IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 29 May 2024, accepted 29 June 2024, date of publication 5 July 2024, date of current version 11 September 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3424231*

RESEARCH ARTICLE

# High Accuracy COVID-19 Prediction Using Optimized Union Ensemble Feature Selection Approach

**ABBAS JAFAR AND MYUNGHO LEE, (Member, IEEE)**
Department of Computer Engineering, Myongji University, Yongin 17058, South Korea

Corresponding author: Myungho Lee (myunghol@mju.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the government (MSIT) (RS-2023-00321688). ※ MSIT: Ministry of Science and ICT (Information and Communication Technology).

**ABSTRACT** Recently, the world has been dealing with a severe outbreak of COVID-19. The rapid transmission of the virus causes mild to severe cases of cough, fever, body aches, organ failures, and death. An increasing number of patients, fewer diagnostic options, and extended waiting periods for test results all put pressure on healthcare systems, increasing the virus's spread. A concise and accurate automatic diagnosis is crucial to identify infected patients in the early stage. This paper proposes a machine learning-based predictive framework to identify COVID-19 cases from clinical data using an optimized union ensemble feature selection (OUEFS) approach. The OUEFS is based on the union ensemble of the feature subsets obtained through a rigorous feature selection (FS) process. It also involves a performance optimization of the ML classifiers. Initially the OUEFS identified key features from the publicly accessible COVID-19 dataset using FS methods such as Mutual Information Feature Selection (MIFS), Recursive Feature Elimination (RFE), and the RidgeCV. The most important features were selected using Top-k thresholding technique. Then selected subsets of features were integrated using a union ensemble approach where an optimal combination of features with enhanced predictive power is derived. This composite feature set was subsequently utilized for model training and evaluation. The classification was conducted using machine learning algorithms such as linear SVM, gradient boosting (GB), logistic regression (LR), and Adaboost to compare their effectiveness on individual and combined feature subsets. We also conducted a Genetic Algorithm (GA) based hyperparameter optimization (HPO) which further refined our training process and enhanced the accuracy of our proposed approach. Experimental results show that the union ensemble of MIFS and RidgeCV FS techniques and the Adaboost classifier and GA HPO achieved 96.30% accuracy. Our optimized union ensemble approach demonstrated superior performance over previous ensemble-based approaches to predict COVID-19 disease, thus offering a robust tool for early and efficient diagnosis without requiring hospital visits.

**INDEX TERMS** Machine learning, feature selection, COVID-19 classification, ensemble learning, hyperparameter optimization.

## I. INTRODUCTION

In late 2019, a severe acute upper respiratory disease named coronavirus (COVID-19) emerged and spread rapidly worldwide quickly. In March 2020, the World Health Organization (WHO) declared COVID-19 a pandemic disease [1]. As of March 21, 2023, approximately 761 million people have been affected globally, with around 7 million deaths reported [2]. The most common clinical symptoms of COVID-19 include cough, tiredness, fever, headache, sore throat, shortness of breath, and chest pain [3]. The primary cause of the virus's rapid spread is through contaminated air with coronavirus

The associate editor coordinating the review of this manuscript and approving it for publication was Yongming Li.

droplets exhaled by COVID-19 patients [4]. In mild cases, patients with other diseases can be easily affected, which increases the mortality rate. Stopping the spread of the virus and controlling its mortality rate is a vital medical challenge.

At present, diverse COVID-19 detection methodologies are under scrutiny, including blood tests, imaging modalities (such as X-ray and CT scans), and Polymerase Chain Reaction (PCR) tests [5], [6]. Although blood tests and imaging studies have been successful, they take a long time to generate results. They also have high resource requirements. Thus, they need to be more comprehensive in addressing the problem's urgency. PCR, a widely employed diagnostic method, has limitations such as a scarcity of PCR laboratories, insufficient PCR kit supplies [6], and prolonged result waiting times [7]. To tackle these challenges, the prioritized identification of infected individuals emerges as extremely important. In response, a concise and accurate automatic diagnosis method is crucial for healthcare practitioners in the early stages of identifying infected patients.

Recently, automated studies have been conducted to identify COVID-19 disease [8], [9], [10], [11], [12]. These studies classify the disease with a high prediction score but have limitations. The majority of them use datasets with fewer features and smaller sizes. Smaller training sets are more likely to overfit and have non-Gaussian noise, lessening the generalizability of ML models [13], [14]. In [18] and [19], individual feature selection techniques were used to extract valuable features, however, they needed to create a useful framework that considered the simultaneous optimization of the data and the model. Recently, ensemble feature selection (EFS) approaches have been developed, inspired by ensemble learning techniques in ML [15]. EFS approaches outperform the individual FS techniques [13], [15], [16], [17], [18]. Thus, they generate a diversified feature subset that can yield an optimal combination of features.

In this paper, we developed an optimized union ensemble feature selection (OUEFS) approach for enhancing the prediction accuracy of COVID-19. The OUEFS is based on the union ensemble of the feature subsets obtained through a rigorous feature selection process. It also involves a performance optimization of the ML classifiers. Our approach began with the precise data preprocessing for the raw COVID-19 clinical symptoms dataset including addressing missing values, data imbalance, and normalization. For the preprocessed data, feature selection strategies such as MIFS, RFE, and RidgeCV were applied to identify key features. The most important feature subset was selected using Top-k thresholding technique. These selected feature subsets are combined through various union combinations to create an optimal ensemble feature subset. Then, we conducted classification using four ML classifiers: LSVM, LR, GB, and Adaboost. Our goal at the classification stage is to predict whether a patient has COVID-19 (yes or no) and identify the best-performing classifier based on the ensemble feature subset. To optimize the selected classifier's performance further, we used a GA to fine-tune the hyperparameters. Finally, we evaluate the model's performance using accuracy, precision, recall, F1-score, and AUC metrics. We name this entire process as the Optimized Union Ensemble Feature Selection (OUEFS) as illustrated in Fig. 1. Experimental results showed that the union ensemble of MIFS and RidgeCV FS techniques, together with the Adaboost classifier and GA-HPO, achieved 96.30% accuracy. This accuracy outperforms all previous ensemble-based methods for COVID-19 prediction. Our paper is the first medical disease detection study using the union of EFS methodology with the meta-heuristic optimization algorithm. Our OUEFS approach has the potential for rapid identification of the virus at an early stage, thus applicable to the healthcare system during a pandemic.

The rest of this paper is organized as follows: Section II summarizes the previous research on the COVID-19 disease diagnosis. Section III provides the background information for developing our proposed OUEFS approach. Section IV presents the OUEFS system with extensive explanations. Section V conducts experiments and offers in-depth analyses of the results. Section VI concludes the paper.

## II. PREVIOUS RESEARCH

A number of machine learning-based solutions have been recently proposed for predicting and detecting COVID-19 patients. We first review previous research using ensemble approaches:

- Kumar [19] developed a COVID-19 prediction model for the mortality risk of coronavirus patients based on their symptoms. A dataset of 75,000 cases with 10 features was collected from the Kaggle public source. Different ML classifiers were used, such as Naïve Bayes (NB), RF, and SVM. They used an ensemble feature selection method to identify the key features and enhance the performance. Bagging and boosting ensemble learning methods were utilized to predict COVID-19 cases accurately.
- Koushik et al. [20] proposed a supervised ML approach to predict COVID-19 infection. The dataset was extracted from the Israel Ministry of Health, consisting of 112,345 samples, with 102,233 COVID-19 negative cases and 10,112 positive cases. They predicted the disease using LR, RF, and KNN classifiers. Furthermore, they applied the MaxVoting ensemble approach to the ML classifiers to boost the final prediction.
- Debjit et al. [17] proposed an optimized approach using the Harris Hawks optimization (HHO) algorithm to detect COVID-19. ML algorithms such as XGB, LightGB, categorical boosting, RF, and SVM were optimized and applied to the publicly available COVID-19 dataset of 1,023,426 samples with a 1.20% positive ratio. They extracted the essential features and calculated their feature importance score using SHAP values. Moreover, ensemble learning combined various optimized ML classifiers to predict the COVID-19 positive cases. The majority voting ensemble approach was used to determine the class label and

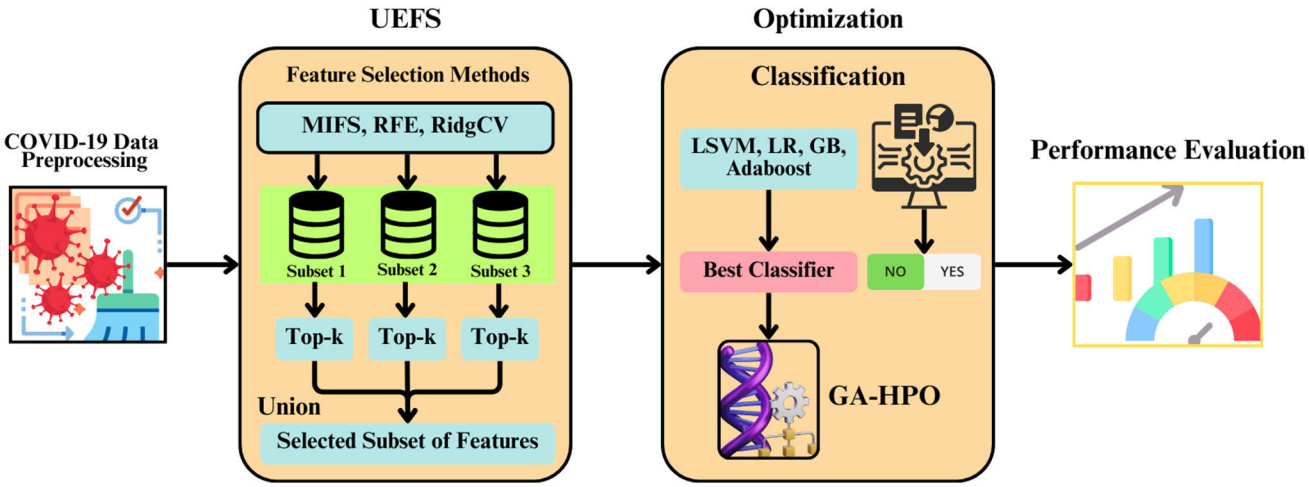


**FIGURE 1.** Overview of proposed OUEFS approach to predict COVID-19 patients.

enhance the final prediction by leveraging multiple classifiers.

There have also been other approaches to predict and detect COVID-19 patients:

- Aljameel et al. [21] proposed an early prediction methodology for coronavirus to increase the survival rate. The dataset, containing 287 COVID-19 patient samples with a 100% positive ratio, was collected from the KFU Hospital in Saudi Arabia and analyzed using LR, RF, and XGB algorithms.
- Awal et al. [24] developed a framework to predict COVID-19 patients using inpatient facility data. Multiple algorithms were used, such as RF, DT, NB, GBC, XGB, and KNN. They extracted the relevant features from the dataset consisting of 11,169 samples with a 2.82% positive ratio using shapely additive explanations analysis.
- Pourhomayoun and Shakibi [22] proposed a predictive system to determine the health and mortality risk of COVID-19 patients. The authors utilized documented data from 2,670,000 patients (all positive cases) worldwide with laboratory-confirmed cases. Various classifiers such as SVM, ANN, RF, DT, LR, and KNN were used. They also employed univariate and multivariate filter and wrapper methods to extract the key features.
- Danacı and Tuncer [23] utilized multiple FS methods to select the most relevant features from COVID-19 data. They used a dataset of 221 patients, including 121 positive and 100 negative patients. They applied various algorithms, including SVM, KNN, DT, and ANN, to predict the disease.

