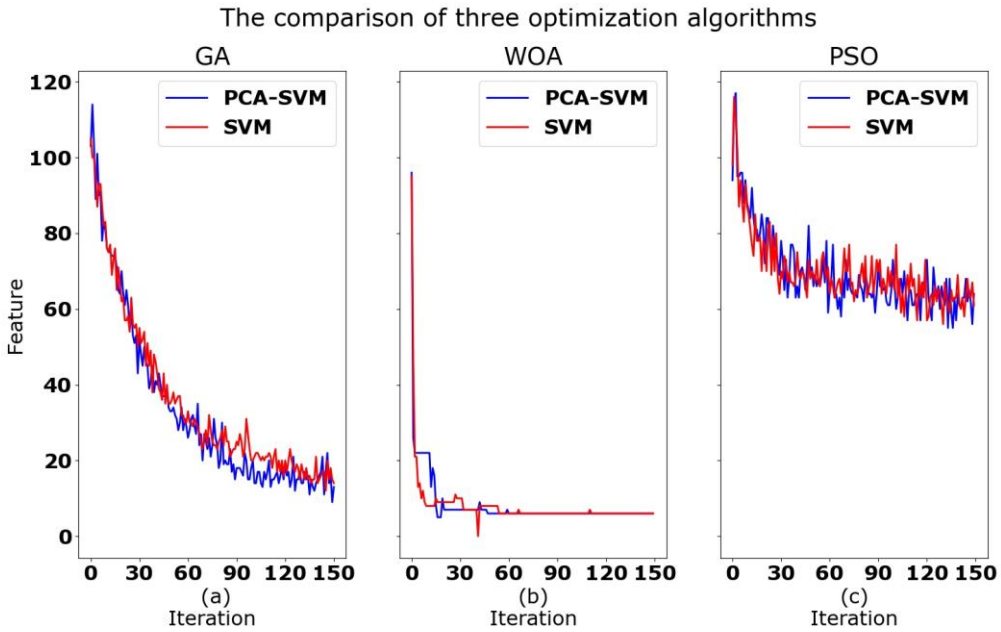


FIGURE 14. COMPARISON OF THREE OPTIMIZATION ALGORITHMS WITH FEATURE OF DATASET 1.

Therefore, PCA has effectively reduced distortion across these algorithms.

Figure 14 compares the feature reduction achieved by six algorithms. In part (a), GA-PCA demonstrates the best to lower performance compared to competitors. Part (c) selects numerous features, deviating from our goal of minimizing the feature count. Therefore, the genetic algorithm outperforms others in this aspect, and PCA effectively accelerates the reduction of selected features.

TABLE 7. AVERAGE COUNT OF FEATURES FOR THE PROPOSED METHOD COMPARED TO OTHER FEATURE SELECTION METHODS IN DATASET 1.

Feature Selection Method	Fold1	Fold2	Fold3	Fold4	Fold5	Fold6	Fold7	Fold8	Average
1) Binary PSO-SVM	60	61	54	59	63	61	56	56	58.75
2) Binary PSO-PCA	60	60	63	64	73	81	53	52	63.25
3) WOA-SVM	**4**	12	**9**	13	7	**3**	6	7	7.62
4) WOA-PCA	9	**11**	**9**	**8**	**4**	6	**4**	**5**	**7**
5) GA-SVM	20	21	14	16	18	17	13	18	17.12
6) GA-PCA (Proposed)	16	12	16	10	17	14	12	13	13.75
7) Random Forest	16	12	16	10	17	14	12	13	13.75
8) XGBoost	16	12	16	10	17	14	12	13	13.75
9) AdaBoost	16	12	16	10	17	14	12	13	13.75
10) Gradient Boosting	16	12	16	10	17	14	12	13	13.75
11) Decision Tree	16	12	16	10	17	14	12	13	13.75
12) Variance (Filter)	17	12	16	10	17	13	12	13	13.88
13) Pearson Correlation (Filter)	16	12	16	10	17	14	12	13	13.75

TABLE 8. STUDENT'S T-TEST AND P-VALUE OF DATASET 1.

