Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Equilibrium-based COVID-19 diagnosis from routine blood tests: A sparse deep convolutional model

Doaa A. Altantawy [*,1], Sherif S. Kishk [2]

Electronics and Communications Engineering Department, Faculty of Engineering, Mansoura University, 60 El-Gomhoria Street, Mansoura, Egypt

ARTICLE INFO

Keywords:
COVID-19
Blood tests
1DCNN
Feature pool sparsification
Equilibrium optimization
Feature selection
Pearson correlation

ABSTRACT

SARS-CoV2 (COVID-19) is the virus that causes the pandemic that has severely impacted human society with a massive death toll worldwide. Hence, there is a persistent need for fast and reliable automatic tools to help health teams in making clinical decisions. Predictive models could potentially ease the strain on healthcare systems by early and reliable screening of COVID-19 patients which helps to combat the spread of the disease. Recent studies have reported some key advantages of employing routine blood tests for initial screening of COVID-19 patients. Thus, in this paper, we propose a novel COVID-19 prediction model based on routine blood tests. In this model, we depend on exploiting the real dependency among the employed feature pool by a sparsification procedure. In this sparse domain, a hybrid feature selection mechanism is proposed. This mechanism fuses the selected features from two perspectives, the first is Pearson correlation and the second is a new Minkowski-based equilibrium optimizer (MEO). Then, the selected features are fed into a new 1D Convolutional Neural Network (1DCNN) for a final diagnosis decision. The proposed prediction model is tested with a new public dataset from San Raphael Hospital, Milan, Italy, i.e., OSR dataset which has two sub-datasets. According to the experimental results, the proposed model outperforms the state-of-the-art techniques with an average testing accuracy of 98.5% while we employ only less than half the size of the feature pool, i.e., we need only less than half the given blood tests in the employed dataset to get a final diagnosis decision.

1. Introduction

COVID-19 pandemic is the contemporary element of worriment across the world. This pandemic is caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) which is found to have a high degree of spread causing a massive death toll. The COVID-19 infection caused clusters of fatal pneumonia with clinical presentation greatly resembling SARS-CoV. In fact, patients experience flu-like symptoms, such as, fever, dry cough, tiredness, and difficulty breathing. However, sometimes, in more severe cases, pneumonia and renal failure develop to death (Huang et al., 2020). By, now, June 2022, more than 500 million confirmed patients have been reported in 222 countries with more than 6 million deaths due to this pandemic (Worldometer, 2020). Fig. 1 indicates the logarithmic modeling of death rate all over the world. Hence, A timely detection and diagnosis of the virus plays a leading role in infection control and accordingly in the death rate reduction. Therefore, developing efficient testing methods to identify COVID-19 infection is a must, in order to start early treatment, and to isolate the infected individuals from the rest.

Polymer chain reaction (PCR) (Zimmermann & Mannhalter, 1996; Corman et al., 2020), and Antibody testing (Serological testing) are the two main testing methods adopted by the global healthcare systems for COVID-19 diagnosis, however, both methods have their own limitations. Despite being the current gold standard for infection diagnosis, PCR has limitations in terms of resources and specimen collection (Ai et al., 2020), besides high cost. In addition, PCR, generally, has high specificity, but low sensitivity with about 20 % false-negative rate (Ferrari et al., 2020; Li D. et al., 2020). Thus, PCR negative test does not negate the possibility of COVID-19, hence, those patients will not receive the appropriate treatment on time. Moreover, there is a global shortage of the availability of PCR test kits. On the other side, tests based on IgM/IgG antibodies have shown a very low sensitivity (18.8 %) and specificity (77.8 %) in diagnosing COVID-19 during its early phase (Burog et al., 2020; Sethuraman et al., 2020). Accordingly, imaging-based

[*] Corresponding author.
E-mail addresses: doaa1adel@mans.edu.eg (D.A. Altantawy), shkishk@mans.edu.eg (S.S. Kishk).
[1] ORCID: https://orcid.org/0000-0001-6476-2934.
[2] ORCID: https://orcid.org/0000-0001-8043-2011.

https://doi.org/10.1016/j.eswa.2022.118935
Received 24 January 2022; Received in revised form 21 June 2022; Accepted 27 September 2022
Available online 3 October 2022

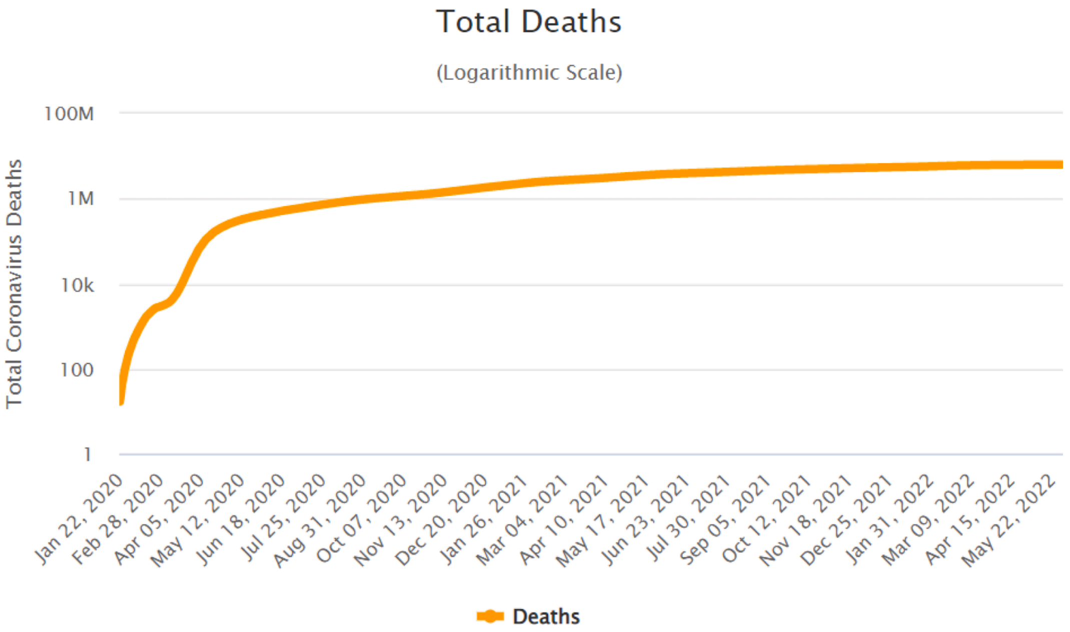


Fig. 1. A logarithmic scale for COVID-19 Monthly total deaths (Worldometer, 2020).

diagnosis methods, such as Chest Radiograph images (CXRs)/ X-rays, computerized tomography (CT) scan, MRI and Ultrasound, besides other laboratory methods, such as routine blood test, can be employed to define the severity of the illness caused by COVID-19.

Till now, COVID-19 pandemic continues challenging the world with the increase demands of hospital beds and medical equipments, especially with the everyday variations of the virus and with the exhausted healthcare workers. This has prompted researchers to investigate alternative automated methods with accurate and fast detection, less expensive, more accessible, and with minimal human interference. Over the years, machine learning (ML) field has gained much popularity for solving numerous real-world problems by producing systems that are capable of learning from examples and improving without being explicitly programmed (Brink et al., 2016). Hence, ML-based approaches have been used in the screening of patients suspected of being contaminated by SARS-CoV2, supporting the medical decision (Alballa & Al-Turaiki, 2021). Lately, several outbreak prediction models for COVID-19 have been developed to make informed-decisions and enforce relevant control measures (Albahri et al., 2020; Bullock et al, 2020; Latif et al., 2020; Alafif et al., 2021). However, due to a high level of uncertainty and lack of essential data, diagnosing COVID-19 by machine learning and soft computing models is still challenging research area.

In this work, we introduce a new COVID-19 detection model based on routine blood tests, see Fig. 1. The main contributions can be summarized as

1. Seeking an optimum dimensionality reduction, besides exploiting the real dependency among features in the adopted feature pool, a sparsification procedure is adopted. Hence, the introduced feature selection techniques can perform better in the discovered sparse domain. This sparsification procedure is performed by a sparse and low-rank decomposition process. The resultant sparse composite of the feature pool is expected to provide features with few pairwise interactions.
2. For more effective feature selection performance, the adopted selection mechanism fuses the selection decisions from a statistical perspective on a side, i.e., Pearson correlation, and from a wrapper perspective, on the other side, i.e., Equilibrium Optimizer (EO) (Faramarzi et al., 2020).
3. Instead of applying the traditional EO in the adopted feature selection procedure, the introduced diagnosis algorithm adopts a new Minkowski-based equilibrium optimizer (MEO) which employs a Minkowski-based scheme for local minimum avoidance, besides a recycling strategy for the worst solutions in order to find the most proper features, i.e., blood tests.
4. For the classification phase, the proposed COVID-19 diagnosis algorithm adopts a 1DCNN model which shows superior performance compared to multiple traditional ML algorithms.
5. The introduced COVID-19 diagnosis algorithm outperforms the state-of-the-art prediction model on all metrics, that are based on routine blood tests, while employing only less than half the size of the feature pool which means less blood tests and less cost which suits the conditions in the developing countries.

The rest of the paper is organized as following: Section 2 indicates related work. In Section 3, details about the employed routine blood test dataset are indicated. In Section 4, the whole proposed methodology is introduced and detailed in some subsections. Section 5 indicates the experimental results with proper discussions. In Section 6, the conclusion is demonstrated.

2. Related work

Machine learning (ML) is a key branch of computational algorithms that are designed to imitate human intelligence by an automatic learning from the surrounding environment. Hence, the machine takes decisions and does predictions / forecasting based on data ML is one of today's most rapidly growing technical topics, lying at the intersection of computer science and statistics, and at the core of artificial intelligence and data science. ML is considered the working horse in the new era of the so-called big data. Different machine learning techniques have been applied successfully in diverse fields, such as, from wireless communications (Tan et al., 2014), computer vision (Khan et al., 2021; Altantawy et al., 2020), finance (Kumbure et al., 2022), entertainment (Porcino et al., 2022), control system (Hedrea and Petriu, 2021) and computational biology to biomedical and medical applications (Chiang et al., 2014; Albu et al., 2019; Upadhyay & Nagpal, 2020).

ML can be used to combat COVID-19 pandemic by improving diagnosis, prevention, monitoring, administration of treatments, disease

Table 1
Comparison of different COVID-19 detection based on routine blood tests.

Authors/ref.	Dataset source	$\frac{N_D/N_+}{N_F/N_{SF}}$ [*1]	Adopted methodology	Accuracy	Sensitivity	Specificity	ROC-AUC
Soares, 2020	Hospital Israelita Albert Einstein, São Paulo, Brazil	$\frac{599/81}{108/16}$	SMOTEBoost, Ensemble of 10 SVM models	–	70.25 %	85.98 %	86.78 %
Banerjee et al., 2020		$\frac{598/81}{108/14}$	RF, LR, GLMNET, ANN	81 %–87 %	43 %–65 %	81 %–91 %	80 %-84 %
de Moraes et al., 2020		$\frac{253/102}{108/15}$	NN, RF, GBT, LR, SVM	–	67.7 %–80.6 %	80 %–85 %	84.2 %–84.7 %
Alves et al., 2021		$\frac{524/48}{108/23}$	DTX, RF, Ensemble of LR, RF, XGBoost, SVM, MLP	88 %	66 %	91 %	86 %
Alakus & Turkoglu, 2020		$\frac{520/80}{108/18}$	Ensemble of ANN, CNN LSTM, RNN CNNLSTM CNNRNN	86.66 %	–	–	62.50 %
de Freitas Barbosa et al., 2021		$\frac{5644/559}{108/24}$	XMLP, SVM, RT, RF, BN, NB	95.159 %	96.8 %	93.6 %	——
AlJame et al., 2020		$\frac{5644/559}{108/18}$	KNNimputer, iForest, SMOTE, Ensemble of RF, LR, and ET	95 %	95 %	95 %	95 %
Wu et al., 2020	Tongji Hospital of Wuhan, China	$\frac{110}{47/7}$	LASSO-LR	——	98 %	91 %	0·997
Yan et al., 2020		$\frac{375/201}{300/3}$	XGBoost	–	83 %	–	———
Cabitza et al., 2021	San Raphael Hospital, Milan, Italy	$\frac{1,624/845}{72}$	RF, NB, LR, SVM, and KNN	83 %–91 %	76 %–92 %	92 %–96 %	83 % − 94 %
Brinati et al., 2020		$\frac{279/177}{13}$	DT, ET, KNN, LR, NB, RF, SVM, TWRF	82 % −86 %	92 % − 95 %	–	–
Shaban et al., 2021		$\frac{279/177}{13}$	FI, DNN	97.658 %	96.55 %	–	–
Yang et al., 2020	New York Presbyterian Hospital/Weill Cornell Medicine (NYPH/WCM)	$\frac{1,822/496}{685/33}$	LR, DT, RF, XGBoost	68.9 %–79.1 %	61.8 % −76.1 %	73.2 %–80.8 %	70.4 %–85.4 %
Joshi et al., 2020	Stanford Health Care, CA, USA	$\frac{390/33}{4}$	LR	–	86–93 %	35–55 %	–
Sun et al., 2020	Hospitals in Zhejiang, China	$\frac{912/361}{31/10}$	LR, DT, RF, SVM. DNN	91 %	87 %	95 %	86.4 %
Langer et al., 2020	Hospital in Milan Italy	$\frac{199/127}{74/42}$	ANN, LR, RF, DT	91.4 %	94.1 %	88.7 %	–
Kukar et al., 2021	University Medical Center, Ljubljana, Slovenia	$\frac{5333/160}{117/35}$	XGBoost, RF, DNN	–	81.9 %	97.9 %	97 %

[*1] N_D is the dataset size, N_+ is the number of COVID-19 positive cases in the employed dataset, N_F is the total number of features in the targeted dataset, and N_{SF} is the number of the selected features in the diagnosis process. Using "–", means not mentioned in the original study.

surveillance and antiviral drug discovery to enhance patients' health outcomes (Bullock et al, 2020; Latif et al., 2020; Alafif et al., 2021). Since the beginning of COVID-19 outbreak, there has been a growing interest in studying the diagnosis of COVID-19, either through the analysis of medical images (Albahri et al, 2020) or routine blood tests (Cabitza et al., 2021) by different ML techniques. These alternative diagnosing methods are less expensive and more accessible. In this section, we review some of these ML-based studies. The diagnosis of COVID-19 in ML terms can be formulated as Binary classification problem, hence, with the trained model, patients can be classified positive or negative COVID-19 or sometimes patients can be checked for the severity of illness (Albahri et al, 2020).

Medical imaging, such as computed tomography (CT) scans and chest X-rays images, are the main two types of datasets that have been employed by different ML techniques and have demonstrated promising results to support the traditional diagnostic techniques of COVID-19, such as molecular biology (RT-PCR) and immune (IgM/IgG) assays. There have been several recent reviews with exclusive focus on X-rays or CT scans (Albahri et al., 2020; Latif et al., 2020; Alafif et al., 2021). Several studies observed that the sensitivity of CT in diagnosing COVID-19 is significantly higher than that of RT-PCR (Ai et al., 2020; fang et al., 2020; Ye et al., 2020). However, CT scans have screening limitations because of the radiation doses, the relative low number of devices available, and the related high costs. In addition, by employing X-rays or CT scans only, COVID-19 can be mistakenly diagnosed as pneumonia or lung cancer (Ibrahim et al., 2021; Mohammad-Rahimi et al., 2021). Recently, in (Dong et al., 2020), some researchers have employed ultrasound imaging as a radiation-free and non-invasive tool for COVID-19 detection, especially for children and pregnant women. Other research groups have explored the opportunities of employing speech and sound analysis for a ML-based COVID-19 detection (Imran et al., 2020; Schuller et al., 2021). In (Zoabi et al., 2021), the authors tried a ML-based prediction of COVID-19 diagnosis based on symptoms.

Recently, different studies have revealed that a routine blood test can play an important role in COVID-19 initial screening (Bao et al., 2020; Gao et al., 2020; Ferrari et al., 2020). Hence, a routine blood test can provide faster and cheaper diagnostic alternative to PCR test with comparable performance via different ML techniques (Brinati et al., 2020; Cabitza et al., 2021). In (Wu, J. et al., 2020), the authors are the pioneers of employing blood results in COVID-19 detection. They utilized a ML algorithm of three stages based on a random forest classification algorithm with several different validation methods to ensure the reliability and reproducibility of their COVID-19 identification algorithm. They achieved high accuracy ~98 %, but the model considered few features, and the dataset is very small to be applicable in real settings. In (Wu et al., 2020; Yan et al., 2020), they employed datasets with different sizes from Tongji Hospital of Wuhan, China. Wu et al. (2020) achieved higher accuracy with smaller-size dataset with larger number of selected features. They build their model based on the maximum relevance minimum redundancy algorithm (mRMR), the least absolute shrinkage (LA) and LASSO logistic regression model. On the other side, (Yan et al., 2020) employed larger-size dataset with larger number of features. However, they selected small group of these features and achieved lower accuracy using a trained model based on XGBoost algorithm. In Feng et al. (2021), the authors continued employing small dataset from single source, i.e., First Medical Center, Beijing, China. However, they developed an innovative predictive model for an early identification of COVID-19 based on candidate features included clinical

Table 2
List of abbreviations.

Abbreviation	Explanation	Abbreviation	Explanation
ML	Machine learning	PCR	Polymer chain reaction
mRMR	maximum relevance minimum redundancy algorithm	SMOTEBoost	an oversampling method based on the SMOTE algorithm (Synthetic Minority Oversampling Technique)
SVM	Support vector machine	RF	Random Forest
LR	Logistic regression	GLMNET	Lasso and Elastic-Net Regularized Generalized Linear Models
ANN/NN	Artificial neural network	DNN	Deep neural network
GBT	Gradient boosting trees	XGBoost	is an optimized distributed gradient boosting library
MLP	Multi-layer perceptron	CNN	Convolutional neural network
LSTM	Long short-term memory (LSTM) is an artificial recurrent neural network (RNN)	NB	Naïve bayes
BN	Bayesian network	iForest	Isolation forest
LASSO	least absolute shrinkage and selection operator	KNN	k-nearest neighbors algorithm
TWRF	Trees Weighting Random Forest	FI	Fuzzy inference
DT	Decision Tree	GNB	Gaussian Naïve Bayes
ET	Extremely Randomized Trees	RSVM	Radial Support Vector Machine
LSVM	Linear Support Vector Machine	QDA	Quadratic Discriminant Analysis
LDA	Linear Discriminant Analysis	EO	Equilibrium optimizer
AdaBoost	Adaptive Boosting trees	MEO	Minkowski-based equilibrium optimizer

symptoms, routine laboratory tests, and other clinical information on admission. They employed LA and LASSO in their prediction model. They produce their own website for COVID-19 diagnosis (Suspected COVID-19 pneumonia Diagnosis Aid System, 2021). In addition, there exist different studies that employed their own dataset from single source, i.e., medical Centre or hospital, with different ML techniques (Joshi et al., 2020; Kukar et al., 2021; Li et al., 2020; Yang et al., 2020; Langer et al., 2020; Sun et al., 2020).

Hospital Israelita Albert Einstein, São Paulo, Brazil has provided a common blood test dataset that has been utilized by different studies, such as (Alakus & Turkoglu, 2020; AlJame et al., 2020; Banerjee et al., 2020; de Moraes et al., 2020; Soares, 2020; Alves et al., 2021; de Freitas Barbosa et al., 2021). In (Alakus & Turkoglu, 2020; Banerjee et al., 2020; de Moraes et al., 2020; Soares, 2020; Alves et al., 2021), the authors employed small version of the original whole dataset. They achieved a medium accuracy, however, with very small number of selected features, based on applying multiple ML algorithms. de Freitas Barbosa et al., 2021 and AlJame et al., 2020 employed the original full dataset. Both achieved high accuracy via multiple ML algorithms employing large number of selected features. Lately, AlJame et al. (2020) revealed better performance via three well-known classifiers, Extremely Randomized Trees, Random Forest and Logistic regression. Their model combines the predictions of three classifiers, as a first level classification, then, they used an extreme gradient boosting (XGBoost), as a second classifier, to achieve a better performance. San Raphael Hospital, Milan, Italy has provided the most recent dataset (OSR dataset), which adopted by different studies under different sizes with different feature mechanisms and variety of ML algorithms (Brinati et al., 2020; Cabitza et al., 2021; Shaban et al., 2021). Table 1 summarizes a comparison between the state-of-the-art techniques, while Table 2 summarizes the list of abbreviations employed in this article.

3. The employed routine blood test dataset

Here, the employed dataset for COVID-19 prediction is routine blood-test results performed on group of patients on admission to the ED department at the San Raffaele Hospital, *ospedale San Raffaele,* (OSR), from February 19, 2020, to May 31, 2020. The OSR dataset consists of two subgroups (Brinati et al., 2020; Cabitza et al., 2021) with different sample size and different blood features. (1) A larger sub-dataset consisting of 1736 sample with 35 features, named as "COVID-specific dataset".[3] (2) A smaller one consisting of 279 sample with 15 features, denoted as "CBC dataset".[4] The features set existed in OSR dataset is detailed in Table 3. These features represent the numerical ones besides additional ones, like gender, age, and ID number. We always exclude ID number before processing. In Fig. 3, the label distribution, i.e., swab result, of both employed sub-datasets are indicated. In addition, Fig. 4 indicates the distribution of COVID-19 examination results over the age and the gender of the samples.