Table 1 summarizes the previous research explained above.

## III. BACKGROUND

This section provides the background information used to develop the OUEFS system. It includes a detailed discussion of the FS approaches, ML classifiers, the GA optimization approach, and performance evaluation metrics. These components ensure the advancement and effectiveness of the automatic diagnosis system.

### A. INDIVIDUAL FEATURE SELECTION APPROACHES

Feature selection (FS) is a pivotal data processing element that extracts key features. It is a process to minimize computing costs with its capability to prevent overfitting and preserve the models' predictive abilities [25]. There are three distinct types of FS strategies: filter, wrapper, and embedded. The choice of the FS method relies on the problem being solved.

Filter feature selection (FFS) methods rank features according to their unique characteristics. They are more computationally efficient than wrapper methods to minimize the error rate [26]. The filter method has one major drawback of suboptimal performance; it may retain redundant feature in its set. Standard filter methods are Fisher's score, information gain, and mutual information [9], [27]. Wrapper feature selection (WFS) methods are more effective than filter methods in selecting the most relevant features [28]. However, they can be computationally expensive and more prone to overfitting because the search strategy in the wrapper evaluates each candidate feature set. Well-known wrapper methods include forward, backward, exhaustive search selection, and recursive feature elimination [9], [27]. Embedded feature selection (EFS) methods integrate FS into the learning algorithm and perform FS as a separate step [29]. They are typically more efficient than filter and wrapper methods because they do not require the repeated execution of the learning algorithm. Tree-based and RidgeCV algorithms are examples of embedded methods [9], [27].

FS aims to improve classification performance by eliminating irrelevant and redundant features. However, there are also limitations. First, many methods must account for redundancy among selected features, potentially retaining correlated features that provide little additional information.

**TABLE 1. Summary of previous research to predict COVID-19 disease.**

| Year | Authors | Objective | ML Methods | Ensemble approach |
|---|---|---|---|---|
| 2021 | Koushal Kumar [19] | Predicting the COVID-19 mortality risk using ensemble feature selection approach | NB, RF, SVM | Yes |
| 2021 | Koushik et al. [20] | Identification of COVID-19 infection using epidemiology labeled dataset for positive and negative COVID-19 cases in Mexico | LR, DT, SVM, NB, ANN | Yes |
| 2022 | Debjit et al. [17] | Build an optimized ML model to predict COVID-19 at an early stage using big data | XGB, LGB, CB, RF, SVM | Yes |
| 2021 | Aljameel et al. [21] | Evaluation and prediction of COVID-19 patients based on their characteristics monitored at home | LR, RF, XGB | No |
| 2020 | Pourhomayoun et al. [22] | Build a predictive model for COVID-19 to determine the health risk and mortality risk | SVM, ANN, RF, DT, LR | No |
| 2022 | Danaci et al. [23] | Diagnose COVID-19 cases using multiple feature selection approaches from biochemical parameters data | SVM, KNN, DT, ANN | No |
| 2021 | Awal et al. [27] | Quick optimized method to evaluate COVID-19 patients using inpatient facility data | RF, DT, NB, XGB, KNN | No |

Second, individual filter-based approaches can introduce a bias towards a specific subset, overlooking potentially valuable features. Finally, inconsistent prediction accuracy can occur during classification depending on the chosen feature subset. Different algorithms can select varying sets of features from the same data. This inconsistency necessitates further exploration of integration-based methods that prioritize diversity and accuracy in the chosen feature set. Table 2 summarizes the kinds of FS strategies, methods, and their advantages and limitations.

**TABLE 2. FS strategies with their advantages and limitations.**

| FS Strategies | FS Methods | Advantages | Limitations |
|---|---|---|---|
| Filter | •Fisher's Score<br>•Information gain<br>•Chi-Square Test<br>•Mutual Information | •Computationally efficient<br>•Model-independent<br>•Easy to interpret | •Ignores feature interactions<br>•Suboptimal performance |
| Wrapper | •Forward FS<br>•Backward FS<br>•Exhaustive FS<br>•Recursive Feature Elimination | •Model-specific<br>•Capture feature interactions<br>•Interaction with classifier | •Computationally expensive<br>•Risk of overfitting<br>•Model-dependent |
| Embedded | •Regularization L1, L2<br>•Random Forest importance<br>•RidgeCV | •Balances efficiency and performance<br>•Capture feature interactions | •Model-dependent<br>•Less interpretable |

To address these issues with FS methods, integrated FS approach was developed to select multiple FS methods to achieve high accuracy. Table 2 shows that each FS strategy has several specific feature selection methods, amongst which we select mutual information (MI), recursive feature elimination (RFE), and RidgeCV techniques to extract the essential features. MI was chosen for filter-based selection because it handles high-dimensional data well. It identifies informative features by measuring their dependency on the target variable [29], [30], [31]. RFE was selected for wrapper-based selection because it successfully finds relevant features through a recursive process of FS and model building [32], [33]. RidgeCV was chosen for embedded selection because it incorporates FS within model training, reduces overfitting by adding a penalty against complexity, and enhances generalizability [34]. Combining these methods, we aim to create an optimal feature subset that improves computational efficiency and predictive accuracy.

### 1) MUTUAL INFORMATION FEATURE SELECTION

François et al. [35] first proposed the idea of mutual information (MI) as a metric for measuring the degree to which two variables are interdependent. Mutual information measures how much one variable's uncertainty is reduced when knowledge of the other variable is obtained. This mathematical measurement can be expresses as Equation 1 below:

$$I(X,M) = H(X) - H(X|M) = \sum_{\substack{x_i \in X \\ M_j \in Y}} P(x_i, m_j) log \frac{P(x_i, m_j)}{P(x_i) \times P(m_j)} \quad (1)$$

Here, $I(X,M)$ is the MI between feature subset $X$ and class $M$, $H(X)$ is the entropy of $X$ subset, and $H(X|M)$ is the conditional entropy of $X$ subset given $M$ class. Furthermore, $P(x_i, m_j)$ is the joint probability with a $x_i$ value and $m_j$ class, whereas $P(x_i)$ represents the probability of a feature having $ax_i$ value, and $P(m_j)$ is the probability of a class being $m_j$.

### 2) RECURSIVE FEATURE ELIMINATION

Recursive feature elimination (RFE) is a proper FS method for eliminating irrelevant features from the input feature set and finding the important features that differentiate between classes. This process aims to reduce the feature set's complexity while maintaining high precision.

The RFE technique leverages the capabilities of Random Forest (RF) classifiers to perform an iterative evaluation of the variable significance, requiring the execution of multiple classification iterations. The iterative process consists of several key stages: the generation of a novel RF classifier, the assessment using cross-validation techniques, the analysis of feature importance metrics, and the modification of the feature set for subsequent iterations. Every feature in the subset is used in the initial classification round. The worst-performing features are then identified and deleted from the feature set, preparing for the next steps. Additionally, RFE repeats this process to reduce the possibility of dependencies and convergence among the input features.

#### 3) RIDGECV

Ridge regression is a type of regression that can be used to address multicollinearity-induced variation. Multicollinearity occurs when two or more independent variables are significantly correlated. This can cause inflated standard errors and inaccurate estimations of the regression coefficients [28]. To solve this problem, ridge regression penalizes the squared sum of the coefficients. Significant coefficients are punished by this penalty, which helps lower the estimates' variance. In other words, ridge regression decreases the coefficients to zero, which can help increase the stability of the estimates [29]. Equation (2) below shows the ridge regression's objective function:

$$argmin_w = (kXw - yk)^2 + (kw - kw)^2 \tag{2}$$

where $\alpha$ is a positive fixed constant, the coefficients' shrinkage can be adjusted by changing the $\alpha$ value [36].

### B. MACHINE LEARNING CLASSIFIERS

Machine learning classifiers are supervised learning algorithms that can predict the outcome of a data point based on its features. In this paper, we use classifiers to predict the COVID-19 disease. The selected classifiers include LSVM, LR, GB, and Adaboost.

LSVM addresses both classification and regression tasks. It can efficiently solve binary classification tasks, wherein the primary goal is to categorize data points into two distinct classes based on their exhibited characteristics. Key parameters for the LSVM during the training are C, regularization, kernel, hinge_loss, gamma, class weight, and max_iteration.

LR is a probabilistic statistical model to tackle classification tasks. The logistic function is a sigmoid function that estimates probabilities. It performs exceptionally well when the dataset is dimensional and overcomes the overfitting. It performs well on binary output and directly connects the dependent and independent variables using the sigmoid logistic regression. The parameters which make it more efficient are penalty, C, fit_intercept, class_weight, and max_iterations.

GB has gained popularity for its exceptional performance in tackling complex and challenging problems. The technique runs sequentially, improving models iteratively by reducing errors. It aims to minimize the overall loss by utilizing negative gradients. With its characteristics, this classifier handles the missing values in data effectively. The controlling parameters of GB classifiers are n_estimator, learning_rate, max_depth, loss, and max_feature.

Adaptive boosting (Adaboost) is a robust ensemble learning algorithm. It enhances the performance of low-performing classifiers by leveraging the knowledge gained from its errors, which is based on the boosting method. Each tree in the Adaboost depends on the last tree's error. The Adaboost utilizes a sequential ensembling and potentially induces overfitting. Key parameters of Adaboost during the training are n_estimator, n_jobs, learning_rate, base_estimator_param, and verbose.

### C. GA-BASED HYPERPARAMETER OPTIMIZATION

Genetic algorithms (GA) are meta-heuristic optimization techniques inspired by principles of natural selection, specifically tailored to identify high-quality solutions to complex optimization problems [37]. GA has demonstrated its efficacy in addressing a wide range of research challenges. It has played a crucial role in addressing complex challenges, including feature selection, optimization, hyperparameter tuning, neural network searches, clustering, classification, and anomaly detection problems. In this paper, we use GA to optimize the hyperparameters of ML classifiers to improve the overall performance.

The algorithms initiate their exploration by establishing an initial population of solutions randomly generated within the search space. The population under consideration comprises various solutions, each represented by a chromosome. These chromosomes serve as genetic instructions, encoding essential characteristics of the respective solutions. GA utilizes three bio-inspired operators: selection, crossover, and mutation [46]. The process for selection involves the intentional selection of a subset from a population, often showing a preference for individuals with superior fitness levels. Crossover, an essential genetic operator, allows the recombination of chromosomes from two-parent solutions, creating an offspring that inherits characteristics from both progenitors. Mutation serves to introduce variability by modifying one or more values within a chromosome. This diversity helps the program explore different search space populations [38].

GA is robust and effective across various optimization problems, driven by two main advantages. First, GA navigates search spaces using independent individuals, enabling parallel processing and minimizing the risk of getting stuck in local optima encountered by other optimization methods. Second, its implementation is straightforward and adaptable.