Feature Selection	Average AUC	Average Feature	AUC		Total Time
			T-test	P-value	
1) Binary PSO-SVM	0.94±0.04	58.75±2.90	2.73	0.008	5254
2) Binary PSO-PCA	0.96±0.02	63.25±9.11	3.09	0.004	9258
3) WOA-SVM	0.88±0.06	7.63±3.31	4.55	0.0002	1263
4) WOA-PCA	0.92±0.02	**7.00±2.45**	6.22	0.00001	1476
5) GA-SVM	0.99±0.01	17.13±2.57	**-0.07**	**0.47**	2105
6) GA-PCA (Proposed)	**0.99±0.02**	13.75±2.28	—	—	2973
7) Random Forest	0.86±0.7	13.75±2.28	5.00	0.00009	0.28
8) XGBoost	0.87±0.04	13.75±2.28	7.63	<0.00001	0.28
9) AdaBoost	0.87±0.04	13.75±2.28	8.31	<0.00001	0.28
10) Gradient Boosting	0.89±0.05	13.75±2.28	5.00	0.00009	0.52
11) Decision Tree	0.87±0.04	13.75±2.28	7.65	<0.00001	**0.18**
12) Variance (Filter)	0.81±0.07	13.88±2.42	6.68	<0.00001	0.40
13) Pearson Correlation (Filter)	0.78±0.07	13.75±2.28	7.20	<0.00001	9.30

fit and is close to achieving the minimum number of features while also achieving the highest AUC. Part (b) converges rapidly and significantly reduces the number of features; however, this extreme reduction in features leads

In Figures 13 and 14, the methods are compared using the entire dataset over 150 iterations (indeed, in feature selection using optimization algorithms, as the number of iterations increases, more features are progressively

eliminated. Once the number of features reaches a reasonable threshold during any iteration, the execution of the program can be halted. For this reason, we set the maximum number of iterations to 150.). In contrast, the other method divides the data into smaller categories and evaluates different feature selection techniques within each category. Table 6 lists the feature selection methods in the rows, with the results for each category of Dataset 1 presented in the columns. The average results are provided in the last column. Similarly, Table 7 shows the feature selection methods in the rows and the selected features for each category of Dataset 1 in the columns, with the average number of selected features summarized in the final column.

In Table 9, the rows represent the feature selection methods, while the columns display the results in each

TABLE 9. AUC RESULTS OF THE PROPOSED METHOD COMPARED TO OTHER FEATURE SELECTION METHODS IN DATASET 2.

Feature Selection Method	Fold1	Fold2	Fold3	Fold4	Fold5	Fold6	Fold7	Average
1) Binary PSO-SVM	0.92	0.92	0.98	0.97	0.96	0.99	**1.00**	0,96
2) Binary PSO-PCA	0.93	**0.98**	0.97	**1.00**	0.97	**1.00**	0.99	0.97
3) WOA-SVM	0.92	0.95	0.99	0.91	0.97	**1.00**	0.98	0.96
4) WOA-PCA	0.93	0.96	0.99	0.99	0.96	**1.00**	0.99	0.97
5) GA-SVM	**0.98**	**0.98**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.99**
6) GA-PCA (Proposed)	**0.98**	0.97	**1.00**	0.99	**1.00**	**1.00**	**1.00**	**0.99**
7) Random Forest	0.73	0.70	0.98	0.5	0.53	0.65	0.5	0.66
8) XGBoost	0.70	0.74	**1.00**	0.50	0.95	0.94	0.99	0.84
9) AdaBoost	0.74	0.50	0.92	0.5	0.5	0.5	0.5	0.60
10) Gradient Boosting	0.72	0.71	**1.00**	0.93	0.95	0.65	0.52	0.79
11) Decision Tree	0.75	0.50	0.98	0.54	0.96	0.98	0.52	0.75
12) Variance (Filter)	0.71	0.91	0.95	0.81	0.85	0.94	0.99	0.88
13) Pearson Correlation (Filter)	0.5	0.55	0.54	0.50	0.54	0.86	0.60	0.59

TABLE 10. AVERAGE COUNT OF FEATURES FOR THE PROPOSED METHOD COMPARED TO OTHER FEATURE SELECTION METHODS IN DATASET 2.