4. The proposed methodology

In this section, the proposed COVID-19 detection algorithm, as a binary classification problem, is detailed in some subsections, see Fig. 2 where an illustration of the proposed COVID-19 detection algorithm is indicated. In the first and the second subsections, the dataset preparation and feature pool sparsification are demonstrated. In the third one, the proposed feature selection scheme is indicated and finally, in the last subsection, the deep classification model is proposed.

4.1. Dataset preparation

The process of data preparation includes four stages: handling categorical features, handling missing values, outliers detection and elimination, and data balancing.

Handling categorical features: The only categorical features in the OSR datasets are the gender and the covid exam result. Hence, both are mapped to 0 and 1.

Handling missing values: firstly, the samples that have more than 75 % of its features missed are excluded. Secondly, to address data incompleteness, we performed missing data imputation by k-nearest neighbors using the mean value from nearest neighbors. KNN algorithm is useful for matching a data-point with its closest k neighbors in a multi-dimensional space.

Outliers detection and elimination: outliers elimination helps to increase the accuracy of the classification model. Clustering-based approaches (Borlea et al., 2021) can be used for outlier detection (Zhang et al., 2021). However, for detecting anomalies in the adopted OSR dataset, we employed a tree-based approach, i.e., Isolation Forests algorithm (Liu et al., 2008). Isolation Forests (IF or iForest), like Random Forests, are build based on decision trees. It has no pre-defined labels. Hence, it is an unsupervised model like most of outlier detection algorithms. iForest is based on the fact that anomalies are "few and different". In iForest, randomly sub-sampled data is processed in a tree structure based on randomly selected features. The samples that travel deeper into the tree are less likely to be anomalies as they required more cuts to isolate them. Similarly, the samples which end up in shorter branches indicate anomalies as it was easier for the tree to separate them from other observations. We chose iForest as an outlier detection

[3] https://zenodo.org/record/4081318/files/all_training.xlsx?download=1.

[4] https://zenodo.org/record/3886927/files/covid_study_v2.xlsx?download=1.

Table 3
The numerical features in the OSR dataset with its mean value μ, standard deviation σ and missing rate..*MR*

Feature (Abb.)	Description	COVID-specific dataset				CBC dataset			
		Exist.	MR %	μ	σ	Exist.	MR %	μ	σ
Calcium (CA)	A test checks the calcium level in the body that is not stored in the bones	✓	5.35	2.21	0.48	×			
Creatine kinase (CK)	This test measures the amount of an enzyme called creatine kinase (CK) in your blood. CK is a type of protein. The muscle cells in your body need CK to function.	✓	59.44	181.64	405.71	×			
Creatinine (CREA)	A test measures how well your kidneys are performing their job of filtering waste from your blood	✓	4.26	1.16	0.98	×			
Alkaline phosphatase (ALP)	ALP is an enzyme found throughout the body, but it is mostly found in the liver, bones, kidneys, and digestive system. When the liver is damaged, ALP may leak into the bloodstream	✓	27.3	88.54	71.44	✓	53	89.89	89.09
Gamma glutamyl transferase (GGT)	A test assess the body response to glucose	✓	25.11	66.22	135.39	✓	51.25	82.48	132.70
Glucose (GLU)	A test measures the level of glucose (sugar) in a person's blood	✓	5.65	119	57.91	×			
Aspartate aminotransferase (AST)	AST is an enzyme that is normally present in the liver, heart, brain, pancreas, kidneys, and many other muscles and tissues in the body. Enzymes like AST help facilitate fundamental biological processes in these organs and tissues	✓	5.65	45.85	50.67	✓	0.72	54.20	57.61
Alanine aminotransferase (ALT)	A test measures the amount of ALT in the blood. High levels of ALT in the blood can indicate a liver problem, even before you have signs of liver disease, such as jaundice, a condition that causes your skin and eyes to turn yellow. An ALT blood test may be helpful in early detection of liver disease	✓	5.53	39.17	42.55	✓	4.66	44.92	45.50
Lactate dehydrogenase (LDH)	A test looks for signs of damage to the body's tissues. LDH is an enzyme found in almost every cell of your body, including your blood, muscles, brain, kidneys, and pancreas. The enzyme turns sugar into energy	✓	17.45	327.64	211.62	✓	30.47	380.45	193.98
polymerase chain reaction (CRP)	A test measures the amount of CRP in the blood to detect inflammation due to acute conditions or to monitor the severity of disease in chronic conditions	✓	5.59	67	77.8	✓	2.15	90.88	94.4
Potassium (K)	A test checks how much potassium is in the blood	✓	4.61	4.23	0.52	×			
Sodium (NA)	checks how much sodium is in the blood	✓	4.21	138.59	4.58	×			
UREA	Urea is usually passed out in the urine. A high blood level of urea indicates that the kidneys may not be working properly, or that you have a low body water content (are dehydrated)	✓	38.94	48.96	42.47	×			
White blood cell (WBC)	A test measures the count of White blood cells	✓	3.63	8.72	4.64	✓	0.72	8.55	4.86
Red blood cell (RBC)	A test measures the count of Red blood cells	✓	3.63	4.52	0.73	×			
Hemoglobin (HGB)	a protein in your red blood cells that carries oxygen to your body's organs and tissues and transports carbon dioxide from your organs and tissues back to your lungs	✓	3.63	13.14	2.04	×			
Hematocrit (HCT)	A test measures the proportion of red blood cells in your blood. Red blood cells carry oxygen throughout your body. Having too few or too many red blood cells can be a sign of certain diseases	✓	3.63	39.21	5.61	×			
Mean corpuscular volume (MCV)	There are three main types of corpuscles (blood cells) in your blood: red blood cells, white blood cells, and	✓	3.63	87.29	7.06	×			

(*continued on next page*)

Table 3 (*continued*)

Feature (Abb.)	Description	COVID-specific dataset				CBC dataset			
		Exist.	MR %	μ	σ	Exist.	MR %	μ	σ
	platelets. An MCV blood test measures the average size of your red blood cells								
Mean corpuscular hemoglobin (MCH)	It's the average amount in each of your red blood cells of a protein called hemoglobin, which carries oxygen around your body	✓	3.63	29.21	2.72	×			
Mean corpuscular hemoglobin concentration (MCHC)	A test checks the average amount of hemoglobin in a group of red blood cells	✓	3.63	33.45	1.34	×			
Platelets (PLT)	A normal platelet count ranges from 150,000 to 450,000 platelets per microliter of blood	✓	3.63	235.66	94.22	✓	0.72	226.53	101.17
Neutrophils (NET, NE)	a type of white blood cell that helps heal damaged tissues and resolve infections ($10^9/L$, %)	(✓,✓)	(20.85, 20.85)	(6.45, 72.35)	(4.47, 13.26)	(✓,×)	(25.1, ———)	(6.2, ——)	(4.17, ——)
Lymphocytes (LYT, LY)	are a type of white blood cell. They play an important role in your immune system, helping your body fight off infection ($10^9/L$, %)	(✓,✓)	(20.85, 20.85)	(1.37, 18.58)	(0.95, 11)	(✓,×)	(25.1, ———)	(1.18, ———)	(0.81, ——)
Monocytes (MOT, MO)	are a measurement of a particular type of white blood cell. Monocytes are helpful at fighting infections and diseases ($10^9/L$, %)	(✓,✓)	(20.85, 20.85)	(0.62, 7.83)	(0.54, 3.88)	(✓,×)	(25.1, ———)	(0.61, ———)	(0.41, ———)
Eosinophils (EOT, EO)	are a type of disease-fighting white blood cell. This condition most often indicates a parasitic infection, an allergic reaction or cancer ($10^9/L$, %)	(✓,✓)	(20.85, 20.85)	(0.07, 0.88)	(0.14, 1.62)	(✓,×)	(25.1, ——)	(0.06, ———)	(0.13, ——)
Basophils (BAT, BA)	are a type of white blood cell. Like most types of white blood cells, basophils are responsible for fighting fungal or bacterial infections and viruses ($10^9/L$, %)	(✓,✓)	(20.85, 20.85)	(0.02,0.34)	(0.04,0.27)	(✓,×)	(25.45, ——)	(0.01, ———)	(0.04, ——)

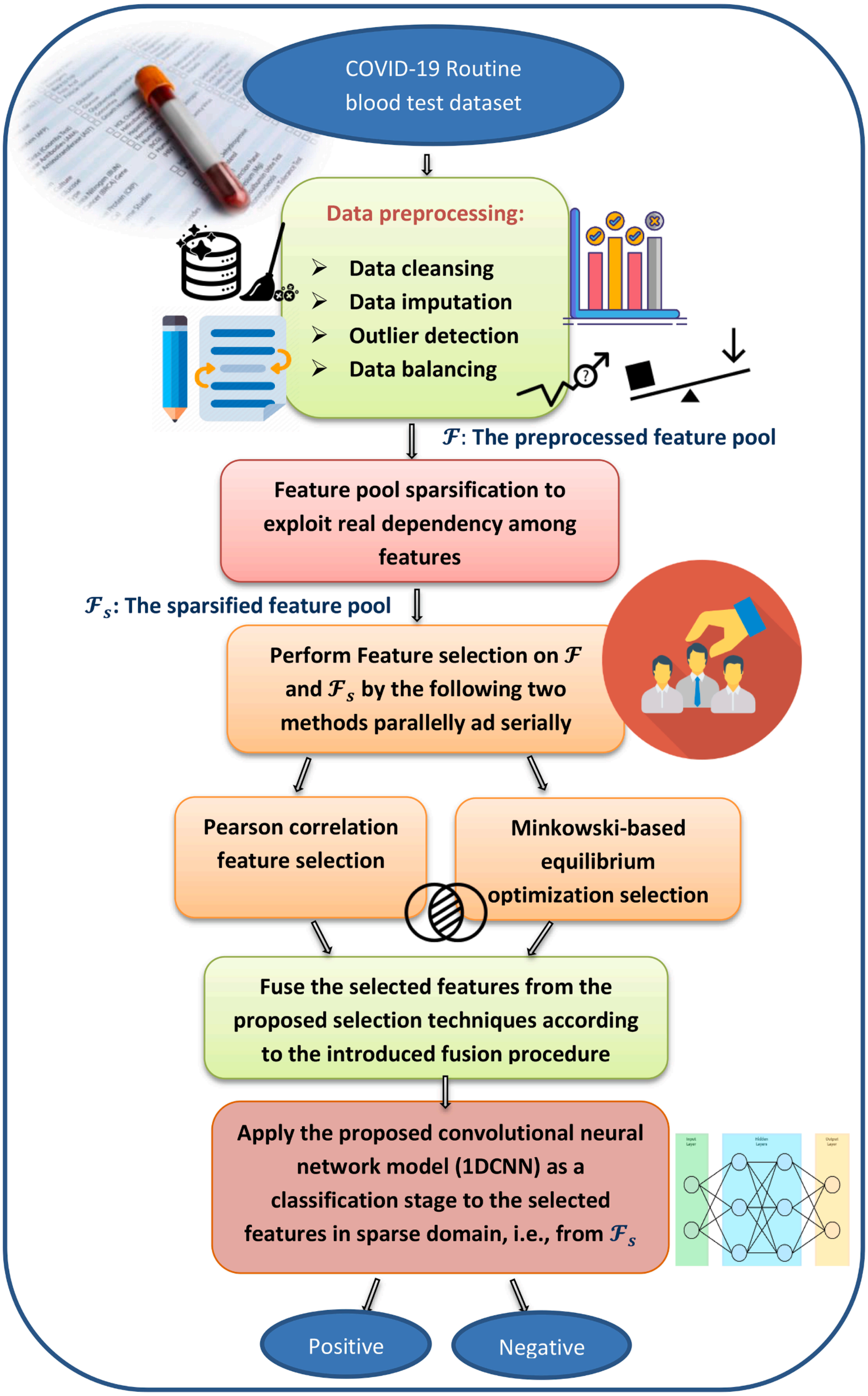

COVID-19 Routine blood test dataset
Data preprocessing:
Data cleansing
Data imputation
Outlier detection
Data balancing
$\mathcal{F}$: The preprocessed feature pool
Feature pool sparsification to exploit real dependency among features
$\mathcal{F}_s$: The sparsified feature pool
Perform Feature selection on $\mathcal{F}$ and $\mathcal{F}_s$ by the following two methods parallelly ad serially
Pearson correlation feature selection
Minkowski-based equilibrium optimization selection
Fuse the selected features from the proposed selection techniques according to the introduced fusion procedure
Apply the proposed convolutional neural network model (1DCNN) as a classification stage to the selected features in sparse domain, i.e., from $\mathcal{F}_s$
Input Layer
Hidden Layers
Output Layer
Positive
Negative

Fig. 2. An illustration of the proposed COVID-19 prediction model.

method because it employs no distance or density measures to detect anomalies which eliminate most of the computational cost of distance calculation in all distance-based and density-based outlier detection algorithms. In addition, iForest has a linear time complexity with a low constant and a low memory requirement, hence it can handle extremely large data size. Fig. 5 indicates the visual results of the detected outliers in COVID-specific dataset via visualizing 3 PCA components.

Data balancing: Having unbalanced data, where the number of samples belonging to one class is significantly lower than those belonging to the other classes, might bias the classification to the majority class. Hence, we performed a synthetic balancing by using Synthetic Minority Oversampling TEchnique (SMOTE) (Chawla et al., 2002). SMOTE is an oversampling technique for generating synthetic samples from the minority class. SMOTE uses linear combinations of two similar samples to construct new data.

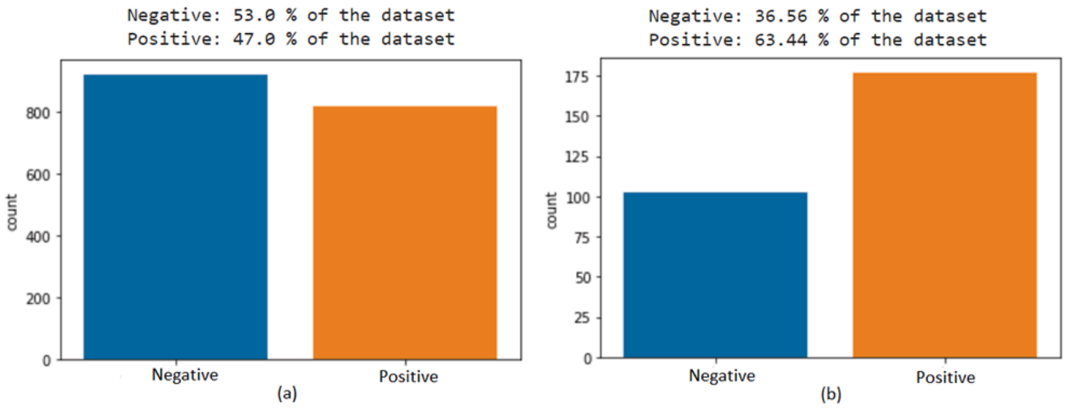


Fig. 3. COVID-19 examination results for COVID-specific dataset in (a) and for CBC dataset in (b).

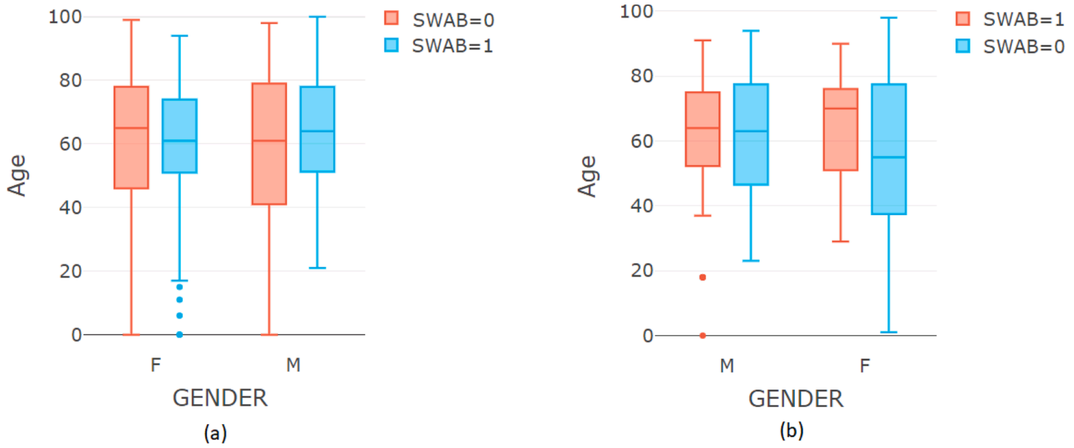


Fig. 4. COVID-19 swab result distribution according to age and gender for COVID-specific dataset in (a) and for CBC dataset in (b).

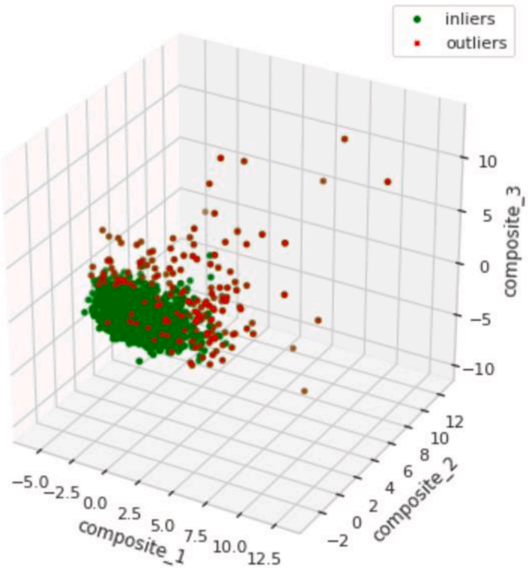


Fig. 5. 3D Visualization of the predicted outliers/inliers in COVID-specific dataset via three PCA components.

4.2. Feature pool sparsification

In order to exploit the real dependency among features in the preprocessed feature pool $\mathscr{F}$ to have a better prediction for the most important features (blood exams), we propose a novel idea of sparsifying the feature pool, i.e., representations which are sparse or of low redundancy. After the sparsification process, we now have two versions of the feature pool, i.e., the original preprocessed feature pool $\mathscr{F}$ and its corresponding sparse one $\mathscr{F}_S$, which are needed in the upcoming feature selection algorithm.

The idea of features sparsification is about decomposing the feature pool to low-rank feature pool and sparse feature pool. Hence, consider having a feature pool $\mathscr{F} = [f_1, f_2, \cdots, f_k]$, where $f_i \in \mathscr{R}^n$ is the i^{th} feature vector for n samples. This feature pool can be assumed neither sparse nor low rank. Hence, its low-rank and sparse structure can be explored by either approximation or decomposition. Robust Principal Component Analysis (RPCA) (Candès et al., 2011) offers a blind separation of low-rank data and sparse noises, i.e., $\mathscr{F} = \mathscr{F}_S + \mathscr{F}_L$, where $\mathscr{F}_L$ is the low-rank component of the feature pool $\mathscr{F}$, while $\mathscr{F}_S$ is the sparse one. Hence, RPCA deals with the targeted sparse component as noise or unwanted part. Hence, we seek for trilateral decomposition, i.e., $\mathscr{F} = \mathscr{F}_S + \mathscr{F}_L + \mathscr{F}_N$, where $\mathscr{F}_N$ is the noise part contaminating the feature pool. This problem is intrinsically different from RPCA. Different studies introduce different styles for a trilateral decomposition of signals for different purposes, such as the work in (Zhou and Tao, 2011; Bouwmans et al., 2017; Altantawy et al., 2020).