### D. PERFORMANCE EVALUATION METRICS

In machine learning, several performance evaluation metrics can be used to assess the performance and effectiveness of a model. This work calculates the evaluation metrics such

as accuracy, precision, recall, F-score, and AUC-ROC. The considered metrics are dependent on the four factors: true positives (*TP*), true negatives (*TN*), false positives (*FP*), and false negatives (*FN*). The accuracy of the ML classification model is obtained by calculating the proportion of correctly classified samples using a testing set. The precision measures the correctness, which is obtained by calculating the proportion of positive samples that are correctly classified. The recall is obtained by calculating the proportion of positive samples classified as positive. The F-score is obtained by calculating the weighted harmonic mean of precision and recall. AUC-ROC metric integrates the true positive rate (TPR) with the false positive rate (FPR) over the entire range of thresholds. The formulas of metrics are shown below:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$
$$Precission = \frac{TP}{TP + FP}$$
$$Recall = \frac{TP}{TP + FN}$$
$$F1 - score = 2(\frac{P * R}{P + R})$$
$$\text{AUC} = \int(\text{TPR})\, \text{d}(\text{FPR})$$

## IV. METHODOLOGY

### A. OVERVIEW

This section presents the methodology of our proposed OUEFS approach to predict COVID-19. Our approach comprises five key stages: data preprocessing, FS, union ensemble, classification, and HPO. Initially, a clinical symptoms-based COVID-19 dataset was selected and preprocessed. The data preprocessing stage involved cleaning the COVID-19 clinical dataset, normalizing it, and balancing it to mitigate the effects of class imbalance. This preparation is crucial for ensuring the data is suitable for effective feature selection and model training. Then, we employed three distinct MIFS, RFE, and RidgeCV FS methods to select the most relevant features [39] using the Top-k feature thresholding approach. The selected features from each FS method were then integrated using the Union Ensemble Feature Selection (UEFS) approach. This integration is critical as it amalgamates the strengths of individual FS techniques, formulating a robust feature set for subsequent modeling. Various ML classifiers were trained and tested on the union ensemble subsets to evaluate the performance in the classification stage. Each classifier was validated using different evaluation metrics to analyze its performance. This comparative analysis assists in selecting the classifier that best fits our prediction model. The best-performing classifier on the ensemble feature subset was later optimized using the GA hyperparameter optimization technique to enhance the classifier's final prediction. This whole process is named Optimized Union Ensemble Feature Selection (OUEFS). Fig. 2 shows the schematic diagram of our OUEFS methodology. The subsections below explain the details of the five key stages.

### B. DATASET

The dataset utilized in this study is sourced from the publicly available COVID-19 survival calculator, comprising data from 1,023,426 individuals, with a distribution of 98.80% COVID-19 negative and 1.20% COVID-19 positive patients, revealing a significant class imbalance. This dataset includes 59 columns, of which 40 are categorical and 19 are numeric, with approximately 13.8% of the data entries exhibiting missing values. (Detailed metadata can be accessed at https://www.covid19survivalcalculator.com/en/download, accessed on August 24, 2023).

### C. DATA PREPROCESSING

Several data preprocessing steps were undertaken to prepare the data for machine learning analysis. Initially, we cleaned the dataset based on the criteria in [20]. Metadata features such as region, immigrant, insurance, prescription, income, etc., that exhibited a high rate of missing values and geographical features irrelevant to the study's objectives were removed. Furthermore, missing values were addressed through an iterative imputation technique: continuous variables were imputed using the mean value, while categorical variables were imputed using the most frequent value. Table 3 shows the final COVID-19 clinical data with all compulsory selected features, which we use for further preprocessing. However, complete metadata, including feature name, description, type, and information on missing values for all 59 features, is provided in Appendix 1.

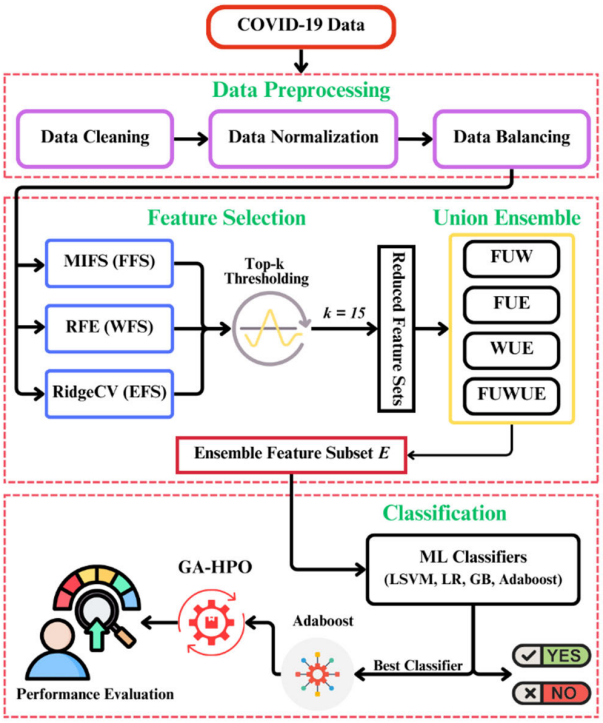


**FIGURE 2.** Schematic diagram of our proposed OUEFS approach.

Normalization was applied to ensure uniformity across the features, using Min-Max scaling to rescale categorical and continuous variables. This step helps mitigate any bias arising from differences in measurement scales and distribution ranges, facilitating a more accurate interpretation of the machine learning models' results. Min-Max normalization can be performed using the formula in Equation 3:

$$y' = \frac{y - \min(Y)}{\max(Y) - \min(Y)}, \tag{3}$$

where $y$ is a feature in dataset $Y$. $y$ is transformed by the Min-Max scaler into a normalized $y'$ within a specified range defined by the minimum ($\min(Y)$) and maximum ($\max(Y)$) values of $Y$.

To address the class imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) was employed to augment the minority class synthetically, achieving a 1:3 ratio between the COVID-19 positive and negative classes. This balancing technique is essential for enhancing the predictive accuracy of the models across different class labels, ensuring that the model does not favor the majority class. Upon examination of the original instances belonging to the minority class, the SMOTE algorithm is employed to synthesize new instances by leveraging the k-nearest neighbors' concept. The algorithm first consolidates all minority class instances into a set Y. For each instance $Y_{inst}$ within $Y$, a synthetic instance $Y_{new}$ is produced according to the following Equation 4 [40]:

$$Y_{new} = Y_{inst} + rand\ (0, 1) \times (Y_j - Y_{inst}), \tag{4}$$

where *rand(0,1)* represents a randomly generated value within the range [0, 1], and $Y_j$ is a randomly selected sample from the set $\{Y_1, Y_2, \ldots, Y_k\}$, which comprises the $k$ nearest neighbors of $Y_{inst}$. It is worth noting that, unlike other oversampling methods that duplicate minority class instances, the SMOTE algorithm generates novel, high-quality instances that closely resemble samples from the minority class [40], [41]. This process contributes to enhancing dataset balance and mitigating class imbalance issues commonly encountered in machine learning tasks. The dataset was refined through rigorous data cleaning, normalization, and balancing procedures to support robust machine learning analysis.

### D. FEATURE SELECTION METHODS

For our binary classification for predicting COVID-19, we explored the characteristics of the data to improve its usability. One technique in this area is feature extraction (FE), which focuses on reducing the data dimensionality by creating entirely new features from existing ones. The most used FE methods are principal component analysis (PCA), linear discriminant analysis (LDA), and kernel PCA, among many others. FE is often beneficial for datasets containing continuous numerical data. However, our study primarily focuses on FS, as our data consists mainly of categorical features, as shown in Table 3. In contrast to FE, FS aims to identify and retain the most informative existing features from the dataset for further analysis and classification.

**TABLE 3. Description of the COVID-19 clinical dataset.**

| No | Feature Name | Feature code | Description | Type |
|---|---|---|---|---|
| 1 | sex | SEX | Determine gender differences | Categorical |
| 2 | age | AGE | Patients age in number of years | Categorical |
| 3 | BMI | BMI | Measures the body fatness | Numerical |
| 4 | smoking | SMK | Indicating as a smoker or not | Categorical |
| 5 | alcohol | ALC | Alcohol drink (yes, no) | Numerical |
| 6 | cannabis | CNB | Cannabinoid's chemicals | Numerical |
| 7 | amphetamines | APT | Drugs to speed up the body | Numerical |
| 8 | cocaine | CCN | Synthetically prepared drug | Numerical |
| 9 | contacts_count | CTC | Contacts with patients or not | Numerical |
| 10 | working | WKG | Individuals working or not | Categorical |
| 11 | rate_red_risk_single | RRR | Determine the risk-reducing | Categorical |
| 12 | rate_reducing_mask | RRM | The rate of mask-reducing | Categorical |
| 13 | covid19_sympt om | CDS | Possible symptoms of coronavirus | Categorical |
| 14 | covid19_contac t | CDC | Individuals count with COVID | Categorical |
| 15 | asthma | AST | Breathing problem | Categorical |
| 16 | kidney_disease | KDD | Kidney working properly or not | Categorical |
| 17 | liver_disease | LVD | Patient has liver disease or not | Categorical |
| 18 | compr_immune | CPI | Individuals with weak immune | Categorical |
| 19 | heart_disease | HTD | Healthy or unhealthy heart | Categorical |
| 20 | lung_disease | LGD | Patient with a lung problem | Categorical |
| 21 | diabetes | DBT | Having high blood sugar or not | Categorical |
| 22 | hiv_positive | HIV | HIV-positive or negative | Categorical |
| 23 | hypertension | HPT | Due to various health concerns | Categorical |
| 24 | other_chronic | OTC | Patients with chronic diseases | Categorical |
| 25 | nursing_home | NSH | Nursing home | Categorical |
| 26 | health_worker | HTW | Working in healthcare | Categorical |
| 27 | covid19_positiv e | COVI D-19 | COVID-19 (yes, no) | Categorical |

Our approach uses three FS methods (MIFS, RFE, and RidgeCV) and combines the results to create an ensemble feature subset. These methods are included to diversify the FS process while improving our approach's regularity. The strategic use of diverse methodologies can yield significant

advantages in boosting performance. Using multiple methods increases the computing cost. Thus, selecting individual FS methods and combining them to get an optimal ensemble feature subset presents notable advantages regarding computational efficiency and predictive accuracy.

Each FS method chooses relevant features and ranks them according to their importance scores. All three FS methods are known as rankers. They do not simply select a few features to concentrate on; instead, they give equal weight to all of them. We decided to employ these rankers for the following reasons: (i) each has its selection evaluation criteria and exhibits a substantial level of heterogeneity within the final ensemble; (ii) all the methods (or rankers) assume a dataset with a balanced class distribution and do not explicitly account for class imbalance; (iii) none of the methods can individually recognize redundant features [30].

To use the ranking-based FS techniques, we first need to sort the features that can be used. A threshold needs to be defined to obtain a subset of important features. In this paper, we used the Top-k features thresholding method [42], referring to the prior studies [43], [44]. In the Top-k threshold, k is the total number of most significant features in the dataset based on their importance scores, typically derived from feature selection methods. The input consists of a set of features $F$ with corresponding importance scores, along with an integer $k$ representing the desired number of top features to select. The algorithm sorts the features in descending order of their scores, creating a ranked list $R$. It then selects the top $k$ features from this list to form a subset $R_k$, which is subsequently used for building predictive models (see Algorithm 1). Our method focused on choosing the top $k = 15$ features according to their importance scores for the COVID-19 disease prediction.