Feature Selection Method	Fold1	Fold2	Fold3	Fold4	Fold5	Fold6	Fold7	Average
1) Binary PSO-SVM	17	16	17	14	11	13	16	14.85
2) Binary PSO-PCA	15	25	11	12	11	14	14	14.57
3) WOA-SVM	**2**	9	**1**	4	**1**	**1**	2	2.85
4) WOA-PCA	**2**	**3**	2	**3**	2	2	6	2.85
5) GA-SVM	5	6	3	**3**	2	**1**	**1**	3.00
6) GA-PCA (Proposed)	5	4	2	**3**	2	2	**1**	**2.71**
7) Random Forest	5	4	2	**3**	2	2	**1**	**2.71**
8) XGBoost	5	4	2	**3**	2	2	**1**	**2.71**
9) AdaBoost	5	4	2	**3**	2	2	**1**	**2.71**
10) Gradient Boosting	5	4	2	**3**	2	2	**1**	**2.71**
11) Decision Tree	5	4	2	**3**	2	2	**1**	**2.71**
12) Variance (Filter)	19	39	28	53	56	46	48	41.29
13) Pearson Correlation (Filter)	13	11	11	15	12	14	14	12.86

category with Dataset 2. The average of these results is calculated in the last column. Similarly, Table 10 presents the feature selection methods as rows and the selected features in each category with Dataset 2 as columns. The average of these features is calculated in the last column.

TABLE 11. STUDENT'S T-TEST AND P-VALUE OF DATASET 2.

Feature Selection	Average AUC	Average Feature	AUC		Total Time
			T-test	P-value	
1) Binary PSO-SVM	0.97±0.03	14.86±2.10	2.03	0.03	6293
2) Binary PSO-PCA	0.98±0.02	14.57±4.50	1.46	0.08	11109
3) WOA-SVM	0.96±0.03	2.86±2.70	2.27	0.02	1056
4) WOA-PCA	0.98±0.02	2.86±1.36	1.47	0.084	1306
5) GA-SVM	**0.99±0.01**	3.00±1.77	**-0.54**	**0.30**	2112
6) GA-PCA (Proposed)	0.99±0.01	**2.71±1.28**	—	—	3252
7) Random Forest	0.66±0.16	**2.71±1.28**	5.15	0.00011	0.17
8) XGBoost	0.84±0.18	**2.71±1.28**	2.23	0.02271	0.22
9) AdaBoost	0.60±0.16	**2.71±1.28**	6.19	0.00002	**0.16**
10) Gradient Boosting	0.79±0.17	**2.71±1.28**	3.07	0.00484	0.22
11) Decision Tree	0.75±0.21	**2.71±1.28**	2.84	0.00741	0.05
12) Variance (Filter)	0.88±0.09	41.29±12.53	3.02	0.00530	9.53
13) Pearson Correlation (Filter)	0.59±0.12	12.86±1.46	8.49	<0.00001	31.11

Student's t-test and P-value are indeed commonly used statistical tests to compare the performance of different models or methods in machine learning. The P-value helps assess the significance of the observed differences between two models, indicating the probability of obtaining the observed results by random chance alone. A smaller P-value suggests a more significant difference between the models.

On the other hand, Student's t-test evaluates whether the means of two sets of data are significantly different from each other. It provides a measure of the extent to which the observed differences between the models are likely due to actual differences in their performance rather than random variation. A smaller t-test value indicates a smaller difference between the means of the two models, implying greater similarity.

By applying these statistical tests to the results obtained from different feature selection methods, researchers can quantitatively assess the significance of the observed performance differences and make informed decisions about the effectiveness of each method.

How to calculate Student's t-test is stated in Equation (14), where S_1^2 and S_2^2 correspond to the variances of category one and two, respectively. And N_1 and N_2 represent the number of observations in categories 1 and 2, respectively, while $\overline{X}_1$ and $\overline{X}_2$ denote the means of categories 1 and 2, respectively.

