Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

CHAPTER

Comparative analysis of various supervised machine learning techniques for diagnosis of COVID-19

25

Pijush Dutta[1], **Shobhandeb Paul**[2], **Asok Kumar**[3]

[1]*Department of Electronics and Communication Engineering, Global Institute of Management and Technology, Krishnagar, West Bengal, India;* [2]*Department of Electronics and Communication Engineering, Guru Nanak Institute of Technology, Panihati, West Bengal, India;* [3]*Student Welfare Department, Vidyasagar University, Medinipur, West Bengal, India*

1. Introduction

Coronavirus disease 2019 (COVID-19) is the third virus infection in 2 decades after severe acute respiratory syndrome and the Middle East respiratory syndrome. As these manifestations did not recognize before is the reason it is known as a novel Coronavirus. The cases were seen by different nations from November 2019 to the work date. The ailment influences lungs and causes respiratory ailment with manifestations like seasonal influenza, for example, chills, throat contamination, cough, fever, and in critical cases, difficulty in breathing. The dynamic time of the novel Coronavirus is 14 days. It is proposed by clinical specialists that one can secure himself/herself by washing hands frequently, abstaining from contacting the nose, ears, and face, and by keeping up social distancing (1 m or 3 feet away with others).

Several literature surveys have been conducted on machine learning applied in the prediction of COVID-19 all around the globe. Two different machine learning models SIR and SEIR are based on the Akaike Information Criterion of COVID-19 [1]. From the experimental results, it has been seen that predictions of a confirmed case for SIR model performs much better than an SEIR model. A simplified weather model of a zone for COVID-19 spread is done by taking the input parameters of latitude, temperature, and humidity at increased risk [2]. From the experimental result, it has been that this weather modeling can efficiently predict the higher risk regions of significant community spread of COVID-19. A predictive model is in the Corona Tracker community to predict and forecast COVID-19 cases, deaths, and recoveries [3]. There is a live estimating exercise with monstrous possible ramifications for planning and decision-making forecasts for the affirmed cases of COVID-19 [4]. A programmed location framework is by utilizing three diverse convolution neural system based models (ResNet50, InceptionV3, and Inception-ResNetV2) for the discovery of Coronavirus pneumonia tainted patients utilizing chest X-beam radiographs [5]. From the result, it has been seen that the model gives the most noteworthy order execution agreeably.

Electronic Devices, Circuits, and Systems for Biomedical Applications. https://doi.org/10.1016/B978-0-323-85172-5.00020-4

A smartphone-based app has been proposed to combining the symptoms to predict probable infection data collected from all the app users [6]. A simple mean-field model is for gathering a quantitative picture of the epidemic spreading, time of the peak of confirmed infected individuals, and number infected-recovered-deaths used in a three-country region: China, Italy, and France [5]. An artificial intelligence's profound learning techniques separate COVID-19's particular graphic highlights and give a clinical analysis in front of the pathogenic test, hence sparing crucial time for ailment control [8]. Results show that the result is extremely compelling.

A deep learning model is utilized for recognizing COVID-19 pneumonia on high-resolution X-rays [9]. Tentatively, it has been seen this model relieves the working pressure of radiologists to the control the plague. Various viewpoints including prediction outcomes, disease tracking, medical image processing, computational biology, and medicines to battle against the COVID-19 crisis [10]. A predictive machine learning tool selected three biomarkers to identify crucial disease mortality [11]. Results show a 90% accurate model. An artificial intelligence (AI) algorithm was designed to incorporate chest CT findings with clinical indications to quickly analyze patients who are positive for COVID-19 [6]. There are three different machine learning techniques: support vector machine, artificial neural networks, and regression model to predict the COVID-19 patient's recovery [13]. From the result, it has been seen that the patient's fever, cough, general fatigue, and most probably malaise could not recover from COVID-19.

A deep learning neural network machine learning method predicts confirmed-negative-released-death cases of COVID-19 very close to original data. The performance analysis shows a high rate of accuracy [14]. Machine learning and deep learning models are used for understanding the behavior of future reachability of the COVID-19 across the nations [15]. A cutting edge investigation of the progression of machine learning (ML) and deep learning techniques in the determinations and forecast of COVID-19 was performed [7]. Five ML approaches, logistic regression, partial least squares regression, elastic net, random forest, and bagged flexible discriminated analysis were used to predict outcomes of COVID-19 patients [8]. The sensitivity and specificity for the derivation set and validation set were quite satisfying.