Seeking sparse features, the feature pool can be decomposed in terms of Low-rank and sparse components as

$$\mathscr{F} = \mathscr{F}_S + \mathscr{F}_L + \mathscr{F}_N, rank(\mathscr{F}_L) \le \zeta, card(\mathscr{F}_S) \le \Psi \quad (1)$$

$\mathscr{F}_L$ is a tight rank- ζ approximation to the feature pool $\mathscr{F}$, and $\mathscr{F}_S$ has a cardinality of no more than Ψ. The decomposition problem can be solved by minimizing the decomposition error as

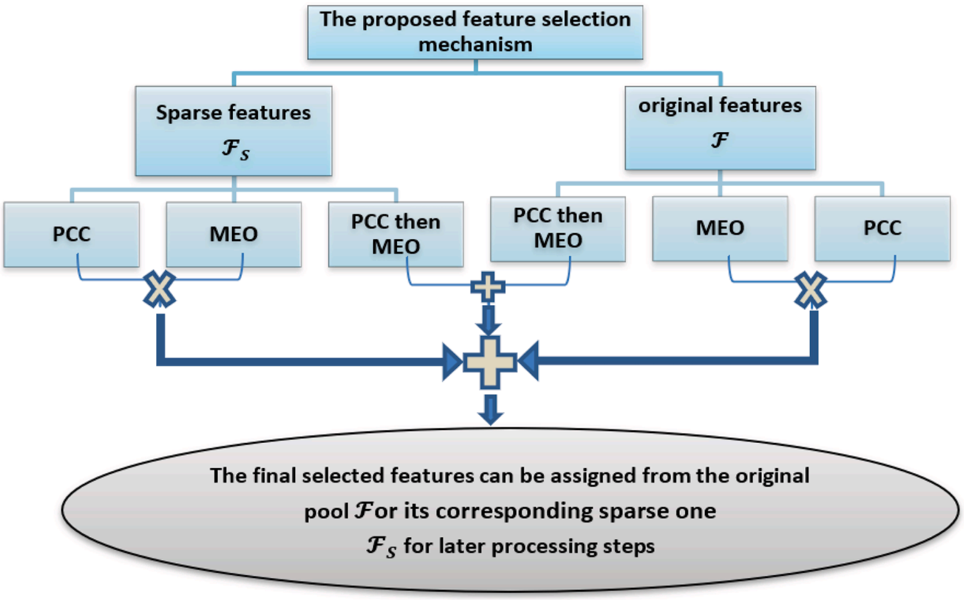

The proposed feature selection mechanism
Sparse features $\mathcal{F}_S$
original features $\mathcal{F}$
PCC
MEO
PCC then MEO
PCC then MEO
MEO
PCC
The final selected features can be assigned from the original pool $\mathcal{F}$ or its corresponding sparse one $\mathcal{F}_S$ for later processing steps

Fig. 6. An illustration of the proposed feature selection technique that is based on a fusion process between Pearson dropping (PCC) and the introduced Minkowski-based equilibrium optimizer (MEO) in a serial and parallel manner in the original features domain $\mathcal{F}$ once and in the proposed sparse domain $\mathcal{F}_S$ another. ″+″ represents combining decisions by OR operations while ″ × ″ represents seeking the intersections of decisions by AND operations.

$$\min_{\mathcal{F}_L,\mathcal{F}_S} \|\mathcal{F} - \mathcal{F}_L - \mathcal{F}_S\|_F^2 s.t.rank(\mathcal{F}_L) \le \zeta, card(\mathcal{F}_S) \le \Psi \tag{2}$$

The optimization problem in Eq. (2) can be solved by alternatively solving two sub-problems until convergence. These two subproblems can be expressed at iteration i as

$$\begin{cases} \mathcal{F}_{Li} = \underset{rank(\mathcal{F}_L)\le\zeta}{\operatorname{argmin}} \|\mathcal{F} - \mathcal{F}_L - \mathcal{F}_{Si-1}\|_F^2 \\ \mathcal{F}_{Si} = \underset{card(\mathcal{F}_S)\le\Psi}{\operatorname{argmin}} \|\mathcal{F} - \mathcal{F}_{Li} - \mathcal{F}_S\|_F^2 \end{cases} \tag{3}$$

The above two subproblems in Eq. (3), particularly, can be solved by updating $\mathcal{F}_{Li}$ via singular value hard thresholding (Candès et al., 2011) of $\mathcal{F} - \mathcal{F}_{Si-1}$ and updating $\mathcal{F}_{Si}$ via entry-wise hard thresholding of $\mathcal{F} - \mathcal{F}_{Li}$, i.e., keeping Ψ entries of $\mathcal{F} - \mathcal{F}_{Li}$ that have the largest absolute values, as

$$\begin{cases} \mathcal{F}_{Li} = \sum_{j=1}^{\zeta} \lambda_j \mathcal{U}_j \mathcal{V}_j^T, SVD(\mathcal{F} - \mathcal{F}_{Si-1}) = \mathcal{U}\Lambda\mathcal{V}^T \\ \mathcal{F}_{Si} = \mathrm{H}_\gamma(\mathcal{F} - \mathcal{F}_{Li}), |\gamma| \le \Psi \end{cases} \tag{4}$$

where H_γ represent entry-wise hard thresholding operation.

The main computational cost of solving the previous subproblems belongs to SVD in updating the low-rank component $\mathcal{F}_{Li}$, especially with large feature pool size $\mathcal{F}$. In (Halko et al., 2009), the authors prove that a matrix can be well approximated by its projection onto the column space of its random projections. This rank-revealing method provides a fast approximation of SVD. Hence, given ζ bilateral random projections (BRP) of an $n \times k$ dense feature pool matrix $\mathcal{F}$ (w.l.o.g, $n \ge k$), i.e., $X_1 = \mathcal{F}B_1$ and $X_2 = \mathcal{F}^T B_2$, where $B_1 \in \mathscr{R}^{k\times\Re}$ and $B_2 \in \mathscr{R}^{n\times\Re}$ are independent Gaussian random matrices, the low-rank component $\mathcal{F}_L$ can be obtained according to (Fazel et al., 2008) as

$$\mathcal{F}_L = X_1 \left(B_2^T X_1\right)^{-1} X_2^T \tag{5}$$

However, B_1 and B_2 are correlated random matrices updated from X_2 and X_1, respectively, and $\mathcal{F}_L$ can be obtained as a tight rank-$\Re$ approximation to a full rank matrix $\mathcal{F}$. Hence, we replace SVD with BRP, since BRP based low-rank approximation is near optimal and efficient in order to significantly reduce the time cost (Zhou and Tao, 2011). However, when singular values of the feature pool $\mathcal{F}$ decay slowly, Eq. (5) may perform poorly, i.e., doesn't guarantee a tight rank-R approximation. Accordingly, the power scheme in (Zhou and Tao, 2011) can be employed with BRP to perform the decomposition process.

According to the power scheme, we instead calculate BRP of a new version of the feature pool matrix $\widetilde{\mathcal{F}} = \left(\mathcal{F}\mathcal{F}^T\right)^q \mathcal{F}$, whose singular values decay faster than $\mathcal{F}$. In particular, $\lambda_i\left(\widetilde{\mathcal{F}}\right) = \lambda_i\left(\widetilde{\mathcal{F}}\right)^{2q+1}$. Both $\widetilde{\mathcal{F}}$ and $\mathcal{F}$ share the same singular vectors. The BRP of $\widetilde{\mathcal{F}}$ can be expressed as

$$X_1 = \widetilde{\mathcal{F}} B_1 \, and \, X_2 = \widetilde{\mathcal{F}}^T B_2 \tag{6}$$

Like Eq. (5), the BRP based ζ rank approximation of $\widetilde{\mathcal{F}}$ is demonstrated as

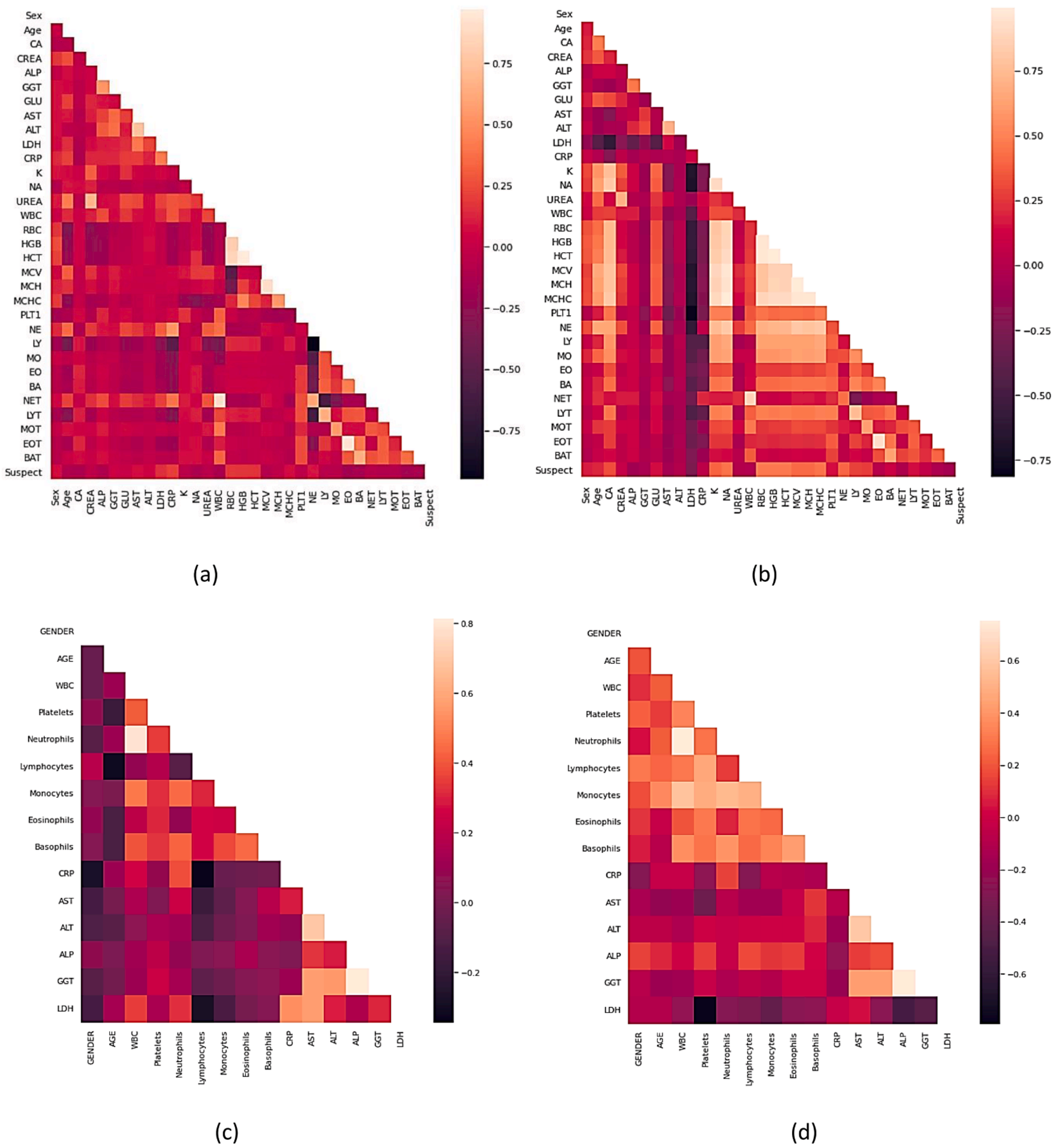
Algorithm 1.

Algorithm 1: The introduced decomposition process for the feature pool $\mathcal{F}$

```
1. Input: $\mathcal{F}, \zeta = 1, \Psi = n \times k, \epsilon = 0.001, q = 1$
2. Initialize: $\mathcal{F}_{L0} := \mathcal{F}, \mathcal{F}_{S0} := 0, i := 0$
3. While $\|\mathcal{F} - \mathcal{F}_L - \mathcal{F}_S\|_F^2 / \|\mathcal{F}\|_F^2 > \epsilon$ do        // the stopping criterion
4. $i := i + 1$
5. $\widetilde{\mathcal{F}}_L = \left[(\mathcal{F} - \mathcal{F}_{Si-1})(\mathcal{F} - \mathcal{F}_{Si-1})^T\right]^q (\mathcal{F} - \mathcal{F}_{Si-1})$        // following the
   formulation $\widetilde{\mathcal{F}} = (\mathcal{F}\mathcal{F}^T)^q \mathcal{F}$ in power scheme; at $q = 1$, we got $\widetilde{\mathcal{F}} = \mathcal{F}$,
   $\mathcal{F} \approx \mathcal{F}_L + \mathcal{F}_S$
6. $X_1 = \widetilde{\mathcal{F}}_L B_1, B_2 = X_1$
7. $X_2 = \widetilde{\mathcal{F}}_L^T X_1 = Q_2 R_2, X_1 = \widetilde{\mathcal{F}}_L X_2 = Q_1 R_1$
8. If rank$(B_2{}^T X_1) \langle \zeta$ then $\zeta := rank(B_2{}^T X_1)$, go to the first step; end
                                                                    // see Eq. (9)
9. $\mathcal{F}_{Li} = \left(\widetilde{\mathcal{F}}_L\right)^{\frac{1}{2q+1}} = Q_1 \left[R_1 \left(B_2{}^T X_1\right)^{-1} R_2{}^T\right]^{\frac{1}{2q+1}} Q_2{}^T$
10. $\mathcal{F}_{Si} = \mathrm{H}_\gamma(\mathcal{F} - \mathcal{F}_{Li}), \gamma$ is the nonzero subset of the first $\Psi$ largest entries of
    $|\mathcal{F} - \mathcal{F}_{Li}|$
                                                                    // See Eq. (4), 9
11. End while
12. Output: $\mathcal{F}_L, \mathcal{F}_S$
```

$$\widetilde{\mathcal{F}}_L = X_1 \left(B_2^T X_1\right)^{-1} X_2^T \tag{7}$$

Hence, in order to obtain the approximation of the original feature



(a)

(b)

(c)

(d)

Fig. 7. Pairwise Pearson correlation of features: (a), (c) for the original feature pool $\mathscr{F}$ while (b), (d) for the sparsified feature pool $\mathscr{F}_S$. The first row for COVID-specific dataset and the second one for CBC dataset.

pool $\mathscr{F}$ with rank r, QR decomposition of X_1 and X_2 is calculated as

$$X_1 = Q_1R_1, X_2 = Q_2R_2 \tag{8}$$

Accordingly, the low-rank composite $\mathscr{F}_L$ and the sparse composite $\mathscr{F}_S$ of the original feature pool $\mathscr{F}$ can be demonstrated as

$$\begin{cases} \mathscr{F}_L = \left(\widetilde{\mathscr{F}}_L\right)^{\frac{1}{2q+1}} = Q_1\left[R_1\left(B_2^TX_1\right)^{-1}R_2^T\right]^{\frac{1}{2q+1}}Q_2^T \\ \mathscr{F}_S = \mathrm{H}_\gamma(\mathscr{F}-\mathscr{F}_L),\ |\gamma|\leqslant \Psi \end{cases} \tag{9}$$

Algorithm 1 summarizes the main steps for the decomposition process seeking the targeted sparse feature pool $\mathscr{F}_S$.

4.3. The proposed feature selection scheme

The goal of feature selection is to find which blood exams are more relevant to COVID-19 prediction. Hence, we can gain three jackpots: first, the number of required exams for the diagnostic decision is reduced and consequently the total price. Second, a dimensionality reduction is obtained and consequently less computations. Third, selecting the appropriate features helps to reduce data redundancy and to avoid noisy data, hence, the classification model performance can be improved.

After the sparsification process, we now have two versions of feature pool, i.e., the original preprocessed feature pool $\mathscr{F}$ and its corresponding sparse one $\mathscr{F}_S$. In addition, we intend to apply-two feature selectors. The first is Pearson correlation-based one (PCC) which provides a quick screen and removal of irrelevant features relying on the characteristics of the data, without any need to complicated machine learning algorithms, thus, it is computationally less expensive. However, PCC can give lower prediction performance. Hence, a second feature selector is needed. Inspired by the traditional Equilibrium optimizer (EO) (Faramarzi et al., 2020), which is a novel physics-based meta-heuristic optimization algorithm, we propose a new Minkowski-based equilibrium optimizer (MEO) which can provide better selection performance compared to the traditional EO. The advantages of such meta-heuristics include their simplicity, independency to the problem, flexibility, and gradient-free nature (Halim et al., 2021).

Having two feature selectors, they can be applied serially or parallelly and can be applied to the two versions of the feature pool, i.e., $\mathscr{F}$ and $\mathscr{F}_S$, then, the different selection decisions can be fused to get the

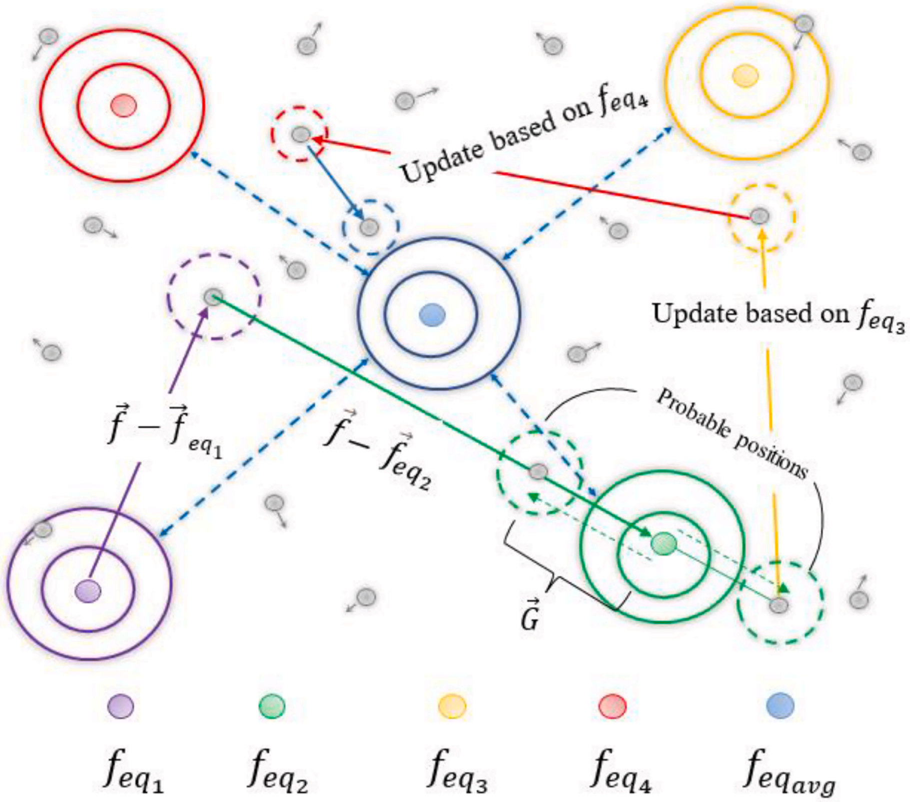

Update based on f_{eq_4}
Update based on f_{eq_3}
Probable positions
$\vec{f} - \vec{f}_{eq_1}$
$\vec{f} - \vec{f}_{eq_2}$
$\vec{G}$
f_{eq_1}
f_{eq_2}
f_{eq_3}
f_{eq_4}
$f_{eq_{avg}}$

Fig. 8. 2D illustration of Equilibrium candidates' collaboration in updating particles' concentration.

most important features as proposed in Fig. 6. Applying the selectors serially is expected to provide the best decisions in contrast to applying the selectors parallelly. Hence, we combine the decisions from the serial application of selectors through (OR) operation and seek the intersection in decisions from the parallel application of selectors through (AND) operation. In the following subsection, the introduced two feature selectors in the fused selection scheme, i.e., PCC, and the new Minkowski-based equilibrium optimizer (MEO) are indicated.

4.3.1. *Pearson correlation-based feature selection*

Features in their native form are not always correlated with each other. After the stage of feature sparsification, a clear and real correlation is exploited between features. The features with an extremely high correlation should be eliminated. Hence, reducing relevant features is helpful to loose the learned model and then eliminate overfitting to a certain extent. Pearson Correlation Coefficient (PCC) is employed here to help in the feature dropping task and it is expressed as in Eq. (10) to evaluate the linear correlation between two feature vectors f_i, f_j

$$PCC\left(f_i, f_j\right) = \frac{COV(f_i, f_j)}{\sigma_{f_i}\sigma_{f_j}} = \frac{E\left[\left(f_i - \mu_{f_i}\right)\left(f_j - \mu_{f_j}\right)\right]}{\sigma_{f_i}\sigma_{f_j}} \tag{10}$$

where COV is the covariance matrix, $\sigma_{f_i}, \sigma_{f_j}$ are the standard deviations of f_i, f_j, respectively, while μ_{f_i}, μ_{f_j} are the respective means. $PCC\left(f_i, f_j\right)$ can be ranged from -1 to 1. "1" indicates full positive correlation, while "-1" implies a negative full correlation. 0 is a sign of non-correlation. Mostly, f_i, f_j show extremely high correlation when $PCC\left(f_i, f_j\right)$ exceeds a threshold of 0.8 and strong correlation when $PCC\left(f_i, f_j\right)$ exceeds threshold of 0.6. In Fig. 7, pairwise Pearson correlation of the original feature pool $\mathscr{F}$ and the sparsified one $\mathscr{F}_S$ is shown. As indicated, after feature sparsification, Pearson maps become more brighter by discovering more correlation between features. For COVID-specific dataset, we have initially 34 features, by applying Pearson elimination to $\mathscr{F}$ and $\mathscr{F}_S$ with threshold 0.8, we got 28 selected features from $\mathscr{F}$, while we got 23 features from $\mathscr{F}_S$ which demonstrates that sparsity allowed us to drop more 5 features. With a threshold of 0.6, the feature pool $\mathscr{F}$ is reduced from 34 to 24 features, while the sparsified feature pool $\mathscr{F}_S$ turned from 34 into 15 features, which means that sparsity allowed us to drop more 9 features. On the other side, for CBC dataset, we have initially 16 features. The features in both $\mathscr{F}$ and $\mathscr{F}_S$ in CBC dataset don't show a correlation higher than 0.8. However, with a Pearson threshold of 0.6, the feature pool $\mathscr{F}$ reduced from 16 to 13 features, while the sparsified one $\mathscr{F}_S$ turned from 16 into 12 features, which means that sparsity allowed us to drop more 1 feature.

4.3.2. *Equilibrium-based feature selection*

4.3.2.1. *The traditional equilibrium optimizer (EO).*

Equilibrium optimization (EO) is originally inspired by the dynamic mass balance equation which describes the conservation of mass that enters, leaves, or generates in a control volume (Faramarzi et al., 2020). In another words, the dynamic mass balance equation is utilized to measure the number of mass entries and be generated in the volume over a period of time. The following three steps indicates the operation of EO.

***Step1:* Initialization.**

Similar to other meta-heuristic algorithms, the EO search starts by initializing the population of candidate solutions/ features/blood exams. For this initialization, a uniform random one in the search space is required. Eq. (11) demonstrates the initial distributed solutions in the search space.

$$f_i^{initial} = f_{min} + rand_i \times (f_{max} - f_{min}), i = 1, \cdots, \mathscr{n} \tag{11}$$

where $f_i^{initial}$ indicates to the ith candidate solutions/features. f_{min}, and

f_{max} are the minimum and maximum bounds for the ith candidate solution $f_i^{initial}$, respectively. *rand* is d-dimensional random vector ranging from zero to one. $\mathscr{n}$ specifies the number of particles/solutions in the group. Then, the equilibrium candidates are determined by a sorting process to their fitness function.