In Tables 8 and 11, GA-PCA emerges with the highest AUC and maintains an acceptable number of selected features. Following closely is GA-SVM, notable for its minimal time expenditure and high P-value, indicating a strong similarity in results with GA-PCA.

TABLE 12. THE PERFORMANCE OF THE PROPOSED APPROACH ON DIFFERENT DATASETS.

ID	Dataset name	Feature		F1-Score		
		Base Line	Phases	Base Line	Phase1	Phase2
1	200+ Financial Indicators of US stocks (2014-2015)	221	4	0.44	0.49	0.66
2	COVID-19 - Clinical Data to assess diagnosis	227	20	0.93	0.91	0.95
3	Diagnosis of COVID-19 and its clinical spectrum	107	17	0.99	0.99	0.99
4	African Country Recession Dataset (2000 to 2017)	49	17	0.91	0.88	0.94

$$t = \frac{\overline{X}_1 - \overline{X}_2}{\sqrt{\left(\frac{(N_1-1)S_1^2 + (N_2-1)S_2^2}{N_1+N_2-2}\right)\left(\frac{1}{N_1}+\frac{1}{N_2}\right)}} \quad (14)$$

TABLE 13. PHASE2: NORMAL RANGE, DIRECTION, AND RANKING OF SELECTED FEATURES IN COVID-19 - CLINICAL DATA TO ASSESS DIAGNOSIS.

ID	Selected Features	Normal Range		Direction	Ranking
		Start	End		
1	DISEASE GROUPING 5	0.65	0.68	⇓	0.909%
2	HTN	0.35	0.72	⇓	–
3	OTHER	0.12	0.76	⇑	0.909%
4	ALBUMIN MIN	-0.08	0.04	⇑	0.909%
5	CALCIUM MEAN	0.38	0.80	⇑	1.818%
6	FFA MIN	-0.61	-0.17	⇓	–
7	HEMATOCRITE MEAN	0.00	1.00	⇓	4.545%
8	LACTATE MIN	-0.16	0.52	⇓	2.727%
9	NEUTROPHILES MEDIAN	-0.38	0.12	⇑	2.727%
10	P02 VENOUS MEDIAN	-0.16	-0.04	⇓	2.727%
11	PCR MIN	0.26	0.28	⇓	4.545%
12	PH VENOUS MIN	-0.42	0.00	⇓	1.818%
13	POTASSIUM MAX	-0.20	0.48	⇑	1.818%
14	TTPA MIN	-0.04	0.36	⇓	2.727%
15	UREA MAX	0.38	0.56	⇓	1.818%
16	TEMPERATURE MEAN	0.00	0.18	⇓	6.364%
17	RESPIRATORY RATE MEDIAN	-0.50	-0.42	⇑	10.909%
18	BLOODPRESSURE SISTOLIC MAX	-0.48	0.96	⇑	4.545%
19	RESPIRATORY RATE DIFF REL	-0.86	-0.38	⇑	10.909%
20	TEMPERATURE DIFF REL	-0.14	-0.07	⇑	6.364%

The outcomes of other algorithms, compared to GA-PCA, exhibit P-values below 0.05 in Table 8, and below 0.1 in Table 11. These lower P-values signify significant and discernible differences in performance compared to GA-PCA.

Furthermore, we conduct a comparative analysis between the results of GA-PCA and other algorithms using Student's T-test, providing additional insights into the significance of the disparities observed.

III. Explain of features by normal calculation of range, direction and ranking of features

In Table 12, we observe the impact of the proposed algorithm on feature reduction and performance across four datasets. Indeed, we have implemented our approach on various datasets and problems to demonstrate that it is well-generalized, rather than being effective only for a specific dataset or problem domain. The table presents the initial number of features in the first column and the final number of features after applying the algorithm in the second column. Subsequent columns display the F1-score values for the entire dataset, performance after feature selection, and performance after sample selection, respectively. The first row highlights a significant improvement in performance achieved by the proposed algorithm, particularly evident in the substantial increase in performance from the initial dataset value. Despite higher initial values for the remaining datasets, notable reductions in features are observed across all datasets, demonstrating the efficacy of the algorithm in feature reduction.