**Algorithm 1** Top-k Feature Thresholding

**Input:**
- Set of features $F = \{f_1, f_2, \ldots, f_n\}$
- Corresponding set of scores $S = \{s_1, s_2, \ldots, s_n\}$ where $s_i$ represents the importance score of features $f_i$
- Integer $k$ representing the desired number of top features to select ($k \leq n$)

**Output:**
Subset $R_K = \{r_1, r_2, \ldots, r_k\}$ containing the top $k$ features based on scores $S$

**Step 1: Sort Features by Scores**
Create a list $R$ by ordering features $F$ based on their scores $S$ in descending order.
$R$ = sort_features_by_scores ($F$, $S$), where $s_{(ri)} \geq s_{(ri+1)}$ for all $i$
**Step 2: Select Top-k Features**
Select the top $k$ features from the sorted list $R$ to form $R_K$.
$R_K = R[0:k]$
**Step 3: Model Building**
Use the subset $R_k$ to build the predictive model.
**Step 4: sort_features_by_score*(F, S)***
Initialize $R$ as an empty list.
Sort features $F$ based on their corresponding scores $S$ in descending order.
Return the sorted list $R$.

### E. UNION ENSEMBLE FEATURE SELECTION APPROACH

As explained earlier, we combined individual FS methods (MIFS, RFE, and RidgeCV) and selected a distinct feature subset. We used union ensemble feature selection (UEFS) to combine the feature subsets produced by individual FS methods. The union ensemble technique with all possible union combinations of individual FS methods used for this study is shown in Fig. 2. The process of combining feature subsets guarantees the retention of important features and greatly contributes to improving predictive performance.

Applying the individual FS technique results in reduced feature subsets, $M_i$ and $M_j$, when using a training dataset with $f$ features. The union operation takes into account all of the features that are current in either $M_i$ or $M_j$ or in both at the same time. The union of feature subsets can be defined mathematically as shown in Equation (5):

$$f \epsilon \left(M_i \cup M_j\right) \leftrightarrow f \varepsilon M_i \, V \, f \varepsilon M_j \tag{5}$$

The union of two or N feature subsets can be defined in Equation (6) and (7):

$$\cup \left\{M_i, M_j\right\} = M_i \cup M_j \tag{6}$$

and

$$\cup \{M_i, M_j, , \ldots, M_n\} = M_i \cup M_j \cup \ldots \cup M_n \tag{7}$$

Therefore, the ensemble feature subset using union operation can be obtained by the Equation (8) below:

$$E = M_i \cup M_j \cup \ldots \cup M_n \tag{8}$$

Other ensemble FS approaches, such as intersection or multi-intersection, were also considered. However, they lead to fewer features and fail to retain important features. Initial evaluation showed much lower performance than the UEFS, thus we did not pursue them further.

### F. GA-HPO

Genetic algorithm is meta-heuristic optimization technique inspired by principles of natural selection, specifically tailored to identify high-quality solutions to complex optimization problems [37]. These algorithms leverage biologically inspired operations, including mutation, crossover, and selection, to explore solution spaces [38] efficiently. Fig. 3 depicts the basic structure of a GA in the context of machine learning HPO. It operates in the steps below:

1. Beginning with a diverse population of potential solutions represented as chromosomes (each coding a set of hyperparameter combinations), each individual's fitness (chromosome) is evaluated using an objective function.
2. If the best individual (chromosome) satisfies the optimization criteria, the process terminates, assuming that this individual represents the solution to the problem.
3. If the optimization criteria are not met, a new generation is created. Pairs or individuals are randomly

selected and subjected to crossover and mutation operations.
4. The resulting individuals are selected based on their fitness to produce new offspring.

The performance of a GA hinges on a set of control parameters such as population size, crossover and mutation rates, and selection strategy. We determine the population size through experiments with various systems, which appear less sensitive to system size when confined within a small bounded network. Based on statistical evaluations of simulation outcomes, it was observed that a moderate population size ranging from 10 to 30 yields satisfactory results. The crossover rate ('Cr') dictates how frequently the crossover operator is applied, with higher rates ('Cr') facilitating quicker generation of new individuals. The mutation is crucial for maintaining population diversity and escaping local optima. Experiments have shown that crossover rates ranging from 0.6 to 0.9 and mutation rates between 10% to 60% offer effective computational performance, with lower crossover rates and higher mutation rates enhancing computational efficiency.

GA is robust and effective across various optimization problems driven by two main advantages. First, GA navigates search spaces using independent individuals, enabling parallel processing and minimizing the risk of getting stuck in local optima encountered by other optimization methods. Second, GA's implementation is straightforward and adaptable.

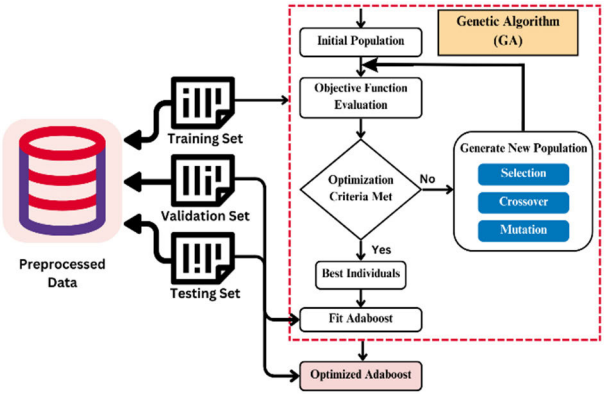


**FIGURE 3.** GA-HPO for Adaboost classifier.

### G. CLASSIFICATION

In classification, we evaluated the UEFS method using LSVM, LR, GB, and Adaboost classifiers. The experimental results will be shown later in Section V. Ensemble methods were used to choose the subset of features with the highest prediction performance. The Adaboost classifier was the best performing and was used for later experiments to predict COVID-19 patients.

Adaboost is an ensemble approach that combines the strengths of various 'T' weak classifiers to improve classification performance. Depending on how well each weak classifier performed during each iteration, the weight that the Adaboost assigns varies. The final prediction function 'f(x)' in Adaboost is formed as a weighted combination of the weak classifiers, as shown in Equation (9) below, where $\alpha\_t$ is the weight given to the h_t(x) weak classifier.

$$f\ (x) = sign(\Sigma[\alpha\_t * h\_t(x)]) \tag{9}$$

We carefully examined the performance of the Adaboost algorithm in making predictions for the COVID-19 disease using the ensemble feature subsets. We implemented a strategic plan to optimize its Adaboost hyperparameters using GA to achieve higher performance (see Fig. 3). We preprocessed the dataset as discussed above in subsection C and applied SMOTE algorithms to balance the target class (covid19_postive). For GA-HPO, this dataset was split into training, validation, and testing sets using a 70:15:15 ratio. The training set, comprising 975,727 instances, was used to train the machine learning classifier and validate its performance using the validation set. The validation set of 209,084 instances allowed us to fine-tune model hyperparameters and monitor for overfitting during training.

The key parameters tuned for Adaboost include n_estimators, n_jobs, learning_rate, base_estimator_param, verbose, and random_state. Finally, we performed testing using the test set containing 209,085 instances. Table 4 includes the number of instances for each class in each subset.

Our integration of GA-based HPO led to the fine-tuning of the Adaboost model. It was marked by notably higher accuracy and a robust solution to the challenge of COVID-19 patient prediction using clinical data. The detailed procedure of our OUEFS classification methodology is shown in Algorithm 2.

**TABLE 4.** Number of instances for each subset.

| Subsets | Total Instances | COVID19-Positive | COVID19-Negative | Subsets |
|---|---|---|---|---|
| Training | 975727 | 487546 | 488181 | Training |
| Validation | 209084 | 104697 | 104387 | Validation |
| Testing | 209084 | 104705 | 104380 | Testing |

## V. EXPERIMENTAL RESULTS

We conducted in-depth evaluations of the OUEFS approach using the COVID-19 dataset explained in Section IV. In this section, we show the experimental results of ML classifiers using a full feature dataset, individual FS approaches, UEFS, OUEFS (UEFS + GA-HPO), and a comparison of the above results. In particular, UEFS experiments are designed to compare the performance of various union combinations and select the optimal one to optimize hyperparameters further. Moreover, we compare the results of our proposed OUEFS approach with those of the previous research. We used various ML libraries within the Python programming language. All the experiments were conducted on a system equipped with an Intel (R) Core i7-2600 CPU operating at a frequency of 3.40 GHz.

**Algorithm 2** Optimized Union Ensemble Feature Selection(OUEFS)

**Require:**
Dataset D, partitioned as {D_train, D_test}, Total number of features, number of top features to select K, Ensemble methods specified as $\varepsilon \in$ {U, I, M}, Machine learning classifier denoted as Cls

**Ensure:**
- Classification prediction represented as P
- Optimized machine learning classifier denoted as Opt_Cls

**Step 1: Apply Multiple FS Methods**
for each FSMethod in $\{\alpha, \beta, \gamma\}$ do
apply FSMethod on D_train to obtain the respective feature set (FS_i)
end for
**Step 2: Apply K-top Feature Thresholding**
for each FS_i, do
FS_i' = SelectTopKFeatures (FS_i, K)
end for
**Step 3: Ensemble the Selected Feature Subsets using Union**
if $\varepsilon$ = U then
E = Union of all FS_i'
else if $\varepsilon$ = I then
end if
**Step 4: Apply Machine Learning Classifier**
Partition D_train into {D_train_t, D_train_val}
Cls = Train (Cls, D_train_t, E)
$\Phi$ = Evaluate (Cls, D_train_val)
**Step 5: Optimize Hyperparameters using GA**
if $\theta$ is true, then
Opt_Cls = Optimize ($\zeta$, Cls, $\Phi$ )
else
Opt_Cls = Cls
end if
**Step 6: Output Classification Prediction and Optimized ML Classifier**
P = Predict (Opt_Cls, D_test)
return P, Opt_Cls

### A. RESULTS WITH FULL FEATURES

We first compared ML classifiers (LSVM, GB, LR, Adaboost) on the entire set of features (27 features shown in Table 3 ). Table 5 shows that the AdaBoost and GB classifiers achieved high accuracies (93.24% and 91.93% respectively). The Adaboost performs better than others by obtaining a precision of 96.95% and a recall rate of 94.33%. These metrics collectively contribute to an impressive F1-score of 95.62%. With accuracy values of 78.87 and 78.97, the LSVM and LR were the least accurate classifiers.