The objective function, i.e., fitness function, is employed within each optimization process in order to measure the fitness or the quality of each solution. The solution with the best fit is assigned as the best-so-far one for solving the targeted optimization problem. The proposed fitness function is a weighted sum between the classification accuracy based on KNN classifier and the proportion of the number of features/particles selected during each iteration, as

$$\xi = w\ acc + (1-w)\frac{K}{k} \tag{12}$$

where *acc* represents the classification accuracy of the currently selected features. *K* represents the number of the currently selected features, while *k* is the total number of features in the feature pool. *w* is a weighting random coefficient between [0, 1].

***Step 2*: Selecting equilibrium pool and candidates**.

As most of meta-heuristic algorithms that search for food source, EO searches for the equilibrium state of system/problem. At the beginning of the optimization process, the equilibrium state is unknown, i.e., the concentrations that achieve equilibrium are unknowns. The equilibrium state represents the global optimum of the optimization problem which is the final convergence state of the algorithm. However, equilibrium candidates are identified to provide a search domain for the particles. According to Faramarzi et al. (2020), choosing or assigning five equilibrium candidates, mostly, works effectively. The first four $f_{eq_i}, i \in \{1,2,3,4\}$, are the four "best-so-far" particles identified in the population during the whole optimization process and the last one is the particle with concentration equals the arithmetic mean of the previous mentioned four particles, i.e., $f_{eq_{avg}}$. The first four candidates help EO to have better diversification capability, while the last average one enhances the EO exploitation. Of course, the optimization problem has a word on determining the most proper number of candidates. The equilibrium pool $\overrightarrow{F}_{eq}$ is a vector constructed from these candidates as

$$\overrightarrow{F}_{eq} = \left\{\overrightarrow{f}_{eq_1}, \overrightarrow{f}_{eq_2}, \overrightarrow{f}_{eq_3}, \overrightarrow{f}_{eq_4}, \overrightarrow{f}_{eq_{avg}}\right\} \tag{13}$$

***Step 3*: updating the concentration**.when the candidate solutions/features are initialized using Eq. (11), their positions are updated over iterations by

$$\overrightarrow{f_i^{\ddot{I}+1}} = \overrightarrow{f_{eq}^{\ddot{I}}} + \left(\overrightarrow{f_i^{\ddot{I}}} - \overrightarrow{f_{eq}^{\ddot{I}}}\right)\overrightarrow{\Omega_i^{\ddot{I}}} + \frac{\overrightarrow{G_i^{\ddot{I}}}}{\overrightarrow{\alpha_i^{\ddot{I}}}\ \overrightarrow{v_i^{\ddot{I}}}}\left(1 - \overrightarrow{\Omega_{\mathrm{i}}^{\ddot{\mathrm{I}}}}\right) \tag{14}$$

where $\overrightarrow{f_i^{\ddot{I}}}$ and $\overrightarrow{f_i^{\ddot{I}+1}}$ are the original and updated concentrations of solutions/features at $\ddot{I}$ and $\ddot{I}+1$, respectively. $\overrightarrow{f_{eq}^{\ddot{I}}}$ is a randomly selected feature vector from the equilibrium pool $\overrightarrow{F}_{eq}$. The exponential term $\Omega = e^{-\overrightarrow{\alpha}(t-t_0)}$, as indicated in Eq. (14), helps in the main concentration updating role by keeping a good balance between exploration and exploitation in the Equilibrium optimization process. The exponential term Ω relies on the turnover rate α, and the time interval $(t-t_0)$. α is originally varies with time in a real control volume. Hence, it is supposed to be a random vector ranging from zero to one. On the other side, the time interval boundaries are defined as

$$t = \left(1 - \frac{\ddot{I}}{\mathscr{N}_{\ddot{I}}}\right)^{\left(\eta_2 \frac{\ddot{I}}{\mathscr{N}_{\ddot{I}}}\right)} \tag{15}$$

where $\ddot{I}$ is the iteration number, while $\mathscr{N}_{\ddot{I}}$ is the total number of iterations. η_2 is a constant value for controlling the exploitation process.

$$\overrightarrow{t_0} = \frac{1}{\overrightarrow{\alpha}}\ln\left(-\eta_1\ sign(\overrightarrow{r} - 0.5)\left[1 - e^{-\overrightarrow{\alpha}t}\right]\right) + t \tag{16}$$

where η_1 denotes a constant value which controls the diversification and intensification of EO process. By increasing the parameter η_1, the exploration/diversification capability increases while the exploitation/intensification ability decreases. On the other side, the higher the parameter η_2, the higher intensification capability and the lower diversification capability. The term *sign*$(\overrightarrow{r} - 0.5)$ controls the direction of exploitation and exploration based on another random vector, *r*, ranging from zero to one. By employing Eq. (15), (16), the exponential term can be rewritten as

$$\overrightarrow{\Omega^{\ddot{I}}} = \eta_1 sign(\overrightarrow{r} - 0.5)\left[e^{\overrightarrow{-\alpha^{\ddot{I}}}t} - 1\right] \tag{17}$$

Another term to enhance the exploitation phase is the generation rate G which is a first-order exponential decay process demonstrated as

$$\overrightarrow{G_i^{\ddot{I}}} = \overrightarrow{G_0^{\ddot{I}}}e^{-\overrightarrow{\alpha_i^{\ddot{I}}}(t-t_0)} \tag{18}$$

where

$$\overrightarrow{G}_0 = \overrightarrow{\omega}\left(\overrightarrow{f}_{eq} - \overrightarrow{\alpha}\overrightarrow{f}\right), \overrightarrow{\omega} = \begin{cases} 0.5r_1, r_2 \geq p_\omega \\ 0, r_2 < p_\omega \end{cases} \tag{19}$$

where r_1, and r_2 are random numbers in a range from zero to one. ω is defined as the generation rate control parameter, i.e., it controls generation term's contribution to the updating process. p_ω is the probability of how many particles utilize generation term to update their states. For keeping a good balance between exploitation and exploration, p_ω is assigned a value of 0.5.

Fig. 8 indicates a 2D representation of the equilibrium candidates' collaboration to update the concentration of a particle. In this figure, $f_1 - f_{eq}$ is representative of the second term in Eq. (14) and it is responsible for searching the space, i.e., exploration role, to find an optimum point. The large variation between a sample concentration and the equilibrium makes the term $f_1 - f_{eq}$ contribute more to the exploration process of EO. On the other side, the term $f_{eq} - \alpha f_1$ is a representative of the third term in Eq. (14). It introduces small variations in the concentration once a point is found by the exploration process. These small variations contribute to making the solution more accurate. Hence, the term $f_{eq} - \alpha f_1$ contributes more to the exploitation process of EO. In addition, the sign of both the second and the third term helps in the exploration and the exploitation process. Having the same signs makes the variations larger and accordingly searching the full space better, while opposite signs keep small variations which enhances the local searches. In Algorithm 2, a pseudo code to indicate the procedure of the traditional EO is demonstrated.

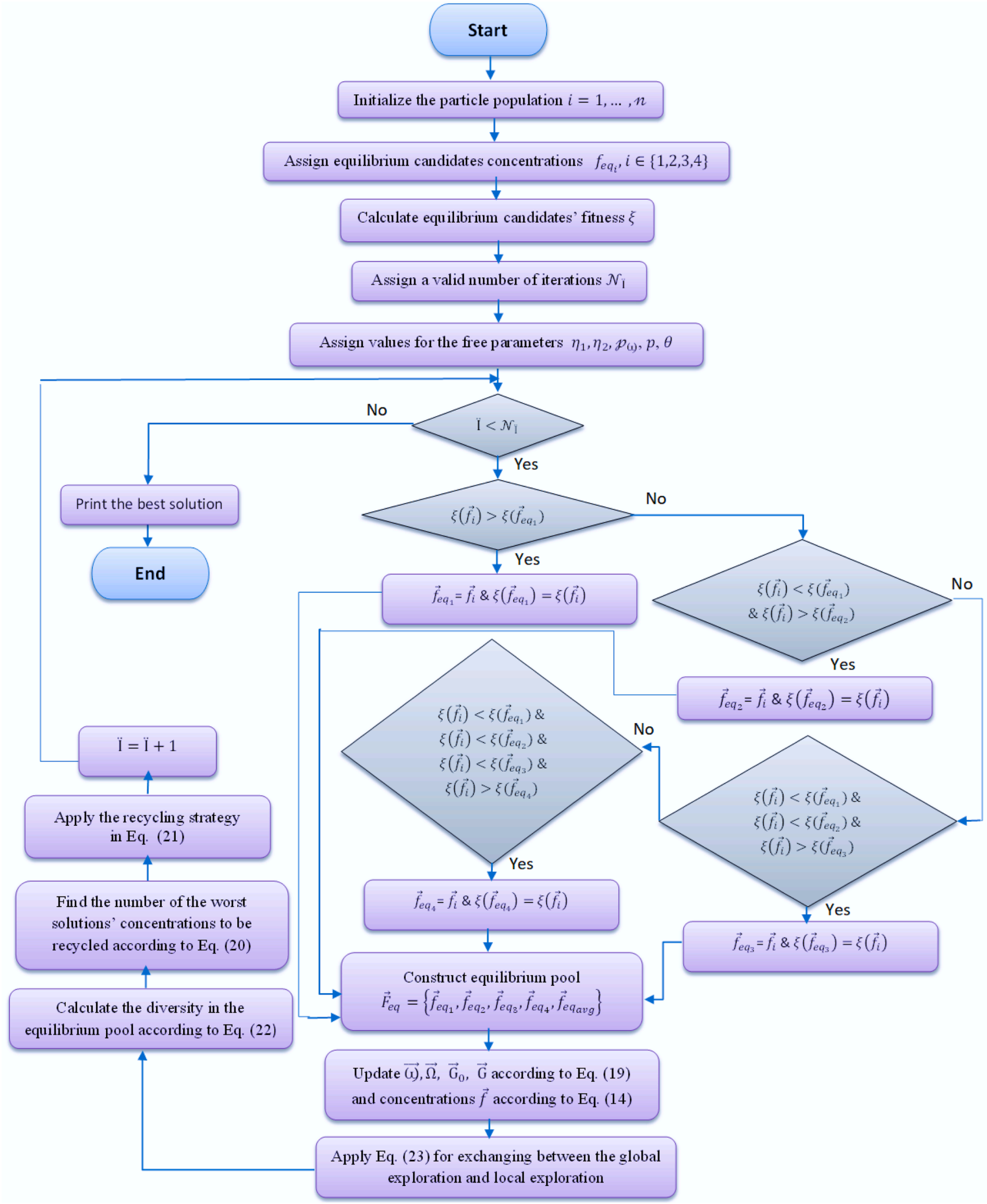

Start
Initialize the particle population $i = 1, \dots, n$
Assign equilibrium candidates concentrations $f_{eq_i}, i \in \{1,2,3,4\}$
Calculate equilibrium candidates' fitness ξ
Assign a valid number of iterations $\mathcal{N}_{\text{I}}$
Assign values for the free parameters $\eta_1, \eta_2, p_{\omega}, p, \theta$
No
$\text{Ï} < \mathcal{N}_{\text{I}}$
Yes
Print the best solution
End
$\xi(\vec{f}_i) > \xi(\vec{f}_{eq_1})$
No
Yes
$\vec{f}_{eq_1} = \vec{f}_i$ & $\xi(\vec{f}_{eq_1}) = \xi(\vec{f}_i)$
$\xi(\vec{f}_i) < \xi(\vec{f}_{eq_1})$ & $\xi(\vec{f}_i) > \xi(\vec{f}_{eq_2})$
No
Yes
$\vec{f}_{eq_2} = \vec{f}_i$ & $\xi(\vec{f}_{eq_2}) = \xi(\vec{f}_i)$
$\text{Ï} = \text{Ï} + 1$
$\xi(\vec{f}_i) < \xi(\vec{f}_{eq_1})$ & $\xi(\vec{f}_i) < \xi(\vec{f}_{eq_2})$ & $\xi(\vec{f}_i) < \xi(\vec{f}_{eq_3})$ & $\xi(\vec{f}_i) > \xi(\vec{f}_{eq_4})$
No
$\xi(\vec{f}_i) < \xi(\vec{f}_{eq_1})$ & $\xi(\vec{f}_i) < \xi(\vec{f}_{eq_2})$ & $\xi(\vec{f}_i) > \xi(\vec{f}_{eq_3})$
Apply the recycling strategy in Eq. (21)
Yes
$\vec{f}_{eq_4} = \vec{f}_i$ & $\xi(\vec{f}_{eq_4}) = \xi(\vec{f}_i)$
Yes
Find the number of the worst solutions' concentrations to be recycled according to Eq. (20)
$\vec{f}_{eq_3} = \vec{f}_i$ & $\xi(\vec{f}_{eq_3}) = \xi(\vec{f}_i)$
Construct equilibrium pool $\vec{F}_{eq} = \{\vec{f}_{eq_1}, \vec{f}_{eq_2}, \vec{f}_{eq_3}, \vec{f}_{eq_4}, \vec{f}_{eq_{avg}}\}$
Calculate the diversity in the equilibrium pool according to Eq. (22)
Update $\vec{\omega}, \vec{\Omega}, \vec{G}_0, \vec{G}$ according to Eq. (19) and concentrations $\vec{f}$ according to Eq. (14)
Apply Eq. (23) for exchanging between the global exploration and local exploration

Fig. 9. Flow chart of the proposed MEO algorithm.

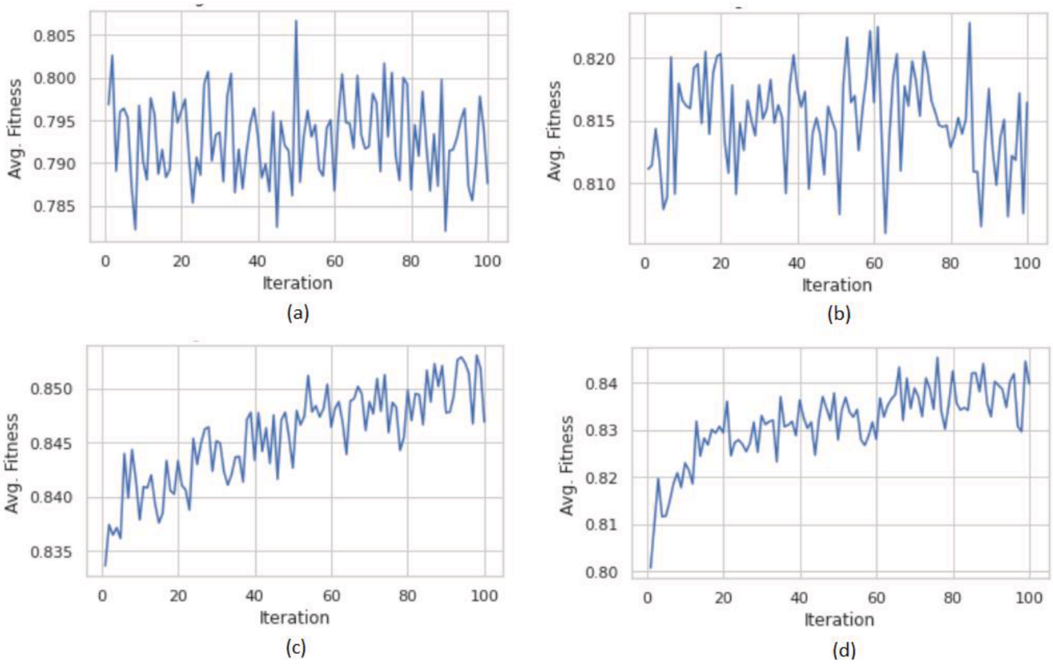


Fig. 10. Comparison of the results of average fitness over iterations for the traditional EO, in the first row, and the proposed MEO, in the second one, for COVID-specific dataset. The first column is the results of the original feature pool $\mathscr{F}$ while the second one for the sparsified feature pool $\mathscr{F}_S$.

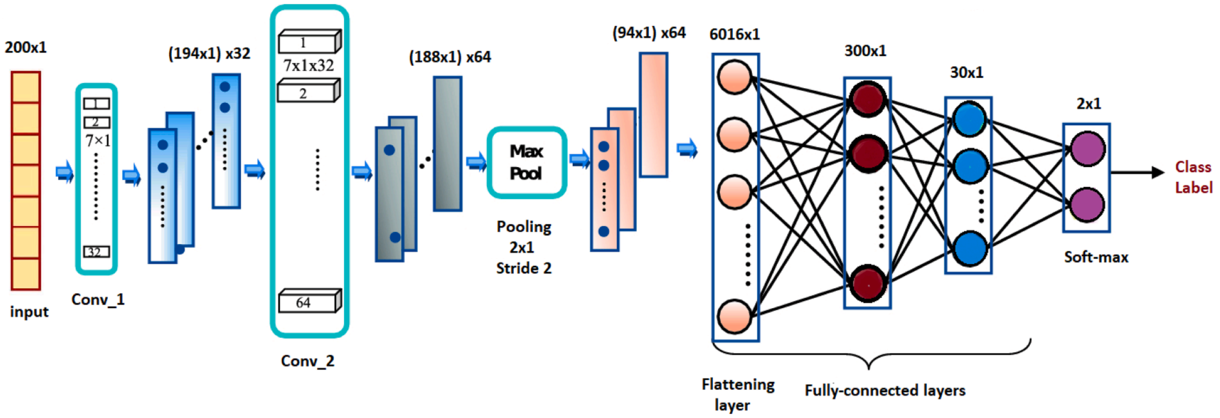


Fig. 11. An example of 1DCNN model for a binary classification problem. In this example, the network consists of two convolutional layers (Conv_1with 32 filters and Conv_2 with 64 filters), Max pooling layer, flattening layer and finally some fully connected layers with soft-max layer.

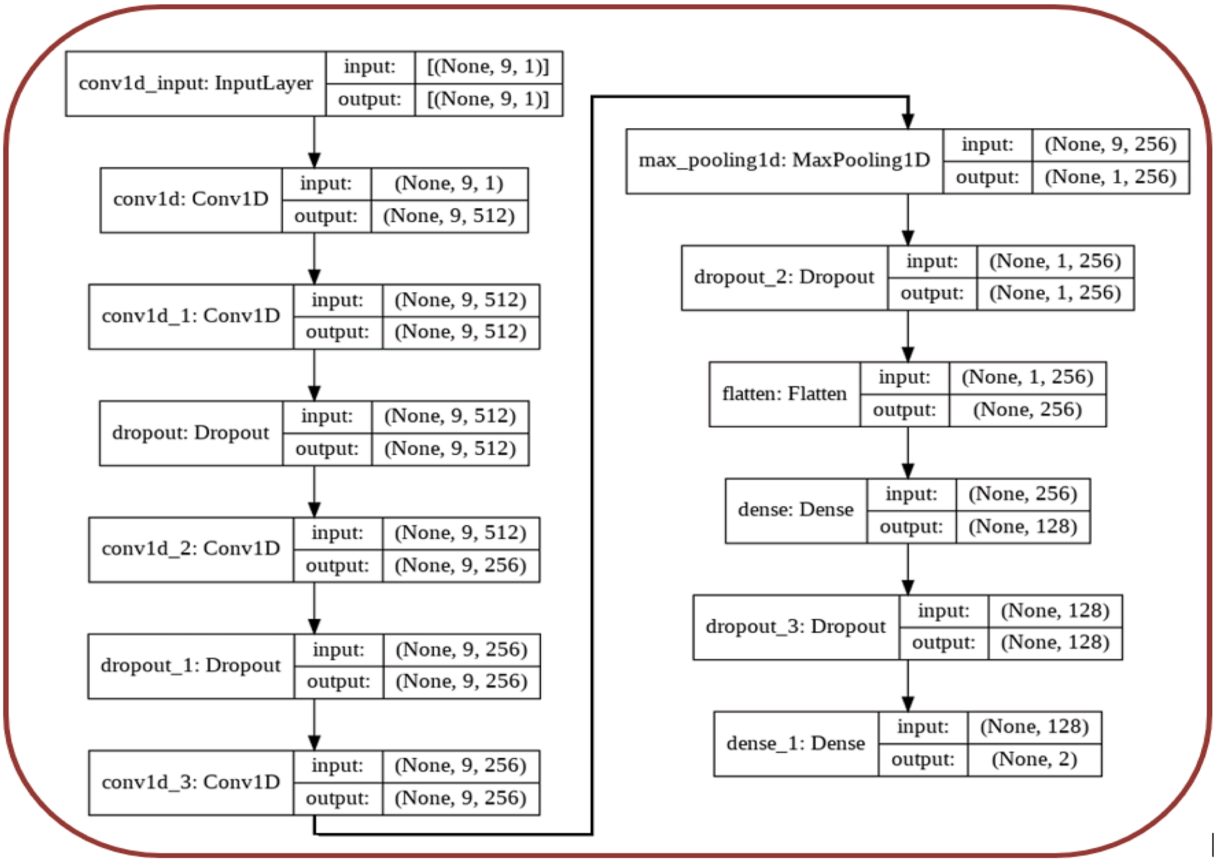
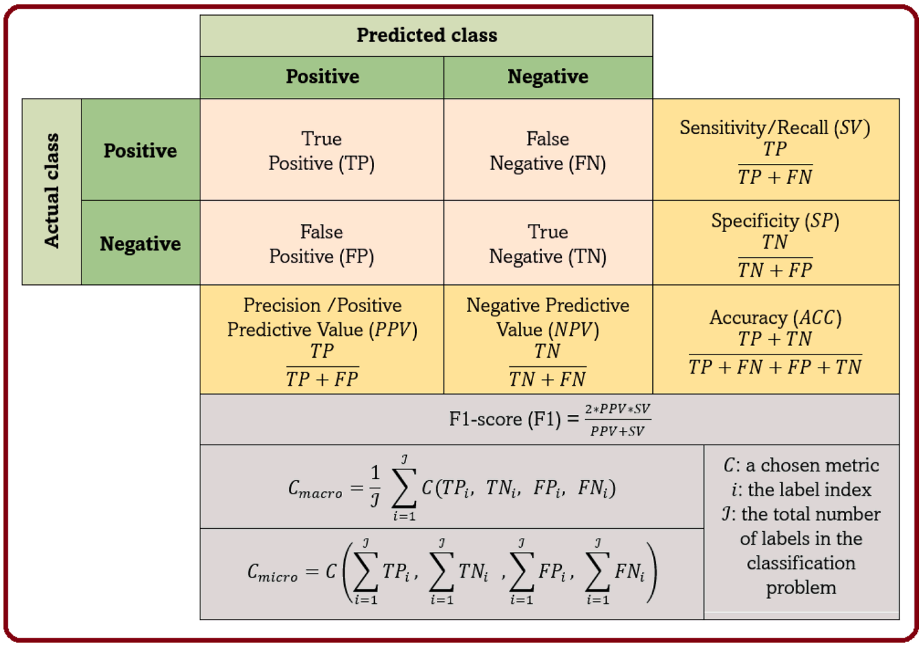


Fig. 12. Summary of the proposed 1DCNN for COVID-19 prediction considering 9 selected features.