In Table 13, 14, 15, and 16, each row corresponds to a selected feature. The first column denotes the row number, while the second column lists the name of the feature. The third column displays the optimal normal range selected by the genetic algorithm for sample selection. As the data is normalized, these ranges are represented in decimal form. To streamline the computational process and reduce complexity, only a percentage of feature values are selected for decoding in the objective function, with other values selected relative to them. The actual values are normalized using MinMaxScaler, ensuring they fall within the interval specified by the beginning and end of the range.

If a feature exhibits a characteristic direction, it is indicated in the third column, denoted by an up or down

arrow to signify a high-risk direction. The fourth column represents the ranking of the selected features, reflecting their importance. The rank of each feature is determined by calculating the frequency of occurrence of each feature, excluding min, max, median, mean, diff, and diff rel, relative to the total frequency of features.

TABLE 14. PHASE2: NORMAL RANGE OF SELECTED FEATURES IN DIAGNOSIS OF COVID-19 AND ITS CLINICAL SPECTRUM.

ID	Selected Features	Normal Range	
		Start	End
1	Leukocytes	1.60	6.40
2	Basophils	26.38	31.42
3	Monocytes	-4.38	13.94
4	Rhinovirus/Enterovirus	-0.67	1.46
5	Chlamydophila pneumoniae	-2.61	0.43
6	Parainfluenza 4	-1.98	-1.01
7	Inf A H1N1 2009	-0.14	0.58
8	Proteina C reativa mg/dL	14.57	15.61
9	Potassium	-3.99	5.21
10	Sodium	-11.93	6.89
11	Influenza B, rapid test	0.01	4.25
12	Aspartate transaminase	14.43	25.12
13	Urine - Density	-0.75	6.34
14	Urine - Leukocytes	-2.14	-1.09
15	Urine - Granular cylinders	-6.01	-0.95
16	Lactic Dehydrogenase	-6.20	2.79
17	pO2 (arterial blood gas analysis)	19.04	26.20

In Table 12, the results are presented using a single metric. However, in Figure 15 and 16, the performance results are displayed across multiple metrics. The first column from the right represents the AUC metric, the middle column displays the F1-score metric, and the left column showcases the Accuracy metric.

TABLE 16. PHASE2: NORMAL RANGE OF SELECTED FEATURES IN 200+ FINANCIAL INDICATORS OF US STOCKS (2014-2015).

ID	Selected Features	Normal Range	
		Start	End
1	Revenue	0.86	2.75
2	Revenue Growth	12.96	31.87
3	Cost of Revenue	-16.55	26.05
4	Gross Profit	-20.65	-16.72

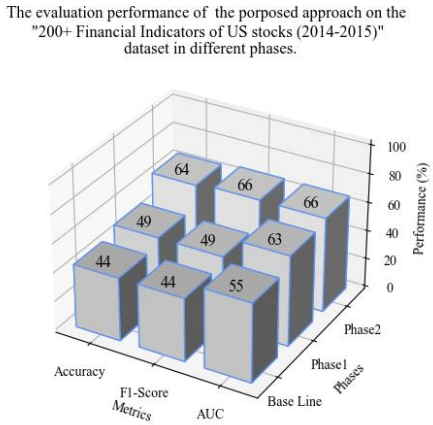


FIGURE 15. THE EVALUATION PERFORMANCE OF THE PROPOSED APPROACH ON THE "200+ FINANCIAL INDICATORS OF US STOCKS (2014-2015)" DATASET IN DIFFERENT PHASES.

The bottom row represents the initial values or the class with all the data, the middle row corresponds to phase 1 or after feature selection, and the top row depicts phase 2 or after sample selection.

In Fig. 15, the data pertains to the dataset of 200+ Financial Indicators of US stocks (2014-2015). All metrics demonstrate improvement across all stages, indicating progress. Conversely, in Fig. 16, the data is related to the COVID-19 - Clinical Data to assess diagnosis dataset. While there is a slight reduction after feature selection across the metrics, selecting the sample not only compensates for this decline but also showcases progress.