**TABLE 5. Experimental results using full features.**

| Classifiers | Accuracy | Precision | Recall | F1-score | AUC |
|---|---|---|---|---|---|
| LSVM | 78.87 | 80.30 | 76.42 | 78.31 | 87.31 |
| GB | 91.93 | 93.94 | 89.49 | 91.69 | 91.90 |
| LR | 78.97 | 80.11 | 76.99 | 78.52 | 87.37 |
| Adaboost | 93.24 | 96.95 | 94.33 | 95.62 | 97.04 |

### B. RESULTS OF INDIVIDUAL FS METHODS

In this subsection, we show the experimental results using the feature subsets selected by the individual FS approaches. The list of features chosen using individual FS techniques (MIFS, RFE, RidgeCV) is shown in Table 6. It also shows a ranking of the selected features, arranged in descending order of their importance scores. These selected feature subsets are used for the experiments later in this subsection and the following subsections. Each selector independently determines the top 15 relevant features by the Top-k features thresholding. As explained above, the most crucial features were chosen using MIFS, RFE, and RidgeCV FS methods.

**TABLE 6. Ranks and importance scores of selected features using different FS methods (Code: Feature Code, Score: Importance Score.)**

| Rank | MIFS | | RFE | | RidgeCV | |
|---|---|---|---|---|---|---|
| No. | Code | Score | Code | Score | Code | Score |
| 1 | APT | 0.090 | OTC | 2.48 | OTC | 0.212 |
| 2 | CCN | 0.083 | CDS | 1.41 | CDS | 0.049 |
| 3 | RRR | 0.071 | RRR | 1.43 | CPI | 0.019 |
| 4 | SEX | 0.047 | SEX | 0.893 | KDD | 0.016 |
| 5 | AGE | 0.038 | CPI | 0.760 | AST | 0.014 |
| 6 | WKG | 0.036 | KDD | 0.491 | RRR | 0.010 |
| 7 | RRM | 0.031 | CDC | 0.476 | SEX | 0.010 |
| 8 | CCT | 0.014 | AST | 0.423 | DBT | 0.007 |
| 9 | AST | 0.011 | DBT | 0.414 | CDC | 0.004 |
| 10 | HPT | 0.007 | LGD | 0.256 | NSH | 0.003 |
| 11 | SMK | 0.007 | HIV | 0.232 | LGD | 0.002 |
| 12 | CNB | 0.006 | HTD | 0.207 | LVD | 0.001 |
| 13 | ALC | 0.005 | SMK | 0.108 | RRM | 0.001 |
| 14 | DBT | 0.003 | ALC | 0.087 | HPT | 0.001 |
| 15 | CPI | 0.003 | HPT | 0.087 | SMK | 0.001 |

In the first experiment, we used the selected subset of 15 features of MIFS from Table 6. These features influence the majority of the final COVID-19 patient prediction. Table 7 shows that the Adaboost classifier had the best classification performance: 94.31% accuracy, 95.29% precision, 93.22% recall, 94.24% F1-score, and 98.61% AUC-ROC. The GB classifier also performed well, with an accuracy of 91.16%. In addition, when compared with full features, Adaboost improved the accuracy by 1.07% (94.31% - 93.24%), while LSVM and LR had balanced performance and attained almost the same level of accuracy.

Table 7 also shows the results of ML classifiers using a subset of RFE-based 15 most important features. The accuracies achieved by the LSVM, GB, LR, and Adaboost models are 79.04%, 79.71%, 79.03%, and 79.63%, respectively. Adaboost performed highly in the other evaluation measures (precision and recall, among others). Compared with the MIFS, all classifiers perform low on the RFE-based feature subset. However, the final prediction depends entirely on the number of features the individual FS method selects.

Experimental results using a subset of RidgeCV-based 15 most important features in Table 7 indicate that most

**TABLE 7. Experimental results of classifiers using a selected subset of individual FS methods.**

| Classifiers | Accuracy | Precision | Recall | F1-score | AUC |
|---|---|---|---|---|---|
| MIFS + LSVM | 76.76 | 78.00 | 74.44 | 76.18 | 84.87 |
| MIFS + GB | 91.16 | 92.95 | 89.05 | 90.96 | 91.16 |
| MIFS + LR | 76.78 | 77.82 | 74.83 | 76.30 | 84.92 |
| MIFS + Adaboost | 94.31 | 95.29 | 93.22 | 94.24 | 98.61 |
| RFE + LSVM | 79.04 | 77.10 | 82.66 | 79.78 | 88.13 |
| RFE + GB | 79.71 | 76.95 | 84.87 | 80.72 | 79.71 |
| RFE + LR | 79.03 | 77.16 | 82.51 | 79.74 | 88.14 |
| RFE + Adaboost | 79.63 | 77.15 | 84.23 | 80.53 | 88.75 |
| RidgeCV + LSVM | 78.86 | 78.25 | 79.96 | 79.10 | 87.56 |
| RidgeCV + GB | 79.27 | 78.84 | 80.04 | 79.44 | 79.27 |
| RidgeCV + LR | 78.83 | 78.16 | 80.05 | 79.09 | 87.58 |
| RidgeCV + Adaboost | 78.86 | 77.26 | 81.84 | 79.48 | 88.22 |

classifiers showed comparable performance. However, the Adaboost had an advantage in other crucial evaluation metrics such as AUC and recall. The Adaboost obtained 78.86% accuracy, 77.26% precision, 81.84% recall, 79.48% F1-score, and 88.22% AUC. GB showed slight improvements. Compared with RFE, the performance of classifiers was almost at the same level. However, their performance is low when compared with the MIFS. This is because the RFE and RidgeCV selected nearly similar types of features.

Although we did not use PCA in our study as a first choice since it is not well suited for the categorical data type, we also conducted experiments by applying PCA for feature extraction. PCA derives a new set of features, principal components (PCs), from linear combinations of the original variables. These PCs are structured to be orthogonal and are formulated sequentially based on the amount of variance they capture from the original dataset. Our study employed PCA to identify and extract the top 15 features based on their variance from the data. This refined feature subset was split into training and testing sets to evaluate four different ML classifiers: LSVM, GB, LR, and AdaBoost. Among these, GB exhibited the highest overall accuracy at 77.64% and a robust balance between precision and recall, as reflected in its F1 score of 75.23%. All models demonstrated competent performance metrics, as shown in Table 8. This highlights that while PCA effectively reduces dimensionality, alternative feature selection methods such as MIFS capture the predictive nuances of the dataset better.

The above results show that the FS approaches heavily influence classifiers' performance. The MIFS method achieved better performance across all classifiers. Adaboost was identified as the most effective classifier, particularly when combined with the MIFS FS approach.

**TABLE 8. Experimental results of classifiers using a selected subset of the PCA feature extraction method.**

| Classifiers | Accuracy | Precision | Recall | F1-score | AUC |
|---|---|---|---|---|---|
| PCA + LSVM | 72.59 | 85.93 | 54.32 | 66.56 | 79.23 |
| PCA + GB | 77.64 | 84.79 | 67.60 | 75.23 | 77.68 |
| PCA + LR | 73.27 | 85.36 | 56.46 | 67.96 | 79.66 |
| PCA + Adaboost | 76.66 | 83.83 | 66.32 | 74.05 | 84.54 |

### C. RESULTS OF UNION ENSEMBLE METHOD

This section presents the experimental results using the union ensemble FS approach. The approach provides a comprehensive comparison among different combinations of individual FS methods. It also identifies an optimal subset encompassing all potential key features from the database, which can yield high classification performance using ML classifiers. Table 9 shows the ensemble feature subsets obtained using the union method. Union combinations such as (F∪W), (F∪E), (W∪E), and (F∪W∪E)[1] provide a set of up to 22, 23, 19, and 24 candidate features, respectively. Using these combinations, the performance of each ML classifier was then evaluated.

Table 10 shows that Adaboost surpasses all other classifiers and achieves the highest accuracy rates of 94.55%, 94.66%, 81.79%, and 94.51% in the context of the F∪W, F∪E, W∪E, and F∪W∪E combinations, respectively. It indicates the Adaboost's superiority over the subsets selected by the individual FS methods (Table 7 ) and the complete set of features (Table 5 ). The methods F∪W, F∪E, and F∪W∪E exhibited comparable results when applied to the union of selected features. This same performance level results from the MIFS-based selected features combined with other methods. In contrast, the W∪E combination shows the lowest performance, resulting in an accuracy of 81.79%. However, compared with RFE and RidgeCV individual FS methods, Adaboost using W∪E combinations improved the accuracy by 2.16% and 2.93% over RFE and RidgeCV, respectively. These results show that the union ensemble strategy is effective, especially when used with the MIFS method.

The performance results highlight that the UEFS approach is useful for disease classification. Adaboost, among other ML classifiers, yields the highest performance. Furthermore, the performance is mainly influenced by the choice of individual FS methods to extract the key features. Notably, the MIFS method adeptly extracted key features from the database, substantially contributing to the final predictions. The results in Tables 7 and 10 show that evaluating ML classifiers on the MIFS feature subset and exploring all possible UEFS combinations of MIFS features with other individual FS methods yields superior performance. For the rest of the experiments, we utilized the best-performing union

[1] F = filter (MIFS); W = wrapper (RFE); E = embedded (RidgeCV).

**TABLE 9.** List of important selected features by the UEFS methods.

| Rank | Union | | | |
|---|---|---|---|---|
| No. | FUW | FUE | WUE | FUWUE |
| 1 | SEX | SEX | OTC | SEX |
| 2 | AGE | AGE | CDS | AGE |
| 3 | SMK | SMK | RRR | SMK |
| 4 | ALC | ALC | SEX | ALC |
| 5 | CNB | CNB | CPI | CNB |
| 6 | APT | APT | KDD | APT |
| 7 | CCN | CCN | CDC | CCN |
| 8 | CCT | CCT | AST | CCT |
| 9 | WKG | WKG | DBT | WKG |
| 10 | RRM | RRM | LGD | RRM |
| 11 | CCT | CCT | HIV | CCT |
| 12 | AST | AST | HTD | AST |
| 13 | HPT | HPT | SMK | HPT |
| 14 | DBT | DBT | ALC | DBT |
| 15 | CPI | CPI | HPT | CPI |
| 16 | OTC | OTC | CDC | OTC |
| 17 | HIV | CDC | NSH | CDC |
| 18 | LGD | KDD | LVD | KDD |
| 19 | HTD | CDC | RRM | CDC |
| 20 | KDD | NSH | | NSH |
| 21 | CDC | LGD | | LGD |
| 22 | CDS | LVD | | LVD |
| 23 | | RRM | | RRM |
| 24 | | | | HIV |

subset (F∪E) that yields accurate results with an accuracy of 94.66%.