		Predicted class		
		Positive	**Negative**	
Actual class	**Positive**	True Positive (TP)	False Negative (FN)	Sensitivity/Recall (SV) $\frac{TP}{TP+FN}$
	Negative	False Positive (FP)	True Negative (TN)	Specificity (SP) $\frac{TN}{TN+FP}$
		Precision /Positive Predictive Value (PPV) $\frac{TP}{TP+FP}$	Negative Predictive Value (NPV) $\frac{TN}{TN+FN}$	Accuracy (ACC) $\frac{TP+TN}{TP+FN+FP+TN}$
		F1-score (F1) = $\frac{2*PPV*SV}{PPV+SV}$		
		$C_{macro} = \frac{1}{\mathcal{J}}\sum_{i=1}^{\mathcal{J}} C(TP_i,\ TN_i,\ FP_i,\ FN_i)$		C: a chosen metric i: the label index $\mathcal{J}$: the total number of labels in the classification problem
		$C_{micro} = C\left(\sum_{i=1}^{\mathcal{J}} TP_i, \sum_{i=1}^{\mathcal{J}} TN_i, \sum_{i=1}^{\mathcal{J}} FP_i, \sum_{i=1}^{\mathcal{J}} FN_i\right)$		

Fig. 13. The employed evaluation metrics.

Table 4
The validation results of the effect of employing the data preparation steps, i.e., SMOTE for data balancing and iForest for outlier detection, on the proposed COVID-19 diagnosis algorithm on the employed datasets.

dataset	Case	Case name	ACC	PPV	SV	F1	AUC	SP	Features*
Covid-specific dataset	1	Imbalanced w/ outliers	0.83	0.834	0.816	0.824	0.890	0.773	19/33
	2	Balanced w/ outliers	0.866	0.876	0.918	0.896	0.929	0.921	18/33
	3	Imbalanced w/o outliers	0.894	0.882	0.902	0.891	0.956	0.901	19/33
	4	Balanced w/o outliers	**0.988**	**0.985**	**0.975**	**0.979**	**0.988**	**0.985**	13/33
CBC dataset	1	Imbalanced w/ outliers	0.771	0.783	0.9	0.833	0.824	0.806	7/13
	2	Balanced w/ outliers	0.923	0.921	0.956	0.938	0.976	0.938	6/13
	3	Imbalanced w/o outliers	0.906	0.902	0.956	0.92	0.95	0.931	9/13
	4	Balanced w/o outliers	**0.994**	**0.985**	**0.993**	**0.986**	**0.998**	**0.986**	6/13

[*] The selected number of features (x) out of the total size of the original feature pool (y); (x/y)

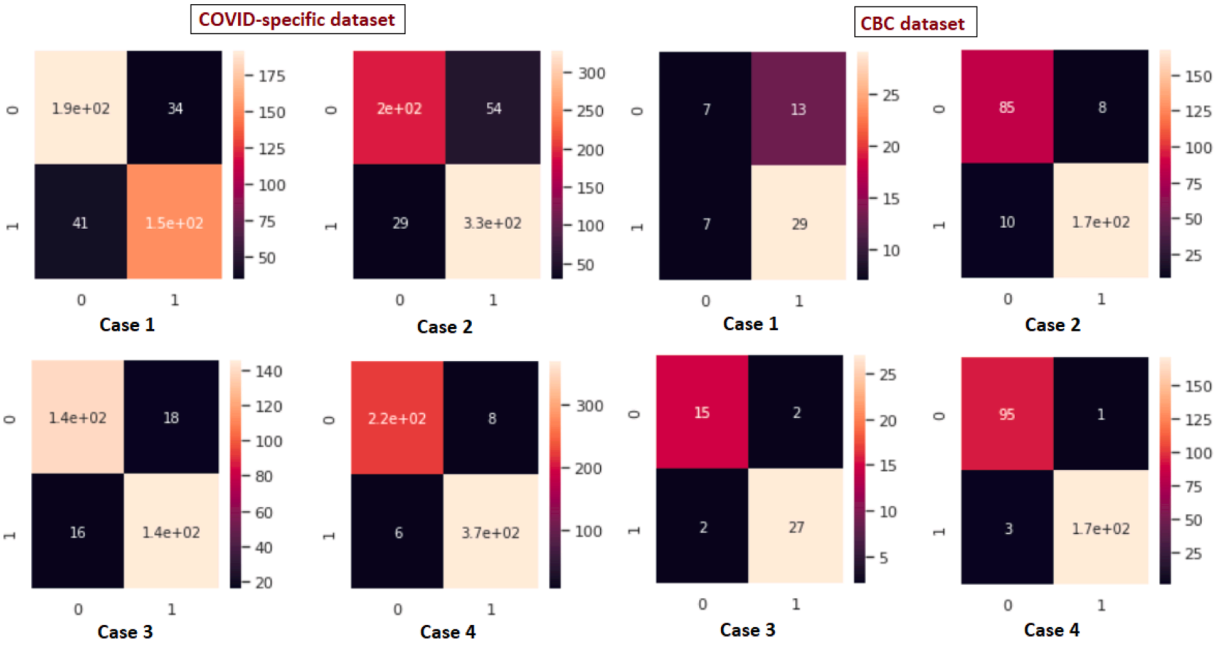


Fig. 14. Confusion matrices of testing the proposed COVID-19 prediction algorithm adopting the four cases indicated in Table 4 showing the effect of SMOTE and iForest on the performance.

Table 5
Validation results of applying all features, and PCC and MEO-based feature selection, separately, in different cases for COVID-specific dataset. The best performance is marked by bold font. (–) is the number of selected features.

	Train and testing for the original samples in $\mathscr{F}$						Train and testing for the sparse samples in $\mathscr{F}_s$					
	ACC	PPV	SV	F1	AUC	SP	ACC	PPV	SV	F1	AUC	SP
All features (33)	0.94	0.943	0.96	0.952	0.986	0.961	0.942	0.944	0.96	0.954	**0.987**	**0.966**
PCC-based feature selection for the original features in $\mathscr{F}$ (22 features)	0.939	0.94	0.96	0.95	0.98	0.958	0.943	0.947	**0.961**	0.954	0.986	0.958
PCC-based feature selection for the sparse features in $\mathscr{F}_s$ (14 features)	0.932	0.935	0.958	0.946	0.977	0.958	0.939	0.942	0.96	0.95	0.983	0.955
MEO-based selection for the original features in $\mathscr{F}$ (12 features)	0.925	0.93	0.95	0.94	0.98	0.948	0.932	0.94	0.952	0.945	0.98	0.961
MEO-based selection for the sparse features in $\mathscr{F}_s$ (12 features)	0.934	0.94	0.96	0.947	0.983	0.958	**0.944**	**0.95**	0.96	**0.955**	0.984	0.958

Table 6
Validation results of applying all features, and PCC and MEO-based feature selection, separately, in different cases for CBC dataset. The best performance is marked by bold font. (–) is the number of selected features.

	Train and testing for the original samples in $\mathscr{F}$						Train and testing for the sparse samples in $\mathscr{F}_s$					
	ACC	PPV	SV	F1	AUC	SP	ACC	PPV	SV	F1	AUC	SP
All features (13)	0.989	0.985	0.997	0.991	0.999	1	**0.99**	**0.988**	**0.995**	**0.992**	**1**	**1**
PCC-based feature selection for the original features in $\mathscr{F}$ (10 features)	0.987	0.982	0.997	0.99	0.999	1	0.986	0.983	0.995	0.989	0.999	1
PCC-based feature selection for the sparse features in $\mathscr{F}_s$ (9 features)	0.989	0.987	0.995	0.991	0.999	1	0.989	0.987	0.998	0.991	0.999	1
MEO-based selection for the original features in $\mathscr{F}$ (8 features)	0.99	**0.988**	1	0.992	0.999	1	0.986	0.984	0.994	0.989	0.999	1
MEO-based selection for the sparse features in $\mathscr{F}_s$ (6 features)	0.961	0.963	0.974	0.968	0.989	0.964	0.969	0.968	0.983	0.974	0.996	0.988

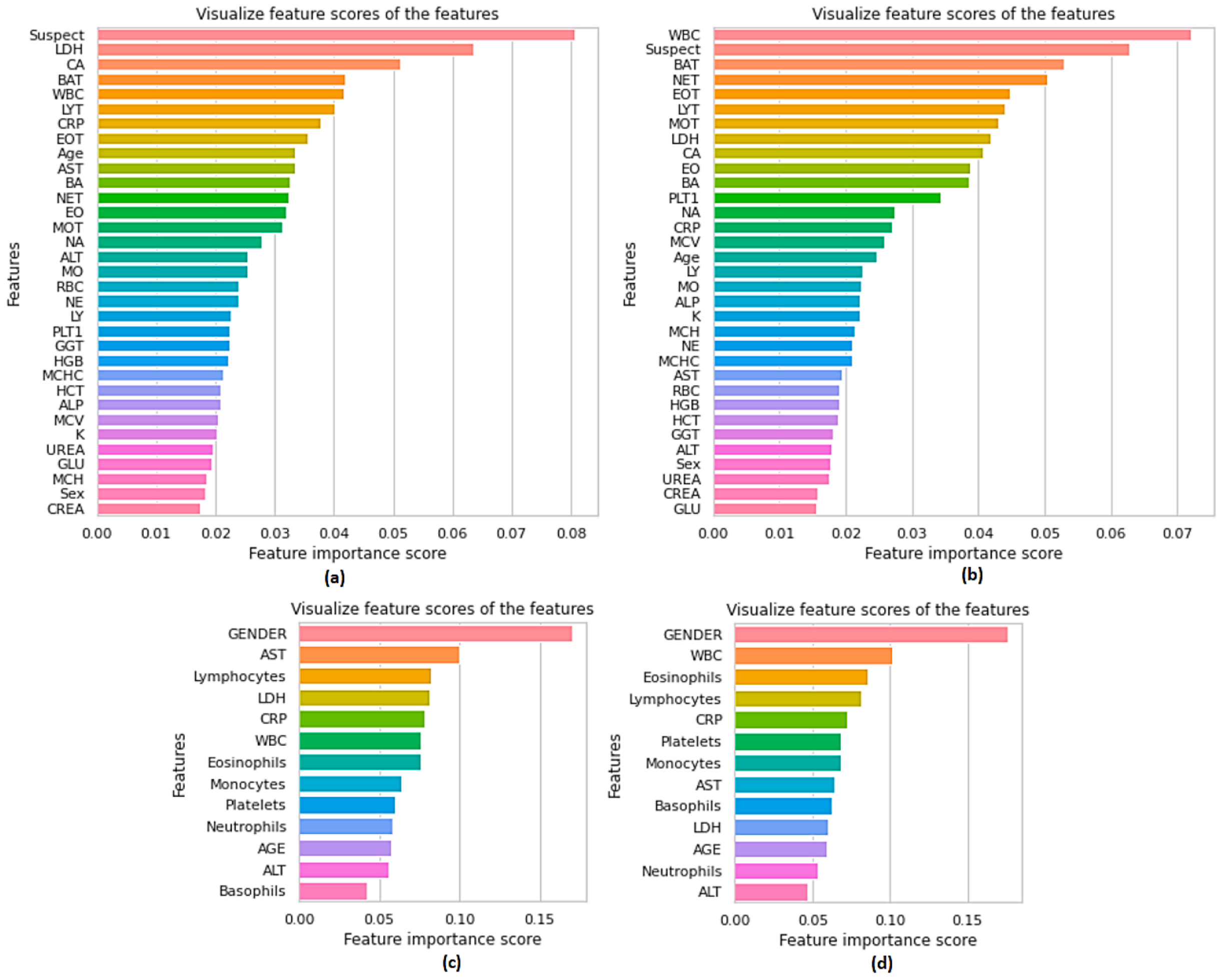


Fig. 15. AdaBoost feature importance employing all features for COVID-specific dataset in the first row and CBC dataset in the second row. (a) and (c) in the features original domain while (b) and (d) in the sparse domain.

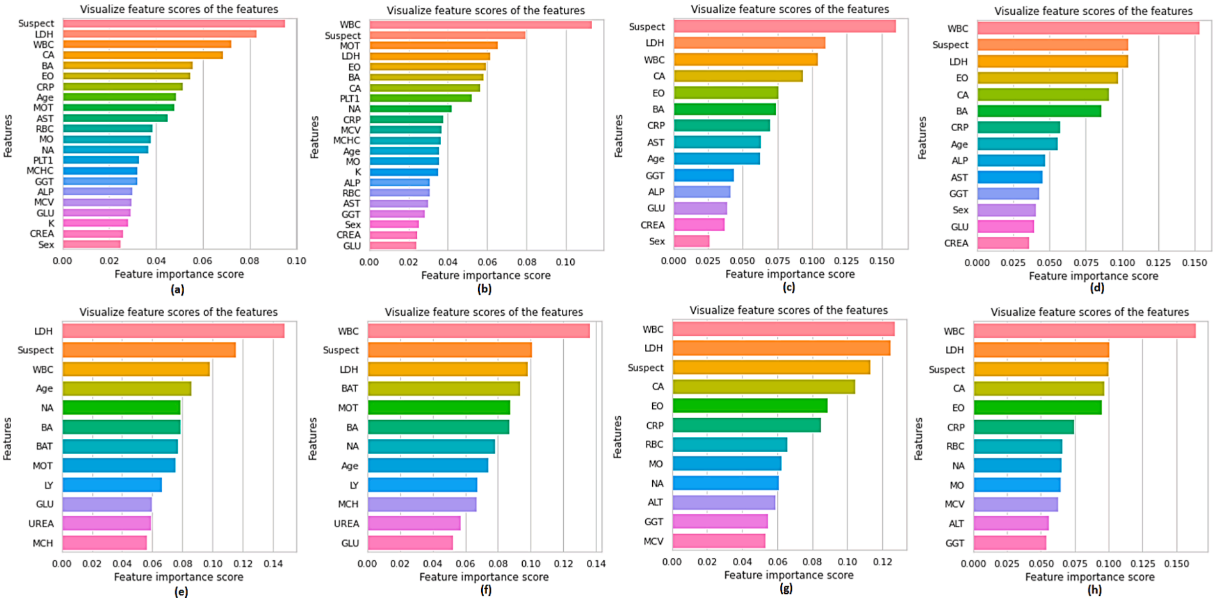


Fig. 16. AdaBoost feature importance, for COVID-specific dataset, adopting the followings: 1. PCC-based feature selection in the features original domain (22 feature selected) while applying training and testing once for the original samples in $\mathscr{F}$ (a), and another for the sparse samples in $\mathscr{F}_s$ (b). 2. PCC-based feature selection in sparse domain (14 feature selected) while applying training and testing once for the original samples in $\mathscr{F}$ (c), and another for the sparse samples in $\mathscr{F}_s$ (d). 3. MEO-based feature selection in features original domain (12 feature selected) while applying training and testing once for the original samples in $\mathscr{F}$ (e), and another for the sparse samples in $\mathscr{F}_s$ (f). 4. PCC-based feature selection in sparse domain (12 feature selected) while applying training and testing once for the original samples in $\mathscr{F}$ (g), and another for the sparse samples in $\mathscr{F}_s$ (h).

Algorithm 2.

Algorithm 2: Pseudo code of the traditional EO optimizer

```
Initialize the solution's/ particle's population randomly, $i = 1, \cdots, \mathscr{n}$;
                                        // Eq. (11), $\mathscr{n}$ is the search agents
Assign a small number to the equilibrium candidates' objective/ fitness function $\xi$;
                                        // $\xi = 0.0001$
Select the equilibrium candidates $\overrightarrow{f}_{eq_1}, \overrightarrow{f}_{eq_2}, \overrightarrow{f}_{eq_3}, \overrightarrow{f}_{eq_4}$ from the population
Update the states of candidate solutions using search equation (Eq. (14))
Assign values to the following free parameters $\eta_1 = 2, \eta_2 = 1, p_\omega = 0.5$
While $\ddot{I} < \mathscr{N}_{\ddot{I}}$            // the iteration no. $\mathscr{N}_{\ddot{I}} =$ 100
  For $i = 1 : \mathscr{n}$
  Calculate the fitness function of the $i^{th}$ particle $\xi(\overrightarrow{f}_i)$    // follow Eq. (12) to calculate $\xi$
    If $\xi\left(\overrightarrow{f}_i\right)\rangle\xi(\overrightarrow{f}_{eq_1})$
      Replace $\overrightarrow{f}_{eq_1}$ with $\overrightarrow{f}_i$ and $\xi(\overrightarrow{f}_{eq_1})$ and $\xi\left(\overrightarrow{f}_i\right)$
    Elseif $\xi\left(\overrightarrow{f}_i\right)\langle\xi(\overrightarrow{f}_{eq_1})\ \&\xi\left(\overrightarrow{f}_i\right)\rangle\xi(\overrightarrow{f}_{eq_2})$
      Replace $\overrightarrow{f}_{eq_2}$ with $\overrightarrow{f}_i$ and $\xi(\overrightarrow{f}_{eq_2})$ and $\xi\left(\overrightarrow{f}_i\right)$
    Elseif $\xi\left(\overrightarrow{f}_i\right)\langle\xi(\overrightarrow{f}_{eq_1})\ \&\ \xi\left(\overrightarrow{f}_i\right)\langle\xi(\overrightarrow{f}_{eq_2})\ \&\xi\left(\overrightarrow{f}_i\right)\rangle\xi(\overrightarrow{f}_{eq_3})$
      Replace $\overrightarrow{f}_{eq_3}$ with $\overrightarrow{f}_i$ and $\xi(\overrightarrow{f}_{eq_3})$ and $\xi\left(\overrightarrow{f}_i\right)$
    Elseif $\xi\left(\overrightarrow{f}_i\right)\langle\xi(\overrightarrow{f}_{eq_1})\ \&\ \xi\left(\overrightarrow{f}_i\right)\langle\xi(\overrightarrow{f}_{eq_2})\ \&$
  $\xi\left(\overrightarrow{f}_i\right)\langle\xi(\overrightarrow{f}_{eq_3})\&\xi\left(\overrightarrow{f}_i\right)\rangle\xi(\overrightarrow{f}_{eq_4})$
      Replace $\overrightarrow{f}_{eq_4}$ with $\overrightarrow{f}_i$ and $\xi(\overrightarrow{f}_{eq_4})$ *and* $\xi\left(\overrightarrow{f}_i\right)$
    End If
  End for
  $\overrightarrow{f}_{eq_{avg}} = \left(\overrightarrow{f}_{eq_1} + \overrightarrow{f}_{eq_2} + \overrightarrow{f}_{eq_3} + \overrightarrow{f}_{eq_4}\right) \Big/ 4$
```

(*continued on next column*)

(*continued*)