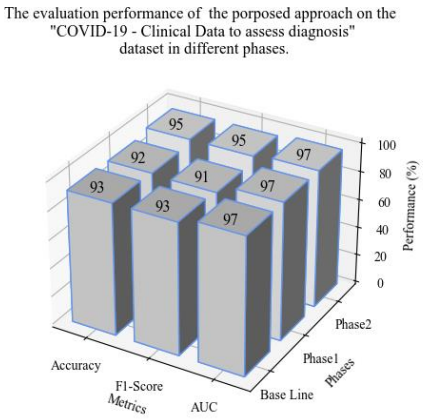


FIGURE 16. THE EVALUATION PERFORMANCE OF THE PORPOSED APPROACH ON THE "COVID-19 - CLINICAL DATA TO ASSESS DIAGNOSIS" DATASET IN DIFFERENT PHASES.

TABLE 15. PHASE2: NORMAL RANGE AND DIRECTION OF SELECTED FEATURES IN AFRICAN COUNTRY RECESSION DATASET (2000 TO 2017).

ID	Selected Features	Normal Range		Direction
		Start	End	
1	pop	37.13	101.67	⇓
2	emp	6.73	63.21	⇓
3	emp_to_pop_ratio	0.26	0.55	⇑
4	hc	1.85	2.52	⇑
5	ccon	425958.66	516639.53	⇓
6	cda	20856.78	726816.93	⇓
7	cn	409063.39	2598259.61	⇓
8	ck	0.02	0.02	⇓
9	ctfp	0.58	0.94	⇓
10	cwtfp	0.14	0.60	⇓
11	rconna	671006.10	678599.50	⇓
12	rdana	546327.73	573495.20	⇓
13	rnna	267127.36	1223063.84	⇓
1	rkna	0.67	2.60	⇑
15	rtfpna	0.77	1.46	⇓
16	rwtfpna	0.74	0.98	⇓
17	labsh	0.32	0.35	⇑
18	irr	0.02	0.41	⇓
19	delta	0.07	0.09	⇓

DISCUSSION AND CONCLUSION

This research aims to achieve the following objectives in Dataset 1:

- Increase the accuracy in flattening the demand curve for intensive care beds with 97% AUC score. Although a high AUC was achieved in the first phase, the model's greater focus on saving lives makes it even more valuable. Additionally, while new datasets may not yield the desired high results in the first phase, our method has the potential to achieve the expected outcomes in the second phase.
- Selection of 20 influential features as listed in Table 13.
- Determination of the normal range for each feature as shown in Table 13.
- Selection of the direction for each feature, as explained in Table 13.
- Determination of the rank and importance of each feature, as specified in Table 13.

This paper delves into the analysis of medical data and blood tests, aiming to elucidate key characteristics. Unlike traditional blood tests where each parameter has an independent target class (e.g., blood sugar for diabetes), Dataset 1 combines all characteristics to predict admission to the ICU for confirmed COVID-19 patients. Given the collective impact of these features on the target class, meticulous feature selection in phase one was imperative. GA-PCA emerged as the optimal feature selection algorithm, generating selected features for phase two. This phase aimed to eliminate noisy and ineffective features to mitigate their adverse effects on the normal range. Sampling with a normal range helped address feature imbalance issues.

Furthermore, feature direction was explored to understand the collective movement of features towards changing the target class. For instance, in Dataset 1, each feature's direction indicates the patient's risk level and their need for ICU care. Additionally, features were ranked within the evolutionary algorithm based on their frequency across batch-implemented data, expressed as a percentage relative to the total feature frequency. This comprehensive approach facilitates a deeper understanding of feature dynamics and their impact on the target class.

FUTURE WORK

Determining the normal range of a feature holds diverse applications across science and industry. One notable application lies within the automotive and transportation sector. Our future endeavors could involve establishing the normal load limit for heavy transport vehicles. This entails understanding the optimal load capacity or range that a vehicle can safely transport while minimizing wear and tear or the likelihood of mechanical failures.

Such analysis could involve monitoring various parameters, including sensor data embedded within the vehicle, which is recorded over time intervals. These parameters could include factors such as engine performance, temperature, pressure, and more. By categorizing samples into healthy and damaged classes, predictive models can be developed to anticipate potential failures.