**TABLE 10.** Experimental results using union ensemble feature subsets.

| Classifiers | Accuracy | Precision | Recall | F1-score | AUC |
|---|---|---|---|---|---|
| Union (F∪W) | | | | | |
| LSVM | 78.28 | 79.64 | 75.91 | 77.73 | 86.76 |
| GB | 90.79 | 92.66 | 88.57 | 90.57 | 90.79 |
| LR | 78.40 | 79.55 | 76.38 | 77.93 | 86.83 |
| Adaboost | 94.55 | 95.85 | 93.12 | 94.47 | 98.81 |
| Union (F∪E) | | | | | |
| LSVM | 78.21 | 79.63 | 75.74 | 77.63 | 86.70 |
| GB | 91.27 | 93.14 | 89.09 | 91.07 | 91.27 |
| LR | 78.31 | 79.52 | 76.19 | 77.82 | 86.76 |
| Adaboost | 94.66 | 95.96 | 93.24 | 94.58 | 98.79 |
| Union (W∪E) | | | | | |
| LSVM | 75.40 | 77.81 | 70.97 | 74.24 | 82.90 |
| GB | 79.64 | 81.90 | 76.03 | 78.85 | 79.63 |
| LR | 75.39 | 77.68 | 71.17 | 74.28 | 82.91 |
| Adaboost | 81.79 | 84.13 | 78.30 | 81.11 | 90.02 |
| Union (F∪W∪E) | | | | | |
| LSVM | 78.29 | 79.68 | 75.89 | 77.74 | 86.75 |
| GB | 90.91 | 92.79 | 88.68 | 90.69 | 90.90 |
| LR | 78.37 | 79.55 | 76.30 | 77.89 | 86.80 |
| Adaboost | 94.51 | 95.87 | 93.01 | 94.42 | 98.74 |

We also evaluated the efficacy of intersection and multi-intersection methods. However, the performance was lower than that of the union method. The low performance stemmed from the minimal number of features and the selected subsets derived from individual FS methods. For example, the intersection approach discarded features based on overlapping FS methods. It led to the loss of valuable information related to the target variable. With fewer features, classifiers might memorize the training data but need to generalize the new cases, which affects the performance.

### D. RESULTS OF OUEFS METHOD USING GA-HPO AND COMPARISON WITH OTHER METHODS

Experimental results in the previous subsections showed that the Adaboost using the (F∪E) combination outperformed all the other experiments. To further boost the performance of the Adaboost, we used the GA-HPO method to provide the best possible combinations of hyperparameters for the model. The acquired configured set of hyperparameters improves the performance during the model training on the (F∪E) subset. The hyperparameters of the Adaboost model with the range and optimal values obtained by GA are explained earlier in Section IV-F. Experimental results show that the Adaboost classifier and tuned hyperparameters obtained using GA outperformed all the previous research in predicting COVID-19 disease and achieved the highest accuracy of 96.30%. As mentioned, we named this whole process OUEFS (UEFS + GA-HPO).

Fig. 4 and Fig. 5 show performance comparisons of all methods, including the OUEFS approach, in terms of accuracy and AUC, respectively. Compared with the other studies, the OUEFS technique performed better and correctly predicted the COVID-19 patients. Fig. 4 shows that the OUEFS approach achieved the highest accuracy of 96.30%, an improvement of 1.64% above the previous best union ensemble (F∪E) method as shown in Fig. 6. Compared with the full feature, the OUEFS technique improves the accuracy by 3.06%. Similarly, compared with the average accuracy of the previous 13 studies (82.28%), our proposed OUEFS obtained an accuracy of 96.30%, showing a 14.02% performance improvement. The OUEFS technique also improved performance for the other evaluation metrics and accurately recognized the positive samples.

We further show an AUC-based comparison of all studies in Fig. 5. The OUEFS achieved a superior AUC score of 98.99%. It also outperformed well for the other evaluation metrics, including precision of 98.02%, recall of 97.56%, F1-score of 97.01%, and Gini coefficient of 94.05%. The higher results indicate that our approach accurately observed COVID-19-positive cases and significantly enhanced the performance of the Adaboost algorithm.

The computing time associated with the above experiments is further shown in Fig. 7. The Full Feature method, which uses all available features without any selection, recorded a computing time of 3.69 seconds, serving as a baseline for comparison. Among individual FS methods, MIFS showed

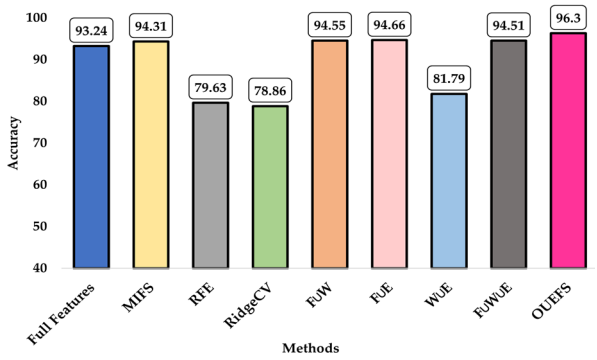


**FIGURE 4.** Accuracy comparison of all approaches.

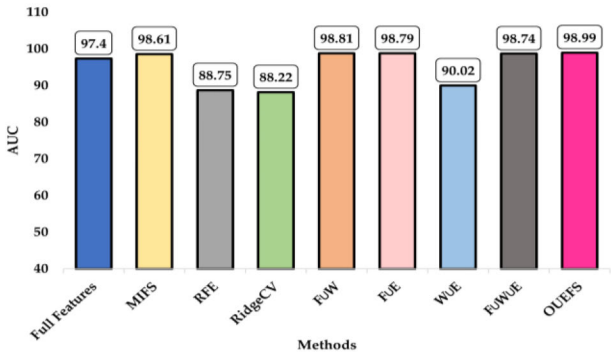


**FIGURE 5.** AUC comparison of all approaches.

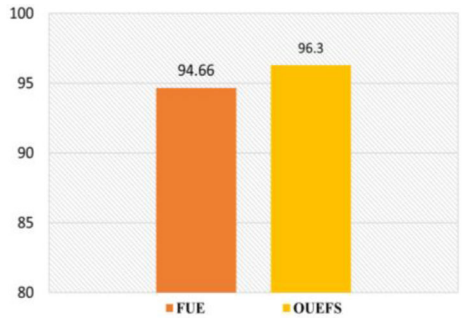


**FIGURE 6.** Accuracy comparison of FUE and OUEFS experiments.

a computing time close to the Full Feature method, while RFE exhibited a slightly lower time of 3.64 seconds. For ensemble feature selection methods, FUW recorded a computing time of 3.67 seconds, and FUE improved the time with 3.43 seconds. The WUE method had a higher computing time of 4.29 seconds, suggesting that combining wrapper and embedded methods increases the computing time due to algorithmic complexity. Finally, the FUWUE (Filter-Union-Wrapper-Embedded) method showed a computing time of 3.31 seconds. This indicates that while it integrates multiple feature selection strategies, it maintains a reasonable computing time. Our OUEFS approach provides a balanced and efficient solution, showing a computing time of 3.43 seconds while significantly improving the Adaboost classifier's accuracy for COVID-19 prediction. Thus, OUEFS is not only effective in enhancing model performance but also efficient in computing time.

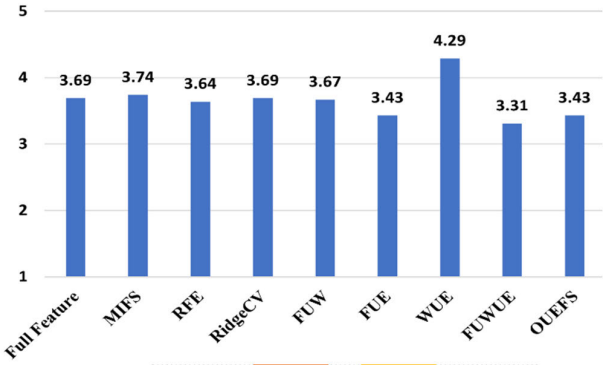


**FIGURE 7.** Computation time (seconds) comparison of all the experiments.

### E. PERFORMANCE COMPARISON WITH EXISTING ENSEMBLE-BASED MODELS

Table 11 compares our proposed OUEFS technique with the previous state-of-the-art ensemble-based optimization methods focusing on classification accuracy. OUEFS approach consistently outperforms other approaches. Debjit et al. [17] is the previous best (92.23% accuracy) ensemble technique for COVID-19 identification. Using the same clinical dataset consisting of 1,023,426 samples with a 1.20% positive ratio, our approach achieved 4.07% better accuracy (96.30%).

**TABLE 11.** Performance comparison of our proposed OUEFS with ensemble-based optimization approaches.

| Year | Previous Research | Accuracy (%) |
|---|---|---|
| 2021 | Koushal Kumar [19] | 91.20 |
| 2021 | Koushik et al. [23] | 89.82 |
| 2022 | Debjit et al. [17] | 92.23 |
| **2024** | **Proposed** | **96.30** |

### F. MODEL INTERPRETATION USING EXPLAINABLE SHAP ANALYSIS

Understanding the global interpretability of predictive models is essential, particularly in classification frameworks where the contribution of predictor attributes significantly impacts model performance. SHAP (Shapley Additive exPlanations) plot, specifically the SHAP summary plot, effectively merges feature importance with the detailed effects of each feature. This method assigns scores to each input feature based on the mean absolute Shapley values, illustrating their utility in predicting a target variable. This visualization prioritizes features and ranks them on the y-axis by importance.

At the same time, the x-axis displays SHAP values that highlight the correlation of each feature with the target outcome: positive values indicate a positive correlation, and vice versa. The color gradient, transitioning from blue (low values) to red (high values), further distinguishes feature values, enhancing interpretability and providing insights into data dispersion. This concise and comprehensive approach augments the efficacy and efficiency of predictive models. It provides a nuanced understanding of how individual features influence predictions, as demonstrated in previous COVID-19 disease prediction studies [46], [47], [48].

In our analysis, SHAP plots, particularly the summary and beeswarm plots, demonstrate their efficacy by delineating the influence and significance of features within our predictive model. For example, in our dataset (Fig. 8), 'sex' and 'rate_reducing_mask' are identified as the most influential predictors, with higher values (represented in red) associated with positive Shapley values indicative of a COVID-Positive outcome. In comparison, lower values (shown in blue) suggest a no-COVID outcome. Conversely, features like 'kidney_disease,' 'asthma,' and other similar conditions show minimal impact. The plot arranges features in descending order of influence, providing a global interpretation that aligns with model-specific explainable AI results and enhances model transparency and validation against established medical knowledge. This integration of SHAP visualizations ensures that interpretations are scientifically robust and practically relevant, fostering a deeper understanding of the model's predictive dynamics.

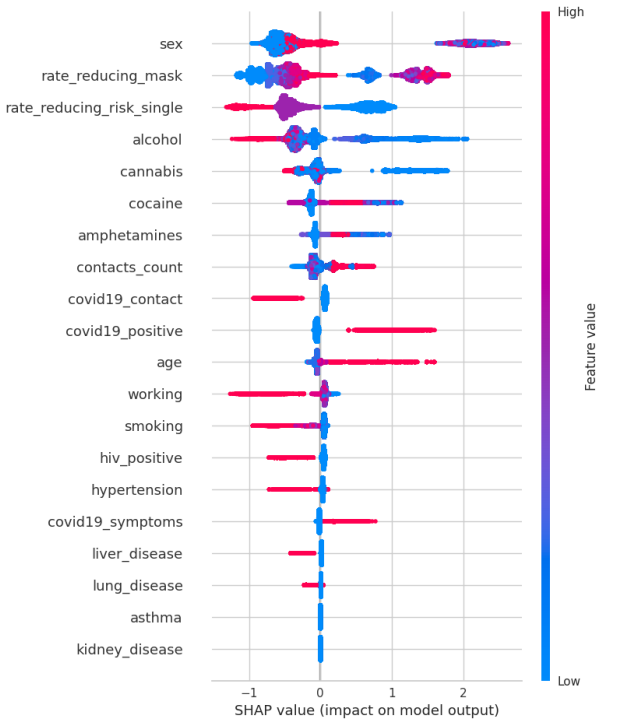


**FIGURE 8.** SHAP analysis.

### G. EXPERIMENTS ON ISRAEL COVID-19 CLINICAL DATASET

We evaluated our approach using another Israel COVID-19 dataset from the open-source GitHub repository extracted from the Israeli Ministry of Health website [49]. The dataset includes ten features representing clinical symptoms, detailed in Appendix 2.