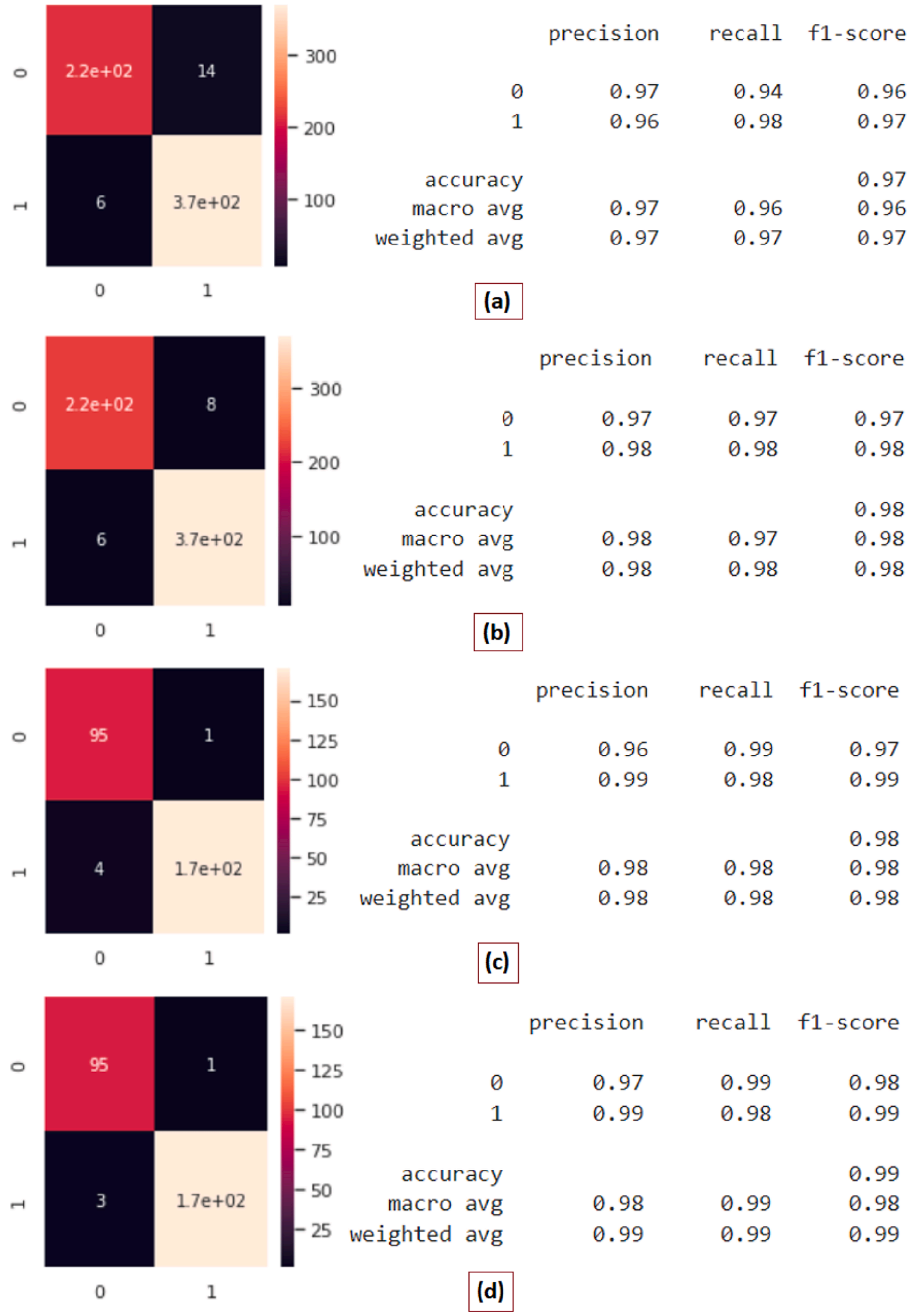
```
  Construct the equilibrium pool $\overrightarrow{F}_{eq} = \left\{\overrightarrow{f}_{eq_1}, \overrightarrow{f}_{eq_2}, \overrightarrow{f}_{eq_3}, \overrightarrow{f}_{eq_4}, \overrightarrow{f}_{eq_{avg}}\right\}$
                                        Accomplish memory saving if $\ddot{I} > 1$
  Assign $t$ according to Eq. (15)
  For $i = 1 : \mathscr{n}$
    Choose one candidate, randomly, from the equilibrium pool $\overrightarrow{F}_{eq}$
    Generate random vectors of $\overrightarrow{r}$ and $\overrightarrow{\alpha}$ from Eq. (17)
    Construct $\overrightarrow{\omega}, \overrightarrow{\Omega}, \overrightarrow{G}_0, \overrightarrow{G}$ according to Eq. (19)
    Update concentration $\overrightarrow{f}$ according to Eq. (14)
  End for
  $\ddot{I} = \ddot{I} + 1$
End While
```

4.3.2.2. The proposed Minkowski-based equilibrium optimizer (MEO). In the proposed MEO, we try to move a set of the worst solutions, i.e., particles with worst fitness, toward the "best-so-far" attempting to find better solution in a smaller number of iterations. However, this recycling idea may cause an entrapment in local minima, and accordingly, the chance of having better global solution is impossible. Hence, in the proposed modified version of EO (MEO), a recycling strategy for the worst solutions is presented with a strategy for local minima suppression. The proposed MEO is indicated in the following subsections.

Recycling strategy for the worst solutions:

As mentioned before, the main purpose of this strategy is to move the worst solutions toward the best-so-far solutions, hence, the chance to find solutions better than the best-so-far solutions can be enhanced. At the same time, the recycling strategy should guarantee to take the solutions away from the local minimum. Hence, the number of worst solutions $\mathbb{N}$ to be recycled is controlled by the following equation.

$$\mathbb{N} = n - round\left(\frac{\ddot{I}}{\mathscr{N}_{\ddot{I}}}(n - \mathrm{n})\right) \tag{20}$$



```
              precision    recall  f1-score

           0       0.97      0.94      0.96
           1       0.96      0.98      0.97

    accuracy                           0.97
   macro avg       0.97      0.96      0.96
weighted avg       0.97      0.97      0.97
```

```
              precision    recall  f1-score

           0       0.97      0.97      0.97
           1       0.98      0.98      0.98

    accuracy                           0.98
   macro avg       0.98      0.97      0.98
weighted avg       0.98      0.98      0.98
```

```
              precision    recall  f1-score

           0       0.96      0.99      0.97
           1       0.99      0.98      0.99

    accuracy                           0.98
   macro avg       0.98      0.98      0.98
weighted avg       0.98      0.98      0.98
```

```
              precision    recall  f1-score

           0       0.97      0.99      0.98
           1       0.99      0.98      0.99

    accuracy                           0.99
   macro avg       0.98      0.99      0.98
weighted avg       0.99      0.99      0.99
```

Fig. 17. Classification reports of testing the proposed COVID diagnosis model based on the proposed fused selection method and 1DCNN in both original domain (a), (c) and sparse domain (b), (d). The first two rows belong to COVID-specific dataset while the other rows belong to CBC dataset.

where n is the size of the initial population, while n denotes a fixed number of the solutions to be updated within each iteration. $\ddot{I}$ is the current iteration number and $\mathscr{N}_{\ddot{I}}$ is the total number of iterations. As indicated from Eq. (20), as the iteration number increases, the recycling strategy is controlled by decreasing the number of the worst solutions to be updated to decrease the chance of local minima entrapment.

After finding the most suitable number of worst particles to be recycled, the recycling mechanism of their concentrations/features are demonstrated as

$$\vec{f}_{worst} = \gamma_1 \vec{f}_{eq_{avg}} + (1-\gamma_1)\vec{f}_{eq_{rand}} + \gamma_2\left(\vec{f}_{eq_{rand}} - \vec{f}_{worst}\right) \tag{21}$$

where γ_1 is a weighting random parameter, in range from zero to one, between the mean equilibrium concentration $f_{eq_{avg}}$ and a randomly selected concentration from the equilibrium pool. This weighting mechanism is proposed to keep a suitable diversity between the worst solutions even after their movements towards the best-so-far ones. γ_2 is a random number ranging from zero to one.

Local minimum avoidance:

To support MEO in their fighting towards the local minima problem for achieving better solutions within their searches, the technique in Eq. (22), (23) is proposed. In this technique, both local and global exploration can be controlled according to the degree of the diversity in the equilibrium pool. The diversity in the equilibrium pool, D_f, is calculated in Minkowski-based manner between each pair of equilibrium concentrations as

Table 7
Computitative comparison between some traditional ML techniques and the proposed 1DCNN model while training and testing performed once in original features domain and another in sparse domain for COVID-specific dataset (13 selected Features out of 33). The top performer is bolded, while the second is underlined.

Training and testing Domain	Classifier	ACC	PPV	SV	F1	AUC	SP	Macro-			Micro-		
								PPV	SV	F1-	PPV	SV	F1-
Original features domain	LSVM	0.84	0.858	0.891	0.874	0.902	0.907	0.836	0.824	0.828	0.84	0.84	0.84
	RSVM	0.897	0.909	0.927	0.918	0.939	0.931	0.893	0.887	0.89	0.897	0.897	0.897
	LR	0.836	0.893	0.837	0.863	0.903	0.867	0.827	0.835	0.829	0.836	0.836	0.836
	RF	0.928	0.941	0.943	0.942	0.98	0.963	0.924	0.923	0.923	0.928	0.928	0.928
	AdaBoost	0.932	0.934	0.958	0.946	**0.983**	**0.984**	0.932	0.924	0.927	0.932	0.932	0.932
	DT	0.883	0.902	0.911	0.906	0.874	0.941	0.877	0.874	0.875	0.883	0.883	0.883
	KNN	0.869	0.868	0.932	0.899	0.936	0.949	0.872	0.85	0.857	0.869	0.869	0.869
	XGBoost	0.899	0.914	0.926	0.919	0.956	0.947	0.896	0.891	0.892	0.899	0.899	0.899
	GNB	0.787	0.794	0.888	0.838	0.858	0.925	0.783	0.756	0.763	0.787	0.787	0.787
	ET	0.931	0.934	0.956	0.945	**0.983**	0.979	0.93	0.923	0.926	0.931	0.931	0.931
	LDA	0.824	0.835	0.894	0.863	0.895	0.92	0.821	0.802	0.808	0.824	0.824	0.824
	QDA	0.783	0.789	0.89	0.836	0.869	0.917	0.78	0.749	0.757	0.783	0.783	0.783
	OURS	**0.967**	**0.974**	**0.958**	**0.965**	0.971	**0.984**	**0.971**	**0.956**	**0.965**	**0.967**	**0.955**	**0.964**
Sparse domain	LSVM	0.844	0.868	0.884	0.875	0.904	0.909	0.838	0.832	0.833	0.844	0.844	0.844
	RSVM	0.894	0.908	0.924	0.916	0.939	0.939	0.89	0.885	0.887	0.894	0.894	0.894
	LR	0.836	0.891	0.841	0.864	0.905	0.875	0.828	0.835	0.829	0.836	0.836	0.836
	RF	0.933	0.939	0.954	0.946	0.982	0.981	0.931	0.926	0.928	0.933	0.933	0.933
	AdaBoost	0.94	0.942	0.964	0.952	0.985	**0.987**	0.94	0.933	0.936	0.94	0.94	0.94
	DT	0.889	0.9	0.923	0.911	0.878	0.944	0.885	0.878	0.881	0.889	0.889	0.889
	KNN	0.892	0.886	0.948	0.916	0.947	0.965	0.897	0.874	0.882	0.892	0.892	0.892
	XGBoost	0.896	0.905	0.93	0.917	0.951	0.952	0.893	0.885	0.888	0.896	0.896	0.896
	GNB	0.822	0.862	0.849	0.855	0.869	0.885	0.811	0.813	0.811	0.822	0.822	0.822
	ET	0.94	0.942	0.963	0.952	0.985	**0.987**	0.94	0.933	0.936	0.94	0.94	0.94
	LDA	0.824	0.837	0.893	0.863	0.896	0.912	0.82	0.803	0.808	0.824	0.824	0.824
	QDA	0.79	0.803	0.88	0.839	0.865	0.912	0.785	0.762	0.768	0.79	0.79	0.79
	OURS	**0.983**	**0.982**	**0.975**	**0.976**	**0.987**	0.984	**0.98**	**0.971**	**0.973**	**0.98**	**0.971**	**0.972**

Table 8
Computitative comparison between some of the traditional ML techniques and the proposed 1DCNN model while training and testing performed once in original features' domain and in another in sparse domain for CBC dataset (6 selected features out of 13). The top performer is bolded, while the second is underlined.

Training and testing Domain	Classifier	ACC	PPV	SV	F1	AUC	SP	Macro-			Micro-		
								PPV	SV	F1-	PPV	SV	F1-
Original features' domain	LSVM	0.828	0.848	0.873	0.860	0.880	0.845	0.822	0.815	0.818	0.828	0.828	0.828
	RSVM	0.893	0.897	0.930	0.913	0.937	0.908	0.892	0.882	0.886	0.893	0.893	0.893
	LR	0.810	0.844	0.844	0.843	0.881	0.828	0.802	0.801	0.801	0.810	0.810	0.810
	RF	0.956	0.954	0.974	0.964	0.994	**0.989**	0.957	0.951	0.953	0.956	0.956	0.956
	AdaBoost	0.976	0.976	0.985	0.980	**0.998**	0.983	0.976	0.974	0.975	0.976	0.976	0.976
	DT	0.932	0.940	0.950	0.944	0.928	0.971	0.932	0.928	0.929	0.932	0.932	0.932
	KNN	0.955	0.958	0.968	0.963	0.988	0.960	0.955	0.951	0.952	0.955	0.955	0.955
	XGBoost	0.946	0.940	0.974	0.957	0.981	0.977	0.949	0.939	0.943	0.946	0.946	0.946
	GNB	0.694	0.854	0.597	0.701	0.827	0.557	0.715	0.720	0.693	0.694	0.694	0.694
	ET	0.977	0.975	**0.988**	0.981	**0.998**	0.983	0.978	0.974	0.976	0.977	0.977	0.977
	LDA	0.802	0.838	0.837	0.837	0.871	0.810	0.794	0.792	0.792	0.802	0.802	0.802
	QDA	0.760	0.889	0.692	0.777	0.877	0.644	0.768	0.779	0.758	0.760	0.760	0.760
	OURS	**0.984**	**0.981**	**0.988**	**0.982**	**0.998**	0.984	**0.98**	**0.984**	**0.981**	**0.98**	**0.983**	**0.98**
Sparse domain	LSVM	0.816	0.841	0.860	0.850	0.871	0.787	0.809	0.804	0.805	0.816	0.816	0.816
	RSVM	0.876	0.870	0.936	0.902	0.936	0.931	0.879	0.860	0.867	0.876	0.876	0.876
	LR	0.801	0.836	0.837	0.836	0.874	0.805	0.793	0.791	0.791	0.801	0.801	0.801
	RF	0.969	0.976	0.974	0.975	0.995	0.966	0.968	0.968	0.968	0.969	0.969	0.969
	AdaBoost	0.980	0.981	0.986	0.983	**0.998**	0.977	0.980	0.978	0.979	0.980	0.980	0.980
	DT	0.938	0.951	0.947	0.949	0.936	0.948	0.935	0.936	0.935	0.938	0.938	0.938
	KNN	0.960	0.968	0.966	0.967	0.988	0.966	0.959	0.959	0.958	0.960	0.960	0.960
	XGBoost	0.942	0.950	0.954	0.952	0.981	0.948	0.940	0.938	0.939	0.942	0.942	0.942
	GNB	0.752	0.877	0.689	0.770	0.842	0.667	0.759	0.769	0.750	0.752	0.752	0.752
	ET	0.980	0.982	0.985	0.983	**0.998**	0.983	0.980	0.978	0.979	0.980	0.980	0.980
	LDA	0.798	0.830	0.840	0.835	0.868	0.822	0.790	0.787	0.788	0.798	0.798	0.798
	QDA	0.777	0.864	0.751	0.803	0.872	0.695	0.773	0.784	0.772	0.777	0.777	0.777
	OURS	**0.991**	**0.983**	**0.99**	**0.985**	**0.998**	**0.984**	**0.981**	**0.988**	**0.987**	**0.981**	**0.987**	**0.981**

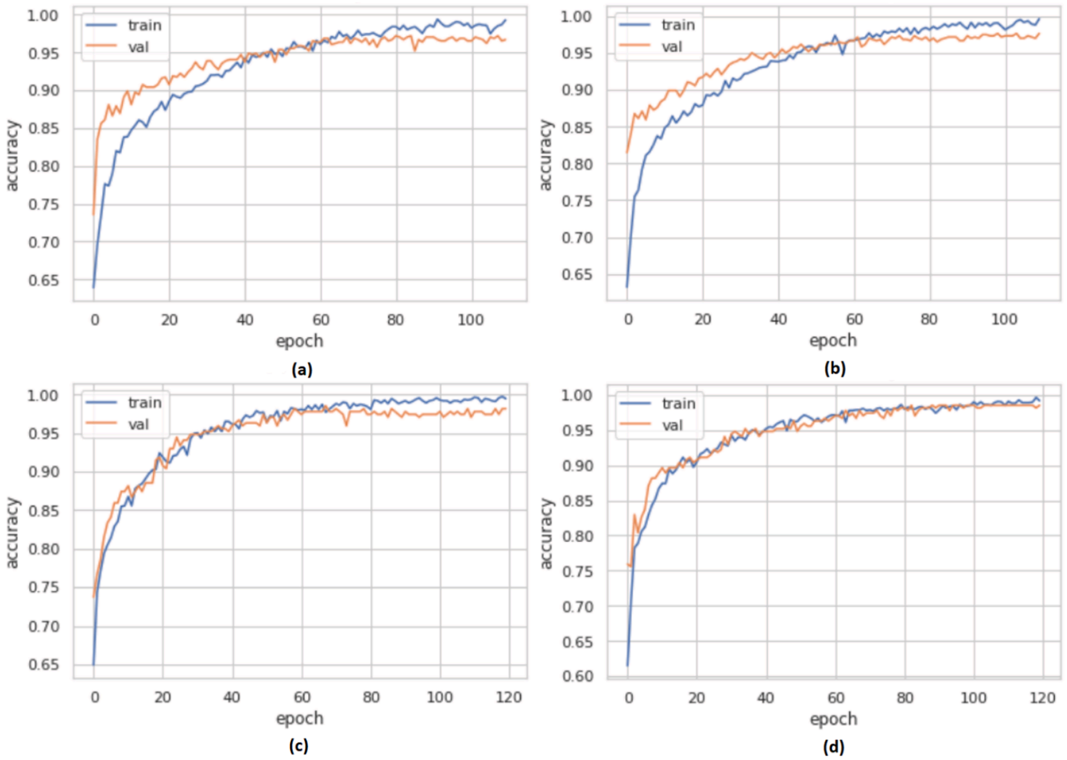


Fig. 18. Training-validation performance in terms of accuracy for the proposed COVID prediction algorithm. The first row for COVID-specific dataset and the other one for CBC dataset. The training in (a), (c) is performed in features original domain and the others (b), and (d) in sparse domain. The training is performed over the selected features by the proposed fused-based feature selection mechanism which results 13 features for COVID-specific dataset and 6 features for CBC-dataset.

$$D_f = \left(\sum_{i=1,j=1}^{\mathscr{N}_{eq}} \left| \overrightarrow{f}_{eq_i} - \overrightarrow{f}_{eq_{j\neq i}} \right|^p \right)^{1/p} \tag{22}$$

where p is the order of Minkowski distance metric. $\mathscr{N}_{eq}$ denotes the number of particles/ solutions in the equilibrium pool. Hence, the avoidance of local minima problem is demonstrated as

$$\overrightarrow{f} = \begin{cases} \overrightarrow{f} + \left(\overrightarrow{f}_{min} + \overrightarrow{rand}\left(\overrightarrow{f}_{max} - \overrightarrow{f}_{min} \right) \right) . \overrightarrow{\gamma}_3, if D_f < \theta \\ \overrightarrow{f} + \left(\frac{D_f}{\mathscr{N}_{eq}} \left(1 - \overrightarrow{\gamma}_4 \right) + \overrightarrow{\gamma}_4 \right) . \left(\overrightarrow{f}_i - \overrightarrow{f}_j \right), if D_f > \theta \end{cases} \tag{23}$$

where γ_3 and γ_4 are two random vectors in range [0,1]. The updating mechanism in Eq. (23) offers a global exploration property within the search boundaries f_{max},f_{min} when the diversity in the equilibrium pool is low, i.e., $D_f < \theta$, where θ is a specific predefined threshold for the degree of diversity. In the other hand, with large diversity, i.e., $D_f > \theta$, the updating mechanism offers a local exploration between two solutions selected randomly from the population, i.e., f_i, and f_j.