A relative investigation was done of ML and soft computing models to foresee the COVID-19 outbreak as an option in contrast to SIR and SEIR models [9]. There were two different numeric strategies: long short-term memory and curve fitting for the forecast of the number of recovered cases and positives cases of COVID-19 cases in India ahead of 30 days [19]. The experimental result also emphasized how many precautions like social isolation and lockdown affect the spread of COVID-19. Machine learning is a part of artificial intelligence is often discovering the pattern of the user data in order to predict the value of the new datasets for which the target value is unknown [20]. The most common task in the learning process is classification. The main task of classification is to classify the data into definite classes.

Since COVID-19 symptoms are similar to that of a nasty cold or flu, there has been a rush in the number of people checking into a doctor's clinic, leading to panic. Before people go to a doctor or medical center, experts say that it is best to practice self-quarantine and then take necessary action. The symptoms that you should be primarily be looking for include breathing difficulties, muscle pain, respiratory infection, high-grade fever, headache or mild conjunctivitis, and pneumonia.

The main objective of this study is to present a generic feature from the input data and apply supervised ML to reduce the generalization error for achieving a more accurate diagnosis. The overall presentation is summarized as follows:

- Identify the important generic features from the input datasets and process them in the ML algorithm to predict COVID-infected persons rather than through the traditional healthcare system.

- Multiple numbers of ML algorithms (bagging algorithm, k-nearest neighbor, and random forest) are applied on the same processing patient datasets.
- Our work exterminates the need to relook at existing algorithms for handing COVID-19 patient data.

The rest of the paper is organized as follows: Section 2 gives the proposed problem formulation containing the risk factor for COVID-19 and precedes datasets. Section 3 describes the ML techniques utilized in this research. Section 4 presents the experimental results followed by the conclusion and future work.

2. Problem formulation

There are several known risk factors for the COVID-19 pandemic. Nonetheless, most instances of COVID-19 cannot be connected to a particular reason. The incubation period refers to the time it takes for a patient to catch the virus and begin to show the symptoms. In this period, experts say that suspected COVID-19 patients can develop any or one of the symptoms related to the viral disease. It has been presumed since the infection spreads through respiratory droplets in the air that it can take anywhere between 3 and 14 days for the symptoms to fully infect, with more prominent symptoms starting to appear around day 5. A basic simplified ML model is shown in Fig. 25.1.

2.1 Data sets description

To build the model we first needed data. We collected data from the website www.covid19india.org during data sets taken from January 31 to March 11, 2020 and started working on this in early

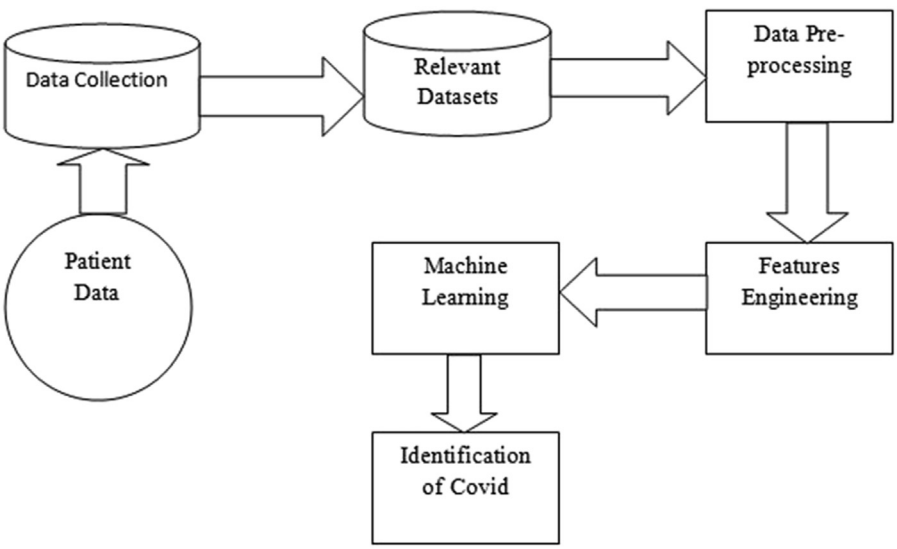

FIGURE 25.1

Simplified ML model for COVID-19.

March when India has already entered into phase II, i.e., in local transmission. We got data on symptoms like age, fever at nominal temperature, cough, headache, mild conjunctivitis, chest congestion, and high fever (these are the major symptoms). Before we could start building the algorithms, we needed to wait until all the patients recovered or died, so we would know the outcome of their cases. For this experiment, we took 64 datasets for training purposes.