First, we processed the data by removing missing values and excluding features that did not contribute to the final prediction. We then applied the Min-Max scaler to rescale categorical and continuous variables. We used the SMOTE data balancing approach to enhance predictive performance and balance the target class. The dataset was split into training (80%) and testing (20%) sets.

Our FS process was rigorous, employing MIFS, RFE, and RidgeCV algorithms. These algorithms ranked features by their importance scores, as shown in Table 12. We used a Top-k threshold method to select the most significant features, with the top 5 features considered. The selected feature subsets were then combined using the union ensemble FS approach. Appendix 3 showcases the ensemble feature subsets obtained through union combinations such as (F∪W), (F∪E), (W∪E), and (F∪W∪E), yielding up to 7, 7, 6, and 7 candidate features, respectively.

We conducted a series of experiments similar to those for the COVID-19 dataset. The experimental results in Table 13 show that the GB and Adaboost classifiers performed well in all experiments. The GB classifier achieved 80.30% accuracy using the full feature set, achieving up to 91.85% accuracy with the MIFS feature subset. All ML classifiers performed well for union combinations, with GB achieving 92.27% accuracy for the FUWUE combination. The GB classifier was also optimized using GA-HPO, improving its performance by 2.05% compared with the previous best during the FUWUE experiment. Fig. 9 shows the accuracy comparison of the GB classifier across all experiments.

**TABLE 12.** Ranks and importance scores of selected features using different FS methods (Code: Feature Code, Score: Importance Score.)

| | MIFS | | RFE | | RidgeCV | |
|---|---|---|---|---|---|---|
| No. | Code | Score | Code | Score | Code | Score |
| 1 | TI | 0.283 | HD | 2.623 | HD | 0.371 |
| 2 | Gender | 0.122 | SoB | 1.987 | Fever | 0.265 |
| 3 | HD | 0.0319 | ST | 1.902 | ST | 0.224 |
| 4 | Fever | 0.027 | Fever | 1.786 | SoB | 0.201 |
| 5 | Cough | 0.019 | Gender | 1.253 | Gender | 0.193 |
| 6 | ST | 0.014 | Cough | 0.933 | Cough | 0.131 |
| 7 | Age60 | 0.010 | Age60 | 0.256 | Age60 | 0.016 |
| 8 | SoB | 0.004 | TI | 0.119 | TI | 0.008 |

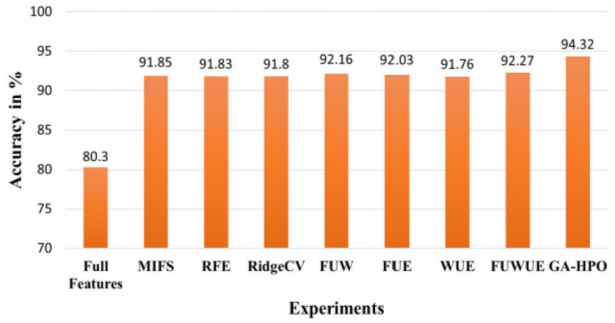


**FIGURE 9.** Accuracy comparison of GB classifier for various experiments.

**TABLE 13.** Experimental results for full features, individual FS methods, union combinations, and GA-HPO.

| Classifiers | Accuracy | Precision | Recall | F1-score | AUC |
|---|---|---|---|---|---|
| Full Features | | | | | |
| LSVM | 79.89 | 79.73 | 68.94 | 75.80 | 76.98 |
| GB | **80.30** | 78.49 | 64.15 | 77.19 | 76.56 |
| LR | 80.17 | 79.26 | 63.15 | 76.62 | 77.02 |
| Adaboost | 79.69 | 77.64 | 71.42 | 75.35 | 76.97 |
| MIFS | | | | | |
| LSVM | 91.73 | 87.12 | 77.51 | 90.56 | 86.42 |
| GB | **91.85** | 83.52 | 78.25 | 84.84 | 82.54 |
| LR | 91.59 | 89.43 | 86.03 | 88.83 | 86.42 |
| Adaboost | 91.48 | 90.42 | 83.81 | 86.42 | 86.37 |
| RFE | | | | | |
| LSVM | 91.46 | 87.47 | 82.08 | 84.56 | 86.92 |
| GB | **91.83** | 92.08 | 80.82 | 83.53 | 84.79 |
| LR | 91.60 | 89.90 | 73.75 | 86.62 | 86.93 |
| Adaboost | 91.46 | 87.47 | 82.08 | 84.56 | 86.91 |
| RidgeCV | | | | | |
| LSVM | 91.69 | 88.38 | 86.40 | 89.50 | 86.93 |
| GB | **91.80** | 90.75 | 91.35 | 83.99 | 85.03 |
| LR | 91.75 | 87.50 | 87.08 | 90.27 | 86.93 |
| Adaboost | 91.26 | 83.58 | 90.98 | 89.73 | 86.93 |
| FUW | | | | | |
| LSVM | 91.78 | 89.72 | 85.21 | 88.31 | 86.80 |
| GB | **92.16** | 87.35 | 91.90 | 85.17 | 85.42 |
| LR | 91.81 | 88.34 | 86.47 | 89.57 | 79.79 |
| Adaboost | 91.72 | 90.85 | 84.42 | 87.41 | 86.75 |
| FUE | | | | | |
| LSVM | 91.65 | 88.27 | 85.11 | 88.13 | 86.92 |
| GB | **92.03** | 86.96 | 91.68 | 84.89 | 85.30 |
| LR | 91.69 | 87.85 | 86.37 | 89.40 | 79.91 |
| Adaboost | 91.59 | 89.25 | 84.20 | 87.08 | 86.87 |
| WUE | | | | | |
| LSVM | 91.42 | 89.65 | 92.26 | 84.80 | 88.80 |
| GB | **91.76** | 93.70 | 91.03 | 83.68 | 84.88 |
| LR | 91.48 | 86.91 | 83.00 | 85.72 | 88.80 |
| Adaboost | 91.40 | 90.64 | 82.08 | 84.57 | 90.78 |
| FUWUE | | | | | |
| LSVM | 91.81 | 88.04 | 87.85 | 91.05 | 91.89 |
| GB | **92.27** | 84.95 | 86.71 | 89.28 | 87.65 |
| LR | 91.92 | 91.14 | 85.44 | 87.31 | 90.70 |
| Adaboost | 91.62 | 90.86 | 90.82 | 82.96 | 84.37 |
| GA-HPO | | | | | |
| GB | **94.32** | 90.45 | 92.84 | 93.01 | 93.30 |

Table 14 compares the result of our proposed approach, utilizing Israeli COVID-19 data, with those of previous state-of-the-art studies. The comparison focuses on two key performance metrics: classification accuracy and AUC. Our ensemble approach demonstrates consistent outperformance over existing methods. The previous best-performing approach, reported by Hossen et al. [50], is an FS ensemble technique for COVID-19 identification, achieving an accuracy of 88.0%. When applied to the same clinical dataset, our method achieves a significantly higher accuracy of 94.32%, representing a 6.32% improvement. Additionally, a study by Zoabi et al. [51] employed a LightGBM classifier for COVID-19 patient classification and attained a best AUC score of 90%. Our approach surpasses this result, achieving a superior AUC of 92.30%, reflecting a 2.30% increase.

**TABLE 14.** Performance comparison of the proposed approach with previous studies.

| Year | Studies | Accuracy (%) | AUC (%) |
|---|---|---|---|
| 2021 | Zoabi et.al. [51] | - | 90.0 |
| 2024 | Hossain et al. [50] | 88.0 | 93.0 |
| **2024** | **Proposed** | **94.32** | **93.30** |

## VI. CONCLUSION

In this paper, we propose an automated COVID-19 prediction system using ML methods on the clinical dataset. The proposed method is intended to function in real-time, identifying COVID-19 patients as early as possible. The proposed OUEFS methodology combines feature selection with the ensemble technique. It combines subsets of features obtained through individual feature selection procedures using the union method. The machine learning classifiers were evaluated using various performance metrics, including accuracy, precision, and AUC.

Additionally, the top classifier, Adaboost, was optimized using the GA-HPO technique, which increased the overall performance by a significant margin and led to the highest accuracy (96.30%) compared with the previous ensemble-based approaches. These results highlight the efficacy of the proposed OUEFS strategy, which has the potential to be implemented in the healthcare systems for the early diagnosis of COVID-19 patients. In addition, our approach can be adaptively suited for predicting other diseases, such as cardiovascular conditions, diabetes, and hypertension.

## REFERENCES

[1] N. Jebril, "World Health Organization declared a pandemic public health menace: A systematic review of the coronavirus disease 2019 'COVID-19,'" Social Sci. Res. Netw., Rochester, NY, USA, SSRN Scholarly Paper ID 3566298, Apr. 2020.
[2] *WHO Coronavirus (COVID-19) Dashboard*. Accessed: Mar. 22, 2022. [Online]. Available: https://covid19.who.int
[3] G. Pascarella, "COVID-19 diagnosis and management: A comprehensive review," *J. Intern. Med.*, vol. 288, no. 2, pp. 192–206, Aug. 2020.
[4] T. Greenhalgh, J. L. Jimenez, K. A. Prather, Z. Tufekci, D. Fisman, and R. Schooley, "Ten scientific reasons in support of airborne transmission of SARS-CoV-2," *Lancet*, vol. 397, no. 10285, pp. 1603–1605, May 2021.
[5] J. Wu, "Rapid and accurate identification of COVID-19 infection through machine learning based on clinically available blood test results," *medRxiv*, Apr. 2020.

[6] H. X. Bai, B. Hsieh, Z. Xiong, K. Halsey, J. W. Choi, T. M. L. Tran, I. Pan, L.-B. Shi, D.-C. Wang, J. Mei, X.-L. Jiang, Q.-H. Zeng, T. K. Egglin, P.-F. Hu, S. Agarwal, F.-F. Xie, S. Li, T. Healey, M. K. Atalay, and W.-H. Liao, "Performance of radiologists in differentiating COVID-19 from Non-COVID-19 viral pneumonia at chest CT," *Radiology*, vol. 296, no. 2, pp. E46–E54, Aug. 2020.

[7] S. Rajaraman and S. Antani, "Training deep learning algorithms with weakly labeled pneumonia chest X-ray data for COVID-19 detection," *medRxiv*, May 2020.

[8] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "A review of feature selection methods on synthetic data," *Knowl. Inf. Syst.*, vol. 34, no. 3, pp. 483–519, Mar. 2013.

[9] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Elect. Eng.*, vol. 40, no. 1, pp. 16–28, Jan. 2014.

[10] V. Kumar, "Feature selection: A literature review," *Smart Comput. Rev.*, vol. 4, no. 3, pp. 211–229, Jun. 2014.