For instance, if sensor data indicates deviations from the normal range, signaling potential failure or wear in a particular vehicle component, proactive maintenance can be performed. This predictive maintenance approach helps replace or repair faulty parts before they lead to accidents or breakdowns, thus enhancing safety and operational efficiency on the road.

REFERENCES

[1] Kaggle, "Covid-19 - clinical data to assess diagnosis," 2024.

[2] Kaggle, "Diagnosis of covid-19 and its clinical spectrum," 2024.

[3] Kaggle, "200+ financial indicators of us stocks (2014-2018)," 2024.

[4] Kaggle, "African country recession dataset (2000 to 2017)," 2024.

[5] M. Tajgardan, R. Khoshkangini, and P. Mashhadi, "Steered metaheuristic optimization approach for vehicle breakdown prediction," in Proceedings of the 2024 8th International Conference on Information System and Data Mining, ICISDM '24, (New York, NY, USA), p. 128–135, Association for Computing Machinery, 2024 , doi: 10.1145/3686397.3686422 .

[6] A. Hamdipour, A. Basiri, and M. Zaare, "Numerical solution of nonlinear equations systems with improved meta-heuristic aro algorithm," Soft Computing Journal, pp. –, 2024 , doi: 10.22052/scj.2025.254888.1244.

[7] M. Ramezani, S. Kalantari, and A. Madadi, "Parameter estimation of a linear regression model by extending least square method based on quantum computing," Soft Computing Journal, pp. –, 2024 , doi: 10.22052/scj.2024.253276.1169.

[8] M. Ghaffar Alishahi, E. Pira, and A. Rouhi, "Development of city councils evolution algorithm for multi-objective optimization problems," Soft Computing Journal, pp. –, 2023 , doi: 10.22052/scj.2024.253070.1154.

[9] M. Zolfaghari, H. Fadishei, M. Tajgardan, and R. Khoshkangini, "Stock market prediction using multi-objective optimization," in 2022 12th International Conference on Computer and Knowledge Engineering (ICCKE), pp. 253–262, IEEE, 2022 , doi: 10.1109/ICCKE57176.2022.9960002.

[10] M. Tajgardan, R. Khoshkangini, A. Orand, and M. Rabbani, "A fault forecasting approach using two-dimensional optimization," in 2024 IEEE 10th World Forum on Internet of Things (WF-IoT), pp. 561–567, 2024 , doi: 10.1109/WF-IoT62078.2024.10811220.

[11] N. Behar and M. Shrivastava, "Pixel-level feature extraction model for breast cancer detection.," Computers, Materials & Continua, vol. 74, no. 2, 2023 , doi: 10.32604/cmc.2023.031949.

[12] C. Yang, H. Jia, L. Dong, H. Zhao, and M. Zhao, "Selection of landsat-8 operational land imager (oli) optimal band combinations for mapping alteration zones," Remote Sensing, vol. 16, no. 2, p. 392, 2024 , doi: 10.3390/rs16020392.

[13] F. Zhang, M. Wang, F. Zhang, Y. Xiong, X. Wang, S. Ali, Y. Zhang, and S. Fu, "Hyperspectral imaging combined with ga-svm for maize variety identification," Food Science & Nutrition, vol. 12, no. 5, pp. 3177–3187, 2024 , doi: 10.1002/fsn3.3984.

[14] Y. Qiu, R. Li, and X. Zhang, "Simultaneous svm parameters and feature selection optimization based on improved slime mould algorithm," IEEE Access, vol. 12, pp. 18215–18236, 2024 , doi: 10.1109/ACCESS.2024.3351943.

[15] B. Shah, A. Gupta, and S. Paul, "A pso-svm-based change detection algorithm for remote sensing optical images," IEEE Access, 2024 , doi: 10.1109/ACCESS.2024.3387940.

[16] I. B. Sya'idah, S. Surono, and G. K. Wen, "Dynamicweighted particle swarm optimization-support vector machine optimization in recursive feature elimination feature selection," MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer, vol. 23, no. 3, pp. 627– 640, 2024.