2.2 Data analysis

Cold, fever, cough, body pain, and malaise: these five were the most common potential symptoms for COVID, as shown in Table 25.1. Figs. 25.2, 25.3, and 25.4 showed the frequency of different attributes of the datasets like sex, age, and symptoms for COVID 19. For the data visualization and plotting, we have used the Matplotlib and Seaborn. Python data visualization library provides a high-level interface for drawing attractive and informative statistical graphics for COVID-19 patients, as shown in Fig. 25.5.

Table 25.1 Data description.

Column	Description	Categorical value	Type
Id	Patient individual id	NA	Numeric
Gender	Patient's gender	male or Female	String category
Age	Patient's age	NA	Numeric
COVID-19	Patient's nature	Yes (=1) or No (=0)	Numeric
Symptom 1 Symptom 2 Symptom 3 Symptom 4 Symptom 5	Symptoms noticed by the patients	Multiple symptoms noticed by the patients	String category

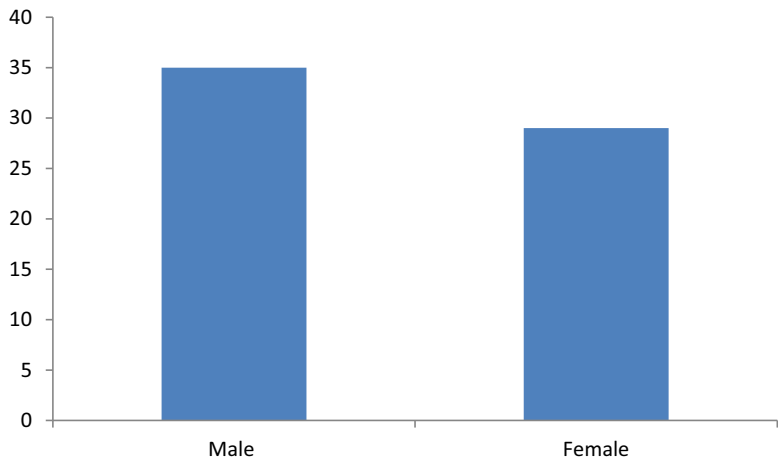

FIGURE 25.2

Sex attributes for COVID-19.

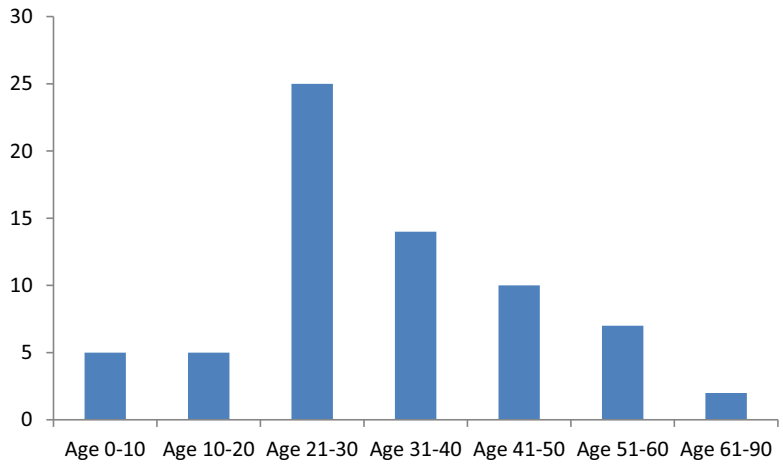

FIGURE 25.3

Age attributes for COVID-19.

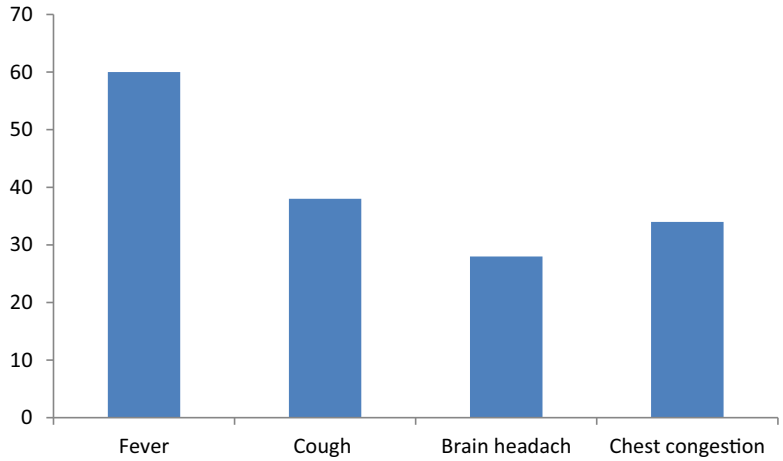

FIGURE 25.4

Different symptoms for COVID-19.