[11] H. Lim, J. Lee, and D.-W. Kim, "Optimization approach for feature selection in multi-label classification," *Pattern Recognit. Lett.*, vol. 89, pp. 25–30, Apr. 2017.

[12] J. Wang, J.-M. Wei, Z. Yang, and S.-Q. Wang, "Feature selection by maximizing independent classification information," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 4, pp. 828–841, Apr. 2017.

[13] T. Zhang, P. Ren, Y. Ge, Y. Zheng, Y. Y. Tang, and C. L. P. Chen, "Learning proximity relations for feature selection," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 5, pp. 1231–1244, May 2016.

[14] P. Liu, Y. Huang, L. Meng, S. Gong, and G. Zhang, "Two-stage extreme learning machine for high-dimensional data," *Int. J. Mach. Learn. Cybern.*, vol. 7, no. 5, pp. 765–772, Aug. 2014.

[15] D. Guan, W. Yuan, Y.-K. Lee, K. Najeebullah, and M. K. Rasel, "A review of ensemble learning based feature selection," *IETE Tech. Rev.*, vol. 31, no. 3, pp. 190–198, May 2014.

[16] A. K. Das, S. Das, and A. Ghosh, "Ensemble feature selection using bi-objective genetic algorithm," *Knowl.-Based Syst.*, vol. 123, pp. 116–127, May 2017.

[17] K. Debjit, M. S. Islam, M. A. Rahman, F. T. Pinki, R. D. Nath, S. Al-Ahmadi, M. S. Hossain, K. M. Mumenin, and M. A. Awal, "An improved machine-learning approach for COVID-19 prediction using Harris hawks optimization and feature analysis using SHAP," *Diagnostics*, vol. 12, no. 5, p. 1023, Apr. 2022.

[18] B. Seijo-Pardo, I. Porto-Díaz, V. Bolón-Canedo, and A. Alonso-Betanzos, "Ensemble feature selection: Homogeneous and heterogeneous approaches," *Knowl.-Based Syst.*, vol. 118, pp. 124–139, Feb. 2017.

[19] K. Kumar, "Machine learning-based ensemble approach for predicting the mortality risk of COVID-19 patients: A case study," in *Intelligent Data Analysis for COVID-19 Pandemic* (Algorithms for Intelligent Systems), M. Niranjanamurthy, S. Bhattacharyya, and N. Kumar, Eds., Singapore: Springer, 2021, pp. 1–25.

[20] C. Koushik, R. Bhattacharjee, and C. S. Hemalatha, "Symptoms based early clinical diagnosis of COVID-19 cases using hybrid and ensemble machine learning techniques," in *Proc. 5th Int. Conf. Comput., Commun. Signal Process. (ICCCSP)*, May 2021, pp. 1–6.

[21] S. S. Aljameel, I. U. Khan, N. Aslam, M. Aljabri, and E. S. Alsulmi, "Machine learning-based model to predict the disease severity and outcome in COVID-19 patients," *Sci. Program.*, vol. 2021, pp. 1–10, Apr. 2021.

[22] M. Pourhomayoun and M. Shakibi, "Predicting mortality risk in patients with COVID-19 using artificial intelligence to help medical decision-making," *medRxiv*, Apr. 2020.

[23] Ç. Danacı and S. A. Tuncer, "Incorporating feature selection methods into machine learning-based COVID-19 diagnosis," *Appl. Comput. Syst.*, vol. 27, no. 1, pp. 13–18, Jun. 2022.

[24] Md. A. Awal, M. Masud, M. S. Hossain, A. A. Bulbul, S. M. H. Mahmud, and A. K. Bairagi, "A novel Bayesian optimization-based machine learning framework for COVID-19 detection from inpatient facility data," *IEEE Access*, vol. 9, pp. 10263–10281, 2021.

[25] A. Jafar and M. Lee, "HypGB: High accuracy GB classifier for predicting heart disease with HyperOpt HPO framework and LASSO FS method," *IEEE Access*, vol. 11, pp. 138201–138214, 2023.

[26] H. Jeon and S. Oh, "Hybrid-recursive feature elimination for efficient feature selection," *Appl. Sci.*, vol. 10, no. 9, p. 3211, May 2020.

[27] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "Feature selection for high-dimensional data," in *Artificial Intelligence: Foundations, Theory, and Algorithms*. Cham, Switzerland: Springer, 2015, pp. 13–28.

[28] D. Panda, R. Ray, A. A. Abdullah, and S. R. Dash, "Predictive systems: Role of feature selection in prediction of heart disease," *J. Phys., Conf. Ser.*, vol. 1372, no. 1, Nov. 2019, Art. no. 012074.

[29] R. Tang and X. Zhang, "CART decision tree combined with Boruta feature selection for medical data classification," in *Proc. 5th IEEE Int. Conf. Big Data Anal. (ICBDA)*, May 2020, pp. 80–84.

[30] L. Sun, T. Yin, W. Ding, Y. Qian, and J. Xu, "Multilabel feature selection using ML-ReliefF and neighborhood mutual information for multilabel neighborhood decision systems," *Inf. Sci.*, vol. 537, pp. 401–424, Oct. 2020.

[31] A. Dabba, A. Tari, S. Meftali, and R. Mokhtari, "Gene selection and classification of microarray data method based on mutual information and moth flame algorithm.," *Expert Syst. Appl.*, vol. 166, Mar. 2021, Art. no. 114012.

[32] R. H. Ali and W. H. Abdulsalam, "The prediction of COVID 19 disease using feature selection techniques," *J. Phys., Conf. Ser.*, vol. 1879, no. 2, May 2021, Art. no. 022083.

[33] M. Toğaçar, B. Ergen, and Z. Cömert, "Classification of white blood cells using deep features obtained from convolutional neural network models based on the combination of feature selection methods," *Appl. Soft Comput.*, vol. 97, Dec. 2020, Art. no. 106810.

[34] A. Lopez-Rincon, L. Mendoza-Maldonado, M. Martinez-Archundia, A. Schönhuth, A. D. Kraneveld, J. Garssen, and A. Tonda, "Machine learning-based ensemble recursive feature selection of circulating miRNAs for cancer tumor classification," *Cancers*, vol. 12, no. 7, p. 1785, Jul. 2020.

[35] D. François, F. Rossi, V. Wertz, and M. Verleysen, "Resampling methods for parameter-free and robust feature selection with mutual information," *Neurocomputing*, vol. 70, nos. 7–9, pp. 1276–1288, Mar. 2007.

[36] M. Mokhtia, M. Eftekhari, and F. Saberi-Movahed, "Feature selection based on regularization of sparsity based regression models by hesitant fuzzy correlation," *Appl. Soft Comput.*, vol. 91, Jun. 2020, Art. no. 106255.

[37] X. Xiao, M. Yan, S. Basodi, C. Ji, and Y. Pan, "Efficient hyperparameter optimization in deep learning using a variable length genetic algorithm," 2020, *arXiv:2006.12703*.

[38] A. Jafar and M. Lee, "Comparative performance evaluation of state-of-the-art hyperparameter optimization frameworks," *Trans. Korean Inst. Electr. Eng.*, vol. 72, no. 5, pp. 607–620, May 2023.

[39] E. M. Karabulut, S. A. Özel, and T. Ibrikçi, "A comparative study on the effect of feature selection on classification accuracy," *Proc. Technol.*, vol. 1, pp. 323–327, Jan. 2012.

[40] X. Tan, S. Su, Z. Huang, X. Guo, Z. Zuo, X. Sun, and L. Li, "Wireless sensor networks intrusion detection based on SMOTE and the random forest algorithm," *Sensors*, vol. 19, no. 1, p. 203, Jan. 2019.

[41] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

[42] L. Morán-Fernández and V. Bolón-Canedo, "Finding a needle in a haystack: Insights on feature selection for classification tasks," *J. Intell. Inf. Syst.*, vol. 62, no. 2, pp. 459–483, Nov. 2023.

[43] L. Xu, R. Wang, F. Nie, and X. Li, "Efficient top-K feature selection using coordinate descent method," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2023, pp. 10594–10601.

[44] S. Deng, Y. Li, J. Wang, R. Cao, and M. Li, "A feature-thresholds guided genetic algorithm based on a multi-objective feature scoring method for high-dimensional feature selection," *Appl. Soft Comput.*, vol. 148, Nov. 2023, Art. no. 110765.

[45] L. J. Muhammad, E. A. Algehyne, S. S. Usman, A. Ahmad, C. Chakraborty, and I. A. Mohammed, "Supervised machine learning models for prediction of COVID-19 infection using epidemiology dataset," *Social Netw. Comput. Sci.*, vol. 2, no. 1, p. 11, Feb. 2021.

[46] M. Rostami and M. Oussalah, "A novel explainable COVID-19 diagnosis method by integration of feature selection with random forest," *Informat. Med. Unlocked*, vol. 30, Jan. 2022, Art. no. 100941.

[47] E. Casiraghi, D. Malchiodi, G. Trucco, M. Frasca, L. Cappelletti, T. Fontana, A. A. Esposito, E. Avola, A. Jachetti, J. Reese, A. Rizzi, P. N. Robinson, and G. Valentini, "Explainable machine learning for early assessment of COVID-19 risk prediction in emergency departments," *IEEE Access*, vol. 8, pp. 196299–196325, 2020.
[48] F. Prinzi, C. Militello, N. Scichilone, S. Gaglio, and S. Vitabile, "Explainable machine-learning models for COVID-19 prognosis prediction using clinical, laboratory and radiomic features," *IEEE Access*, vol. 11, pp. 121492–121510, 2023.
[49] Y. Zoabi, S. Deri-Rozov, and N. Shomron, "Machine learning-based prediction of COVID-19 diagnosis based on symptoms," *NPJ Digit. Med.*, vol. 4, no. 3, 2021, doi: 10.1038/s41746-020-00372-6.
[50] M. J. Hossen, T. T. Ramanathan, and A. Al Mamun, "An ensemble feature selection approach-based machine learning classifiers for prediction of COVID-19 disease," *Int. J. Telemedicine Appl.*, vol. 2024, pp. 1–10, Apr. 2024.
[51] Y. Zoabi, S. Deri-Rozov, and N. Shomron, "Machine learning-based prediction of COVID-19 diagnosis based on symptoms," *NPJ Digit. Med.*, vol. 4, no. 1, pp. 1–5, Jan. 2021.

**ABBAS JAFAR** received the B.S. degree in software engineering from the Government College University Faisalabad, Pakistan, the master's degree from Myongji University, South Korea, in 2018, and the master's degree in 2020. He is currently pursuing the Ph.D. degree. He is currently working as a Research Assistant with the HPC Laboratory, Myongji University. His research interests include AI in healthcare systems, machine learning, deep healthcare, high-performance computing, and performance optimization, with a particular interest in GPU computing.

**MYUNGHO LEE** (Member, IEEE) received the B.S. degree in computer science and statistics from Seoul National University, South Korea, the M.S. degree in computer science, and the Ph.D. degree in computer engineering from the University of Southern California, USA. He was a Staff Engineer at the Scalable Systems Group, Sun Microsystems, Sunnyvale, CA, USA. He is currently working as a Full Professor with the Department of Computer Science and Engineering, Myongji University. His research interest includes high-performance computing: architecture, compilers, and applications.