[17] G. Atteia, R. Alnashwan, and M. Hassan, "Hybrid feature-learning-based pso-pca feature engineering approach for blood cancer classification," Diagnostics, vol. 13, no. 16, p. 2672, 2023 , doi: 10.3390/diagnostics13162672.

[18] W. Ahmed, A. Ali, F. Lodhi, W. Ahmed, and N. Baloch, "Convolutional neural networks with particle swarm optimization: A reliable method for sars-cov-2 detection in x-ray images," None, 2023.

[19] D. C. Dennett and D. C. Dennett, Darwin's Dangerous Idea: Evolution and the Meanins of Life. Simon and Schuster, 1996 , doi: 10.1017/S0012217300017169.

[20] M. Srinivas and L. M. Patnaik, "Genetic algorithms: A survey," computer, vol. 27, no. 6, pp. 17–26, 1994 , doi: 10.1109/2.294849.

[21] D. Whitley and A. M. Sutton, "Genetic algorithms-a survey of models and methods," in Handbook of Natural Computing, pp. 637–671, Springer Berlin Heidelberg, 2012 , doi: 10.1007/978-3-540-92910-9_21.

[22] H. Motieghader, A. Najafi, B. Sadeghi, and A. Masoudi-Nejad, "A hybrid gene selection algorithm for microarray cancer classification using genetic algorithm and learning automata," Informatics in Medicine Unlocked, vol. 9, pp. 246–254, 2017.

[23] C. Ma, W. Hao, F. Pan, and W. Xiang, "Road screening and distribution route multi-objective robust optimization for hazardous materials based on neural network and genetic algorithm," PLoS One, 2018 , doi: 10.1371/journal.pone.0198931.

[24] M. Dorigo and C. Blum, "Ant colony optimization theory: A survey," Theoretical computer science, vol. 344, no. 2-3, pp. 243–278, 2005 , doi: 10.1016/j.tcs.2005.05.020.

[25] R. Khoshkangini, S. Zaboli, and M. Conti, "Efficient routing protocol via ant colony optimization (aco) and breadth first search (bfs)," in Proceedings of the 2014 IEEE international conference on internet of things (iThings), and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom), Taipei, Taiwan, pp. 1–3, 2014 , doi: 10.5555/2762758.2762988.

[26] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," Multimedia Tools and Applications, pp. 1–36, 2020 , doi: 10.1109/ACCESS.2018.2815992.

[27] L. Deng, J. Pei, J. Ma, and D. L. Lee, "A rank sum test method for informative gene discovery," in Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04, (New York, NY, USA), p. 410–419, Association for Computing Machinery, 2004 , doi: 10.1145/1014052.1014099.

[28] S. Mirjalili and A. Lewis, “The whale optimization algorithm,” Advances in engineering software, vol. 95, pp. 51–67, 2016 , doi: 10.1016/j.advengsoft.2016.01.008.

[29] Z. Tao, L. Huiling, W. Wenwen, and Y. Xia, “Ga-svm based feature se lection and parameter optimization in hospitalization expense modeling,” Applied soft computing, vol. 75, pp. 323–332, 2019 , doi: 10.1016/j.asoc.2018.11.001.

About the Authors

Mohsen Tajgardan holds a Master's degree in Computer Engineering from Qom University of Technology. His research interests include neural networks and deep learning.
You can contact him at the following email address:
tajgardan.m@qut.ac.ir
orcid.org/0000-0003-1220-5196

Mahboobeh Shamsi is currently an associate professor at Qom University of Technology and has experience supervising Master's theses. Her research interests include the Internet of Things, big data, and image processing.
You can contact her at the following email address: shamsi@qut.ac.ir

Reza Khoshkangini Senior lecturer
Department of Computer Science and Media Technology
Malmo University, Sweden
You can contact him at the following email address:
reza.khoshkangini@mau.se
+46 40 665 82 73
orcid.org/0000-0002-3797-4605

Abdolreza Rasouli Kenari is currently an assistant professor of Computer Engineering at Qom University of Technology. His research interests include big data and data mining.
You can contact him at the following email address:
rasouli@qut.ac.ir
orcid.org/0000-0003-4817-9380