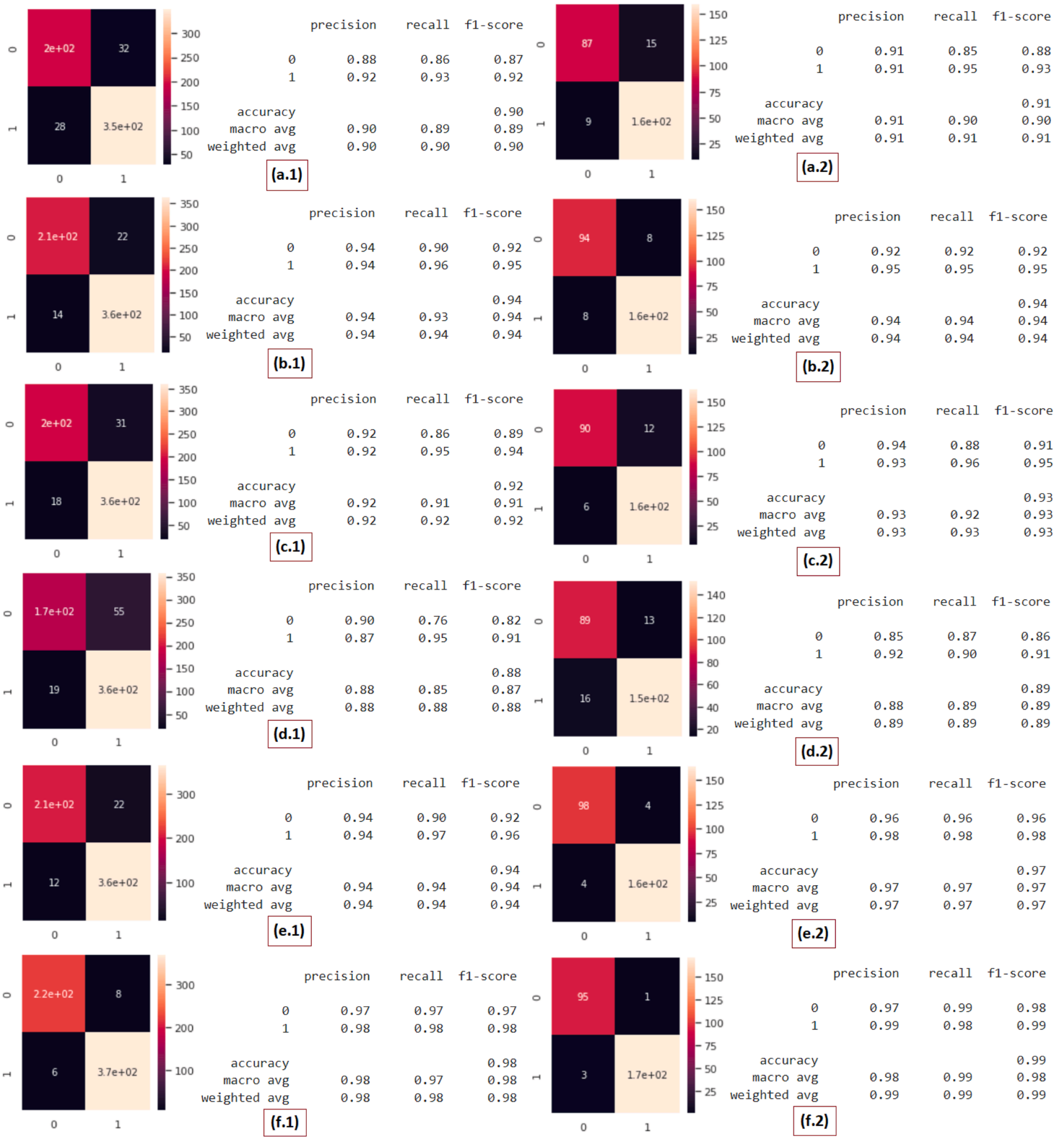
Fig. 9 indicates a flow chart for the proposed MEO. In addition, in Fig. 10, a comparison is set between the traditional EO and the proposed modified version MEO, with Minkowski distance of order $p = 3$, employing the original feature pool once and employing the sparsified one another for COVID-specific dataset. In the original feature domain $\mathscr{F}$, the traditional EO provides a leader fitness of 0.838 with a leader KNN classification accuracy of 0.864, while the proposed MEO provides a leader fitness of 0.87 with a leader KNN classification accuracy of 0.896. on the other side, in the sparse domain $\mathscr{F}_S$, the traditional EO provides a leader fitness of 0.848 with a leader KNN classification accuracy of 0.882, while the proposed MEO provides a leader fitness of 0.878 with a leader KNN classification accuracy of 0.911. Hence, Applying the proposed MEO to the sparsified feature pool $\mathscr{F}_S$ shows the best performance in terms of leader accuracy. In addition, sparse features, even, help the traditional EO to have better performance compared to the traditional features.

4.4. Classification stage:

Following the proposed dataset preprocessing and feature selection, an efficient classifier is needed. Mostly, ensembles of different machine learning classifiers are employed to guarantee better classification performance, such as the diagnosis criteria in (AlJame et al., 2020; Alves et al., 2021). In (Brinati et al., 2020; Cabitza et al., 2021; de Freitas Barbosa et al., 2021), the authors tried to introduce the performance of different machine learning classifiers in comparative way to choose the best classifier. On the other hand, instead of the traditional machine learning techniques, the authors in (Alakus & Turkoglu, 2020; Shaban et al., 2021) employed deep learning techniques in their diagnosis. (Alakus & Turkoglu, 2020) introduced a prediction study for COVID-19 disease with deep learning application models, such as Artificial Neural Network (ANN), Convolutional Neural Networks (CNN), Long-Short Term Memory (LSTM), Recurrent Neural Networks (RNN), CNNLSTM, and CNNRNN. Shaban et al. (2021) proposed a hybrid classification model that consists of two classifiers: fuzzy inference engine and Deep Neural Network (DNN).

Deep Learning (DL) is the latest accomplishment of the machine learning era by providing a multi-level hierarchical architecture with subsequent stages for more effective information processing. DL era is started by (Hinton & Salakhutdinov, 2006), when they explained the role of “the depth” of an ANN in machine learning. In other words, they pointed out the role of increasing the number of hidden layers in increasing the learning ability of networks.

Convolutional Neural Networks (CNN), as a common type of deep neural networks (DNN), are mostly used with two-dimensional data (2D CNN), such as images (Albawi et al., 2017). CNN mainly constructed from convolutional layers with pooling layers, as a feature extraction stages, and fully connected layers for classification. The advantages of



(a.1)

```
              precision    recall  f1-score

           0       0.88      0.86      0.87
           1       0.92      0.93      0.92

    accuracy                           0.90
   macro avg       0.90      0.89      0.89
weighted avg       0.90      0.90      0.90
```

(a.2)

```
              precision    recall  f1-score

           0       0.91      0.85      0.88
           1       0.91      0.95      0.93

    accuracy                           0.91
   macro avg       0.91      0.90      0.90
weighted avg       0.91      0.91      0.91
```

(b.1)

```
              precision    recall  f1-score

           0       0.94      0.90      0.92
           1       0.94      0.96      0.95

    accuracy                           0.94
   macro avg       0.94      0.93      0.94
weighted avg       0.94      0.94      0.94
```

(b.2)

```
              precision    recall  f1-score

           0       0.92      0.92      0.92
           1       0.95      0.95      0.95

    accuracy                           0.94
   macro avg       0.94      0.94      0.94
weighted avg       0.94      0.94      0.94
```

(c.1)

```
              precision    recall  f1-score

           0       0.92      0.86      0.89
           1       0.92      0.95      0.94

    accuracy                           0.92
   macro avg       0.92      0.91      0.91
weighted avg       0.92      0.92      0.92
```

(c.2)

```
              precision    recall  f1-score

           0       0.94      0.88      0.91
           1       0.93      0.96      0.95

    accuracy                           0.93
   macro avg       0.93      0.92      0.93
weighted avg       0.93      0.93      0.93
```

(d.1)

```
              precision    recall  f1-score

           0       0.90      0.76      0.82
           1       0.87      0.95      0.91

    accuracy                           0.88
   macro avg       0.88      0.85      0.87
weighted avg       0.88      0.88      0.88
```

(d.2)

```
              precision    recall  f1-score

           0       0.85      0.87      0.86
           1       0.92      0.90      0.91

    accuracy                           0.89
   macro avg       0.88      0.89      0.89
weighted avg       0.89      0.89      0.89
```

(e.1)

```
              precision    recall  f1-score

           0       0.94      0.90      0.92
           1       0.94      0.97      0.96

    accuracy                           0.94
   macro avg       0.94      0.94      0.94
weighted avg       0.94      0.94      0.94
```

(e.2)

```
              precision    recall  f1-score

           0       0.96      0.96      0.96
           1       0.98      0.98      0.98

    accuracy                           0.97
   macro avg       0.97      0.97      0.97
weighted avg       0.97      0.97      0.97
```

(f.1)

```
              precision    recall  f1-score

           0       0.97      0.97      0.97
           1       0.98      0.98      0.98

    accuracy                           0.98
   macro avg       0.98      0.97      0.98
weighted avg       0.98      0.98      0.98
```

(f.2)

```
              precision    recall  f1-score

           0       0.97      0.99      0.98
           1       0.99      0.98      0.99

    accuracy                           0.99
   macro avg       0.98      0.99      0.98
weighted avg       0.99      0.99      0.99
```

Fig. 19. Classification reports of testing the following studies: (Alakus & Turkoglu, 2020) {18/33–13/13} as (a), (AlJame et al., 2020) {18/33–13/13} as (b), (Cabitza et al., 2021) {33/33–13/13} as (c), (Brinati et al., 2020) {33/33–13/13} as (d), (Shaban et al., 2021) {33/33–13/13} as (e), and Ours {13/33–6/13} as (f). {} denotes {selected features/total no. of features for COVID-specific dataset – selected features/total no. of features for CBC dataset. (?.1) for COVID-specific dataset and (?.2) for CBC dataset.

CNNs can be summarized as following. 1) with a single body, CNNs can guarantee a fusion process between feature extraction and feature classification. 2) The features can be directly optimized from the raw input during the training process. 3) CNNs can deal with large inputs effectively via sparsely-connected neurons outperforming the traditional Multi-Layer Perceptrons (MLP) networks. 4) CNNs are robust to small variants in the input data, such as translation, scaling, skewing and distortion.

1D Convolutional Neural Networks (1DCNNs) have recently been developed (Kiranyaz et al., 2021). They can deal efficiently with 1D signals. 1DCNNs have superior advantages, such as low computational cost due to employing 1D convolutions instead of 2D convolutions in 2DCNNs. Usually, 1DCNN employs small number of hidden layers, hence, we get small number of learning parameters which suits CPU implementations and real-time applications.

To understand the performed operations in a 1DCNN, Fig. 11 indicates a simplified example that provides an overview. Conv is a 1D convolution layer with some feature detectors (filters). The selected number of filters defines how many sliding windows are used. Each filter has a kernel size (filter length) that matches the size/height of the slider window. This window will slide through the data and lead to an output matrix. The first Conv layer learns the basic functions. An additional 1D convolution layer with other filters before pooling allows our model to learn more complex functions. If the $\mathbb{L}$ layer is a convolution layer, the formula for the one-dimensional convolution layer, i.e., the output of this layer, is indicated in Eq. (24). Pooling layer is a layer to reduce variance and computation complexity (e.g., average pooling reduces 75 % of data) and extract low-level features from the neighborhood. Applying Max Pooling moves a window over our data and replaces the values with the maximum value. The pooling layer will remove a certain percentage of our values from the previous layer, creating a new matrix. To further reduce the probability of over-fitting, a drop layer is added. Dense layer is a fully connected layer to ensure better classification results. Flatten is a layer to flatten the multi-dimensional data, resulting from the previous conv layers, which cannot be feed directly into the feed forward neural network. Hence, they are used usually before dense layers to flatten data firstly. The final layer is a dense layer which uses a SoftMax activation function to generate a probability distribution across the output classes. The final output layer consists of neurons (one for each label/output class) including their probability. The output of a fully connected layer $\mathbb{O}\ (\mathscr{M})$ is demonstrated in Eq. (25).

$$\mathscr{M}_j^{\mathbb{L}} = \mathbb{U}\left(\sum_{\mathrm{i}=1}^{\mathrm{c}} \mathscr{M}_{\mathrm{j}}^{\mathbb{L}-1} * \mathscr{H}_{\mathrm{ij}}^{\mathbb{L}} + \chi_{\mathrm{j}}^{\mathbb{L}}\right) \tag{24}$$

where $\mathscr{H}$ denotes the convolution kernels, j represents the number of kernels, c indicates the channel number of input $\mathscr{M}_j^{\mathbb{L}-1}$; $\mathscr{M}_j^{\mathbb{L}-1}$ represents the output from the previous layer. χ denotes the bias corresponding to the kernel. $\mathbb{U}$ denotes the adopted activation function and * represents the convolution operator.

$$\mathbb{O}\ (\mathscr{M}) = \mathbb{U}\left(\mathscr{M}^{\mathbb{L}+1}.\ \ \varpi^{\mathbb{L}+1} + \chi^{\mathbb{L}+1}\right) \tag{25}$$

where ϖ, χ denote the weights and the bias, respectively.

In Fig. 12, a graphical summary the adopted 1DCNN model in the proposed COVID −19 prediction algorithm is shown. As indicated, the proposed network consists of 4 convolutional layers with filters sizes $\{512, 512, 256, 256\}$, all have the same kernel size of 32 and all employ ReLU as activation function. We employed 4 drop out layers. The first three with a dropping factor of 0.2 and the last one with a dropping factor of 0.5. Their main function is to inactivate 20 %, and 50 % of neurons, respectively, in order to prevent overfitting. Then, a flatten layer is utilized to flatten the multi-dimensional data to suit a dense layer of 128 neurons. Finally, a dense layer with 2 neurons and with SoftMax activation function is used to suit the binary classification problem. We have employed Stochastic Gradient Descent (SGD) as an optimizer with a learning rate of 0.001, and momentum of 0.9. The stopping criterion of the training process is when we got no change in the validation accuracy for 5 epochs.

5. Experimental results and discussions

To assess the performance of the proposed algorithm, several performance metrics are employed. We chose six metrics to evaluate the performance: accuracy (ACC), precision (PPV), F1-score, AUC, specificity (SP), and sensitivity (SV). Those metrics are based on the resultant confusion matrix values, i.e., TP, TN, FP, and FN, see Fig. 13 for the metrics formulas. AUC (Area Under The Curve) ROC (Receiver Operating Characteristics) demonstrates the relation between TP rate, on Y-axis, and FP rate, on X-axis. The higher the AUC, the higher the efficiency of the model in differentiating between the problem's two classes.

During the performed experiments of the proposed model, to avoid the risk of over-fitting, the employed dataset is splitted into a training set (75 % of the instances) and a test set (25 % of the instances) using a stratified procedure, then 30 % of the training set is used for validation. During all performed experiments, the models were trained and calibrated on the whole training set. Later, the calibrated models were evaluated on the hold-out test set in terms of the previously mentioned six metrics. In the upcoming subsections, the performance of the proposed diagnosis algorithm is indicated in detail through an ablation study. In this ablation study, while we discuss the impact of a specific step, we keep the other steps of the proposed algorithm the same, see Fig. 2 for an illustration of the whole algorithm.

5.1. *Impact of data preparation*

In this step, as discussed in the methodology section, we employed iForest algorithm for outlier detection and SMOTE for data balancing. Table 4 indicates computitative validation results of the effect of employing the data preparation steps on the proposed COVID-19 diagnosis algorithm while the other main steps are kept the same. As indicated, we got improvements in all evaluation metrics as we move from case (1) of not applying any preprocessing steps to the employed dataset to case (4) of applying both SMOTE and iForest for data balancing and outlier detection, respectively. For iForest, we kept the default parameter in Scikit-learn library, except the number of base estimators, we employed 150. For SMOTE, we kept the default parameters in Imbalanced-learn library. In Table 4, in COVID-specific dataset, we got the biggest enhancement in the specificity with an increase from 0.773 (case 1) to 0.958 (case 4) with a reduction of 6 features. On the other side, in CBC dataset, we got the biggest enhancement in accuracy with an increase from 0.771 to 0.983. In Fig. 14, the confusion matrices of testing the prementioned cases is indicated. These matrices show the effectiveness of the adopted preprocessing steps in enhancing the performance of the proposed COVID-19 prediction model.

5.2. *Impact of feature selection:*

In this subsection, we will discuss the effect of employing the proposed feature selection that is based on a fusion mechanism between the selected features from a correlation-based (PCC) perspective and the modified equilibrium-based (MEO) perspective. Hence, we will discuss first the application of the two adopted feature selection techniques, i.e., PCC and MEO, separately, and then compare to the performance of applying the fusion mechanism. In the solo application, PCC and MEO can be applied to the original feature pool $\mathscr{F}$ (original feature domain) or the sparse one $\mathscr{F}_{\mathrm{s}}$ (sparse domain), then after the selection decisions, the classification step can be applied to samples from $\mathscr{F}$ or $\mathscr{F}_s$. Hence, we have ten conditions to be studied, check Table 5, 6. 1) employing all features in $\mathscr{F}$, 2) employing all sparse features in sparse domain $\mathscr{F}_s$, 3) correlation-based selection in $\mathscr{F}$ and employing the same domain for performing training and testing, 4) correlation-based selection in $\mathscr{F}$ and performing training and testing in sparse domain, i.e., sparsified samples from $\mathscr{F}_s$, 5) correlation-based selection in $\mathscr{F}_s$ and employing the original features domain $\mathscr{F}$ for performing training and testing, 6) correlation-based selection in features original domain $\mathscr{F}$ and performing training and testing in sparse domain $\mathscr{F}_s$, 7) MEO-based selection in the features original domain $\mathscr{F}$ and employing the same domain $\mathscr{F}$ for performing training and testing, 8) MEO-based selection in features original domain performing training and testing in sparse domain, 9) MEO-based selection in sparse domain and employing the original features domain for performing training and testing. 10) MEO-based selection in features original domain and performing training and testing in sparse domain.

To take a look at the selected features in each case and get an indication of its importance in the prementioned ten conditions, Figs. 15, 16 indicates the AdaBoost feature importance in different conditions. As indicated, we can see how the sparse domain exploit the real importance of features compared to the original domain of features. Sparsity gives the highest importance to white blood cell count (WBC). Recently, WBC is considered as a prognostic indicator of COVID-19 (Li et al., 2021). In addition, in Table 5, 6, computitative validation results are indicated as a comparison between the prementioned conditions. As indicated, in COVID-specific dataset, Table 5,employing all features in the features original domain or sparse domain didn't introduce that high performance compared to the other conditions due to the existence of high correlated features, such as these couple of features, Neutrophils (NET, NE), Lymphocytes (LYT, LY), Monocytes (MOT, MO), Eosinophils (EOT, EO). On the other side, performing a correlation-based or MEO-based selection in sparse domain can show superior performance even with smaller number of features than that of original features domain. Hence, it is demonstrated that sparsifying features can exploit more details from samples, hence better performance can be achieved. For CBC dataset, Table 6, we can see that employing all features provides the best performance in sparse domain, but we can see the performance in features original domain is still competing, as well, due to employing less correlated features than that of COVID-specific dataset, see Fig. 7. However, even with employing 6 out of 13 features in the case of MEO-based feature selection in sparse domain, we can obtain accuracy of around 97 % as demonstrated. The introduced performance with CBC dataset competes the state-of-the-art performance by Shaban et al in (2021) which achieves accuracy of 97.6 % employing all features while the proposed prediction algorithm achieves accuracy of 99 % employing all features in sparse domain as indicated in Table 6. According to the demonstrated results in Table 5, 6, the proposed fused feature selection mechanism is introduced to combine the advantages of MEO and PCC-based feature selection. Hence, the most possible performance can be achieved with the least possible number of features. Fig. 17 indicates the classification reports of testing the proposed COVID diagnosis model based on the proposed fused selection method in both original domain and sparse domain. The fusion mechanism selects 13 features for COVID- specific dataset and 6 features for CBC dataset. These features later are entered to the proposed 1DCNN model for a final classification stage. As indicated from Fig. 17, the fusion mechanism in feature selection helps to enhance the performance of the proposed prediction algorithm compared to the results in Table 5, 6, especially when performing training and testing in sparse domain. In sparse domain, we achieved a testing accuracy of 98 %, 99 % for COVID-specific dataset and CBC dataset, respectively. Hence, the proposed COVID prediction algorithm adopts the fused selection mechanism for feature selection. In addition, the training and the testing processes are performed to the sparse samples in $\mathscr{F}_s$.

5.3. Impact of classification model:

In this subsection, we evaluate the proposed COVID-19 diagnosis model by comparing its 1DCNN model with other ML models, such as GNB, DT, ET, GBT, KNN, LR, RF, LSVM, RSVM, XGBoost, LDA, QDA, AdaBoost. A great review of these models can be found in (Tang et al., 2014). With the traditional ML techniques, we employed 10-fold stratified cross-validation to avoid the problem of overfitting. It is the best practice when developing a traditional ML model. Then, a grid search procedure is employed to find the best combination hyperparameters (e.g., learning rate, interaction depth) using AUC as reference measure. In Table 7, 8, computitative comparisons are indicated between the proposed 1DCNN model and the prementioned ML algorithms. Table 7 shows the testing results for COVID-specific dataset while Table 8 demonstrates the testing results for CBC dataset. As indicated, the performance of the proposed 1DCNN with the adopted fused feature selection mechanism shows superior performance compared to the other ML models, especially when the training and testing processes are performed to sparse samples. However, performing the training and testing processes in the original features domain still shows good results and, so far, better than the tradition ML methods. Moreover, we can see that AdaBoost and ET show good performance among the other traditional ML techniques. In Fig. 18, the training-validation performance is indicated, in terms of accuracy, for the proposed COVID-19 prediction algorithm, based on the introduced fusion-based feature selection and 1DCNN model, once for the original samples in $\mathscr{F}$, and another for the sparse samples in $\mathscr{F}_s$.

5.4. Comparison to the state-of-the-art

In this subsection, we compare the proposed COVID-19 prediction algorithm to other prediction methods from previous studies (Alakus & Turkoglu, 2020; AlJame et al., 2020; Brinati et al., 2020; Cabitza et al., 2021; Shaban et al., 2021), see Table 1 for their dependencies. The proposed ERLX method in AlJame et al. (2020), is the most similar algorthim to ours, especially in the data preprocessing steps, but it doesn't have clear feature selection mechanism and it employs ensemble of different ML techniques. The proposed HDS algorithm in Shaban et al. (2021) employed fuzzy inference engine and Deep Neural Network for their prediction scheme. The rest of studies (Brinati et al., 2020; Cabitza et al., 2021) didn't employ any feature selection algorithms and just employ ensemble of different ML algorithms for the classification task. On the other side, Alakus and Turkoglu (2020) introduce a new ensemble of different deep learning models. Hence, for the sake of fair comparison between these prementioned studies, we employed the same preprocessing steps and the same adopted dataset while keeping the other steps adopted by each study. In addition, we employed their available codes, unless there is no one. In Fig. 19, a comparison of testing the prementioned studies is indicated as classification reports. As shown, the proposed COVID-19 prediction model outperformes the state-of-the-art in both datasets, i.e., it achieves accuracy of 98 %, 99 % for COVID-specific and CBC datasets, respectively. In addition, this superior performance is achieved only with less than half the size of the available features/blood exams. Ours employs 6 out of 13 for CBC dataset and 13 out of 33 for COVID-specific dataset, while the other studies employs, mostly, all available features as in (Brinati et al., 2020; Cabitza et al., 2021, Shaban et al., 2021) or larger feature number than ours as in (Alakus & Turkoglu, 2020; AlJame et al., 2020).

6. Conclusion

In this paper, we proposed a novel COVID-19 prediction model based on routine blood tests. In this model, we exploited the benefits of sparsifying the feature pool to get the real dependencies between the employed blood tests. In this employed sparse domain, we succeeded to reduce the feature pool size to less than the half using the adopted hybrid feature selection scheme. This scheme fuses the elimination decisions of Pearson's correlation coefficient and a new Minkowski-based equilibrium optimizer. Then, with a deep convolutional model, we proved that the proposed algorithm can efficiently predict COVID-19 infection with small number of blood tests. Hence, scarce healthcare resources can be more effectively prioritized, especially in developing and low middle income countries. The major limitation in this study is training time of 1DCNN compared to the traditional ML techniques, but still PCR tests typically take hours to perform, and the target is to find alternative predictive models that still compete with accurate results to improve healthcare resource prioritization and inform patient care. Hence, in the future work, we intend to reduce the computational cost of the whole prediction algorithm, especially the training stage.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

We shared links of the employed datasets in the manuscript itself and we have cited their papers

References

Ai, T., Yang, Z., Hou, H., Zhan, C., Chen, C., Lv, W., … Xia, L. (2020). Correlation of chest CT and RT-PCR testing in coronavirus disease 2019 (COVID-19) in China: A report of 1014 cases. *Radiology*. https://doi.org/10.3410/f.737441336.793572890

Alafif, T., Tehame, A. M., Bajaba, S., Barnawi, A., & Zia, S. (2021). Machine and deep learning towards COVID-19 diagnosis and treatment: Survey, challenges, and future directions. *International Journal of Environmental Research and Public Health, 18*(3), 1117. https://doi.org/10.31224/osf.io/w3zxy.

Alakus, T. B., & Turkoglu, I. (2020). Comparison of deep learning approaches to predict COVID-19 infection. *Chaos, Solitons & Fractals, 140*, Article 110120. https://doi.org/10.1016/j.chaos.2020.110120

Albahri, O. S., Zaidan, A. A., Albahri, A. S., Zaidan, B. B., Abdulkareem, K. H., Al-Qaysi, Z. T., ... & Rashid, N. A. (2020). Systematic review of artificial intelligence techniques in the detection and classification of COVID-19 medical images in terms of evaluation and benchmarking: Taxonomy analysis, challenges, future solutions and methodological aspects. Journal of Infection and Public Health, 13(10), 1381-1396.https://doi.org/10.1016/j.jiph.2020.06.028.

Alballa, N., & Al-Turaiki, I. (2021). Machine learning approaches in COVID-19 diagnosis, mortality, and severity risk prediction: A review. *Informatics in Medicine Unlocked, 24*, Article 100564. https://doi.org/10.1016/j.imu.2021.100564

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. In 2017 international conference on engineering and technology (ICET) (pp. 1-6). IEEE. https://doi.org/10.1109/ICEngTechnol.2017.8308186.

Albu, A., Precup, R. E., & Teban, T. A. (2019). Results and challenges of artificial neural networks used for decision-making and control in medical applications. *Facta Universitatis, Series: Mechanical Engineering, 17*(3), 285–308. https://doi.org/10.22190/fume190327035a.

AlJame, M., Ahmad, I., Imtiaz, A., & Mohammed, A. (2020). Ensemble learning model for diagnosing COVID-19 from routine blood tests. *Informatics in Medicine Unlocked, 21*, Article 100449. https://doi.org/10.1016/j.imu.2020.100449

Altantawy, D. A., Saleh, A. I., & Kishk, S. S. (2020). Bi-perspective Fisher discrimination for single depth map upsampling: A self-learning classification-based approach. *Neurocomputing, 380*, 321–340. https://doi.org/10.1016/j.neucom.2019.08.074

Alves, M. A., Castro, G. Z., Oliveira, B. A. S., Ferreira, L. A., Ramírez, J. A., Silva, R., & Guimarães, F. G. (2021). Explaining machine learning based diagnosis of COVID-19 from routine blood tests with decision trees and criteria graphs. *Computers in Biology and Medicine, 132*, Article 104335. https://doi.org/10.1016/j.compbiomed.2021.104335

Banerjee, A., Ray, S., Vorselaars, B., Kitson, J., Mamalakis, M., Weeks, S., … Mackenzie, L. S. (2020). Use of machine learning and artificial intelligence to predict SARS-CoV-2 infection from full blood counts in a population. *International Immunopharmacology, 86*, Article 106705. https://doi.org/10.1016/j.intimp.2020.106705

Bao, J., Li, C., Zhang, K., Kang, H., Chen, W., & Gu, B. (2020). Comparative analysis of laboratory indexes of severe and non-severe patients infected with COVID-19. *Clinica Chimica Acta, 509*, 180–194. https://doi.org/10.1016/j.cca.2020.06.009

Borlea, I. D., Precup, R. E., Borlea, A. B., & Iercan, D. (2021). A unified form of fuzzy C-means and K-means algorithms and its partitional implementation. *Knowledge-Based Systems, 214*, Article 106731.

Bouwmans, T., Sobral, A., Javed, S., Jung, S. K., & Zahzah, E. H. (2017). Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset. *Computer Science Review, 23*, 1–71. https://doi.org/10.1016/j.knosys.2020.106731

Brinati, D., Campagner, A., Ferrari, D., Locatelli, M., Banfi, G., & Cabitza, F. (2020). Detection of COVID-19 infection from routine blood exams with machine learning: A feasibility study. *Journal of Medical Systems, 44*(8), 1–12. https://doi.org/10.1007/s10916-020-01597-4

Brink, H., Richards, J., & Fetherolf, M. (2016). *Real-world machine learning*. Simon and Schuster.

Bullock, J., Luccioni, A., Pham, K. H., Lam, C. S. N., & Luengo-Oroz, M. (2020). Mapping the landscape of artificial intelligence applications against COVID-19. *Journal of Artificial Intelligence Research, 69*, 807–845. https://doi.org/10.1613/jair.1.12162

Burog, A. I. L. D., Yacapin, C. P. R. C., Maglente, R. R. O., Macalalad-Josue, A. A., Uy, E. J. B., Dans, A. L., & Dans, L. F. (2020). Should IgM/IgG rapid test kit be used in the diagnosis of COVID-19. *Asia Pacific Center for Evidence Based Healthcare, 4*, 1–12. https://doi.org/10.47895/amp.v54i0.1558.

Cabitza, F., Campagner, A., Ferrari, D., Di Resta, C., Ceriotti, D., Sabetta, E., … Carobene, A. (2021). Development, evaluation, and validation of machine learning models for COVID-19 detection based on routine blood tests. *Clinical Chemistry and Laboratory Medicine (CCLM), 59*(2), 421–431. https://doi.org/10.1101/2020.10.02.20205070

Candès, E. J., Li, X., Ma, Y., & Wright, J. (2011). Robust principal component analysis? *Journal of the ACM (JACM), 58*(3), 1–37. https://doi.org/10.5772/38267

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357. https://doi.org/10.1613/jair.953

Chiang, H. S., Shih, D. H., Lin, B., & Shih, M. H. (2014). An APN model for Arrhythmic beat classification. *Bioinformatics, 30*(12), 1739–1746. https://doi.org/10.1093/bioinformatics/btu101

Corman, V. M., Landt, O., Kaiser, M., Molenkamp, R., Meijer, A., Chu, D. K., … Drosten, C. (2020). Detection of 2019 novel coronavirus (2019-nCoV) by real-time RT-PCR. *Eurosurveillance, 25*(3), 2000045. https://doi.org/10.17504/protocols.io.bb3qiqmw.

de Freitas Barbosa, V. A., Gomes, J. C., de Santana, M. A., Albuquerque, J. E. D. A., de Souza, R. G., de Souza, R. E., & dos Santos, W. P. (2021). Heg. IA: An intelligent system to support diagnosis of Covid-19 based on blood tests. Research on. *Biomedical Engineering*, 1–18. https://doi.org/10.37473/dac/10.1101/2020.05.14.20102533.

de Moraes, B. A. F., Miraglia, J. L., Donato, T. H. R., & Filho, A. D. P. C. (2020). COVID-19 diagnosis prediction in emergency care patients: a machine learning approach. https://www. medrxiv. org/content/medrxiv/early/2020/04/07/2020.04. 04.20052092. full. pdf. https://doi.org/10.1101/2020.04.04.20052092.

Dong, D., Tang, Z., Wang, S., Hui, H., Gong, L., Lu, Y., … Li, H. (2020). The role of imaging in the detection and management of COVID-19: A review. *IEEE Reviews in Biomedical Engineering, 14*, 16–29. https://doi.org/10.1109/iconat53423.2022.9725885

Fang, Y., Zhang, H., Xie, J., Lin, M., Ying, L., Pang, P., & Ji, W. (2020). Sensitivity of chest CT for COVID-19: Comparison to RT-PCR. *Radiology*. https://doi.org/10.1148/radiol.2020200432

Faramarzi, A., Heidarinejad, M., Stephens, B., & Mirjalili, S. (2020). Equilibrium optimizer: A novel optimization algorithm. *Knowledge-Based Systems, 191*, Article 105190. https://doi.org/10.1016/j.knosys.2019.105190

Fazel, M., Candes, E., Recht, B., & Parrilo, P. (2008, October). Compressed sensing and robust recovery of low rank matrices. In 2008 42nd Asilomar Conference on Signals, Systems and Computers (pp. 1043-1047). IEEE. https://doi.org/10.1109/acssc.2008.5074571.

Feng, C., Wang, L., Chen, X., Zhai, Y., Zhu, F., Chen, H., … Li, T. (2021). A novel artificial intelligence-assisted triage tool to aid in the diagnosis of suspected COVID-19 pneumonia cases in fever clinics. *Annals of Translational Medicine, 9*(3). https://doi.org/10.1101/2020.03.19.20039099

Ferrari, D., Motta, A., Strollo, M., Banfi, G., & Locatelli, M. (2020). Routine blood tests as a potential diagnostic tool for COVID-19. *Clinical Chemistry and Laboratory Medicine (CCLM), 58*(7), 1095–1099. https://doi.org/10.1515/cclm-2020-0398

Gao, Y., Li, T., Han, M., Li, X., Wu, D., Xu, Y., … Wang, L. (2020). Diagnostic utility of clinical laboratory data determinations for patients with the severe COVID-19. *Journal of Medical Virology, 92*(7), 791–796. https://doi.org/10.1002/jmv.25770

Halim, A. H., Ismail, I., & Das, S. (2021). Performance assessment of the metaheuristic optimization algorithms: An exhaustive review. *Artificial Intelligence Review, 54*(3), 2323–2409. https://doi.org/10.1007/s10462-020-09906-6

Halko, N., Martinsson, P. G., & Tropp, J. A. (2009). Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions *http:// arxiv. org/abs/0909.4061. oai: arXiv. org, 909*. https://doi.org/10.1137/090771806.

Hedrea, R. C. R., & Petriu, E. M. (2021). Evolving fuzzy models of shape memory alloy wire actuators. *Science and Technology, 24*(4), 353–365.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science, 313*(5786), 504–507. https://doi.org/10.1126/science.1127647

Huang, C., Wang, Y., Li, X., Ren, L., Zhao, J., Hu, Y., … Cao, B. (2020). Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. *The Lancet, 395* (10223), 497–506. https://doi.org/10.3410/f.737258313.793572876

Ibrahim, D. M., Elshennawy, N. M., & Sarhan, A. M. (2021). Deep-chest: Multi-classification deep learning model for diagnosing COVID-19, pneumonia, and lung cancer chest diseases. *Computers in Biology and Medicine, 132*, Article 104348. https://doi.org/10.1016/j.compbiomed.2021.104348

Imran, A., Posokhova, I., Qureshi, H. N., Masood, U., Riaz, M. S., Ali, K., … Nabeel, M. (2020). AI4COVID-19: AI enabled preliminary diagnosis for COVID-19 from cough samples via an app. *Informatics in Medicine Unlocked, 20*, Article 100378. https://doi.org/10.1016/j.imu.2020.100378

Joshi, R. P., Pejaver, V., Hammarlund, N. E., Sung, H., Lee, S. K., Furmanchuk, A. O., … Banaei, N. (2020). A predictive tool for identification of SARS-CoV-2 PCR-negative emergency department patients using routine test results. *Journal of Clinical Virology, 129*, Article 104502. https://doi.org/10.1016/j.jcv.2020.104502

Khan, A. A., Laghari, A. A., & Awan, S. A. (2021). Machine learning in computer vision: A review. In *EAI Transactions on Scalable Information Systems* (p. e4).

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., & Inman, D. J. (2021). 1D convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing, 151*, Article 107398. https://doi.org/10.1016/j.ymssp.2020.107398

Kukar, M., Gunčar, G., Vovko, T., Podnar, S., Černelč, P., Brvar, M., … Notar, M. (2021). COVID-19 diagnosis by routine blood tests using machine learning. *Scientific Reports, 11*, Article 10738. https://doi.org/10.1038/s41598-021-90265-9

Kumbure, M. M., Lohrmann, C., Luukka, P., & Porras, J. (2022). Machine learning techniques and data for stock market forecasting: A literature review. *Expert Systems with Applications, 116659*. https://doi.org/10.1016/j.eswa.2022.116659

Langer, T., Favarato, M., Giudici, R., Bassi, G., Garberi, R., Villa, F., ... & Fumagalli, R. (2020). Use of Machine Learning to Rapidly Predict Positivity to Severe Acute Respiratory Syndrome Coronavirus 2(SARS-COV-2) Using Basic Clinical Data. 2020-05-01]. https://www. researchsquare. com/article/rs-38576/v1. https://doi.org/10.21203/rs.3.rs-38576/v1.

Latif, S., Usman, M., Manzoor, S., Iqbal, W., Qadir, J., Tyson, G., … Crowcroft, J. (2020). Leveraging data science to combat covid-19: A comprehensive review. *IEEE Transactions on Artificial Intelligence, 1*(1), 85–103. https://doi.org/10.36227/techrxiv.12212516.v2.

Li, D., Wang, D., Dong, J., Wang, N., Huang, H., Xu, H., & Xia, C. (2020). False-negative results of real-time reverse-transcriptase polymerase chain reaction for severe acute respiratory syndrome coronavirus 2: Role of deep-learning-based CT diagnosis and insights from two cases. *Korean Journal of Radiology, 21*(4), 505–508. https://doi.org/10.3348/kjr.2020.0146

Li, J., Wang, L., Liu, C., Wang, Z., Lin, Y., Dong, X., & Fan, R. (2021). Exploration of prognostic factors for critical COVID-19 patients using a nomogram model. *Scientific Reports, 11*(1), 1–6. https://doi.org/10.1038/s41598-021-87373-x

Li, W. T., Ma, J., Shende, N., Castaneda, G., Chakladar, J., Tsai, J. C., … Ongkeko, W. M. (2020). Using machine learning of clinical data to diagnose COVID-19: A systematic review and meta-analysis. *BMC Medical Informatics and Decision Making, 20*(1), 1–13. https://doi.org/10.21203/rs.3.rs-903246/v1.

Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008, December). Isolation forest. In 2008 eighth ieee international conference on data mining (pp. 413-422). IEEE. https://doi.org/10.1109/ICDM.2008.17.

Mohammad-Rahimi, H., Nadimi, M., Ghalyanchi-Langeroudi, A., Taheri, M., & Ghafouri-Fard, S. (2021). Application of machine learning in diagnosis of COVID-19 through X-ray and CT images: A scoping review. *Frontiers in Cardiovascular Medicine, 8*, 185. https://doi.org/10.3389/fcvm.2021.638011

Porcino, T., Rodrigues, E. O., Bernardini, F., Trevisan, D., & Clua, E. (2022). Identifying cybersickness causes in virtual reality games using symbolic machine learning algorithms. *Entertainment Computing, 41*, Article 100473. https://doi.org/10.1016/j.entcom.2021.100473

Schuller, B. W., Schuller, D. M., Qian, K., Liu, J., Zheng, H., & Li, X. (2021). COVID-19 and computer audition: an overview on what speech & sound analysis could contribute in the SARS-CoV-2 corona crisis. *Frontiers in Digital Health, 3*, 14. https://doi.org/10.3389/fdgth.2021.564906

Sethuraman, N., Jeremiah, S. S., & Ryo, A. (2020). Interpreting diagnostic tests for SARS-CoV-2. *Jama, 323*(22), 2249–2251. https://doi.org/10.1001/jama.2020.8259

Shaban, W. M., Rabie, A. H., Saleh, A. I., & Abo-Elsoud, M. A. (2021). Detecting COVID-19 patients based on fuzzy inference engine and Deep Neural Network. *Applied Soft Computing, 99*, Article 106906. https://doi.org/10.1016/j.asoc.2020.106906

Soares, F. (2020). A novel specific artificial intelligence-based method to identify COVID-19 cases using simple blood exams. MedRxiv, https://doi.org/10.1101/2020.04.10.20061036; 2020. https://doi.org/10.1101/2020.04.10.20061036.

Sun, N. N., Yang, Y., Tang, L. L., Dai, Y. N., Gao, H. N., Pan, H. Y., & Ju, B. (2020). A prediction model based on machine learning for diagnosing the early COVID-19 patients. medRxiv. https://doi.org/10.1101/2020.06.03.20120881.

Suspected COVID-19 pneumonia Diagnosis Aid System (2021). Retrieved from https://intensivecare.shinyapps.io/COVID19/. Accessed June 11, 2022.

Tan, G. W. H., Ooi, K. B., Leong, L. Y., & Lin, B. (2014). Predicting the drivers of behavioral intention to use mobile learning: A hybrid SEM-Neural Networks approach. *Computers in Human Behavior, 36*, 198–213. https://doi.org/10.1016/j.chb.2014.03.052

Tang, J., Alelyani, S., & Liu, H. (2014). Data classification: Algorithms and applications. *Data Mining and Knowledge Discovery Series*, 37–64.

Upadhyay, P. K., & Nagpal, C. (2020). Wavelet based performance analysis of SVM and RBF kernel for classifying stress conditions of sleep EEG. *Science and Technology, 23* (3), 292–310.

Worldometer (2020). Retrieved from https://www.worldometers.info/coronavirus/. Accessed June 11, 2022.

Wu, G., Zhou, S., Wang, Y., & Li, X. (2020). Machine learning: a predication model of outcome of SARS-CoV-2 pneumonia. 2020. https://doi.org/10.21203/rs.3.rs-23196/v1.

Wu, J., Zhang, P., Zhang, L., Meng, W., Li, J., Tong, C., … Li, S. (2020). Rapid and accurate identification of COVID-19 infection through machine learning based on clinical available blood test results. *MedRxiv*. https://doi.org/10.1101/2020.04.02.20051136

Yan, L., Zhang, H. T., Xiao, Y., Wang, M., Sun, C., Liang, J., … Yuan, Y. (2020). Prediction of criticality in patients with severe Covid-19 infection using three clinical features: A machine learning-based prognostic model with clinical data in Wuhan. *MedRxiv, 27*. https://doi.org/10.1101/2020.02.27.20028027

Yang, H. S., Hou, Y., Vasovic, L. V., Steel, P. A., Chadburn, A., Racine-Brzostek, S. E., … Wang, F. (2020). Routine laboratory blood tests predict SARS-CoV-2 infection using machine learning. *Clinical Chemistry, 66*(11), 1396–1404. https://doi.org/10.1101/2020.06.17.20133892

Ye, Z., Zhang, Y., Wang, Y., Huang, Z., & Song, B. (2020). Chest CT manifestations of new coronavirus disease 2019 (COVID-19): A pictorial review. *European Radiology, 30*(8), 4381–4389. https://doi.org/10.1007/s00330-020-06801-0

Zhang, Z., Feng, Q., Huang, J., Guo, Y., Xu, J., & Wang, J. (2021). A local search algorithm for k-means with outliers. *Neurocomputing, 450*, 230–241. https://doi.org/10.1016/j.neucom.2021.04.028

Zhou, T., & Tao, D. (2011, October). Godec: Randomized low-rank & sparse matrix decomposition in noisy case. In Proceedings of the 28th International Conference on Machine Learning, ICML 2011.

Zimmermann, K., & Mannhalter, J. W. (1996). Technical aspects of quantitative competitive PCR. *BioTechniques, 21*(2), 268–279. https://doi.org/10.2144/96212rv01

Zoabi, Y., Deri-Rozov, S., & Shomron, N. (2021). Machine learning-based prediction of COVID-19 diagnosis based on symptoms. *npj Digital Medicine, 4*(1), 1–5. https://doi.org/10.1038/s41746-020-00372-6