From the figures, we can easily visualize the cases that we have from the binary datasheet. The diagonal values show the correctly predicted value that gives the accuracy of the training data set (divided in the ratio of 70:30). Fig. 25.6 gives the heat map (normalized confusion matrix) representation of the binary datasheet. Here, we can visualize the data as a darker value gives the more correct value of each symptom or the parameters in this case, and the rest of the blocks show the negative values that are quite a good result for our analysis, as we are focused on analyzing symptoms as 0 and 1, and then the final output of the result, and this would give the model flexibility to train itself in a better way, thus producing the expected test results, which could be used to determine the model's accuracy.

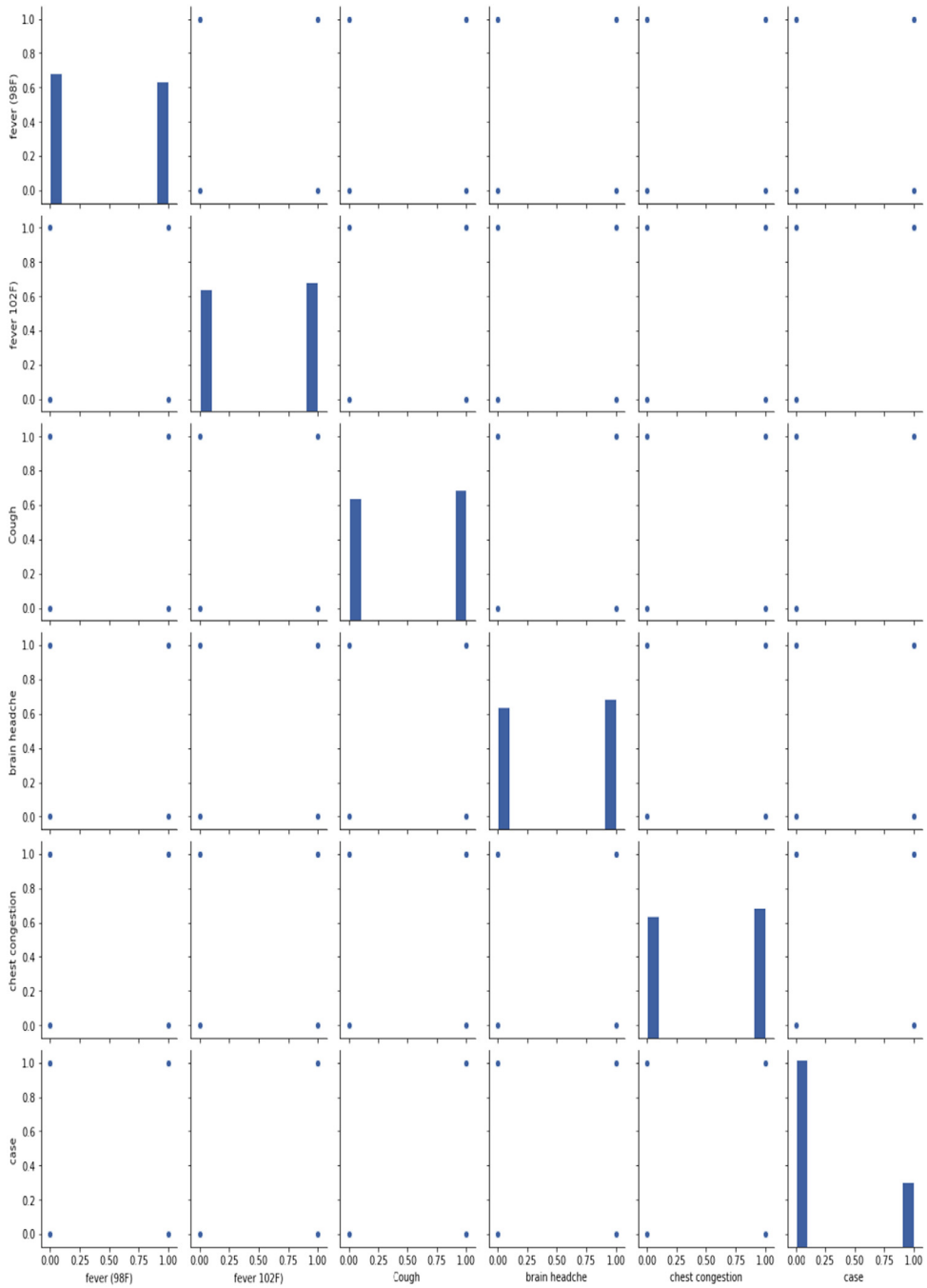

FIGURE 25.5

Data visualization and plotting of COVID-19 parameters.

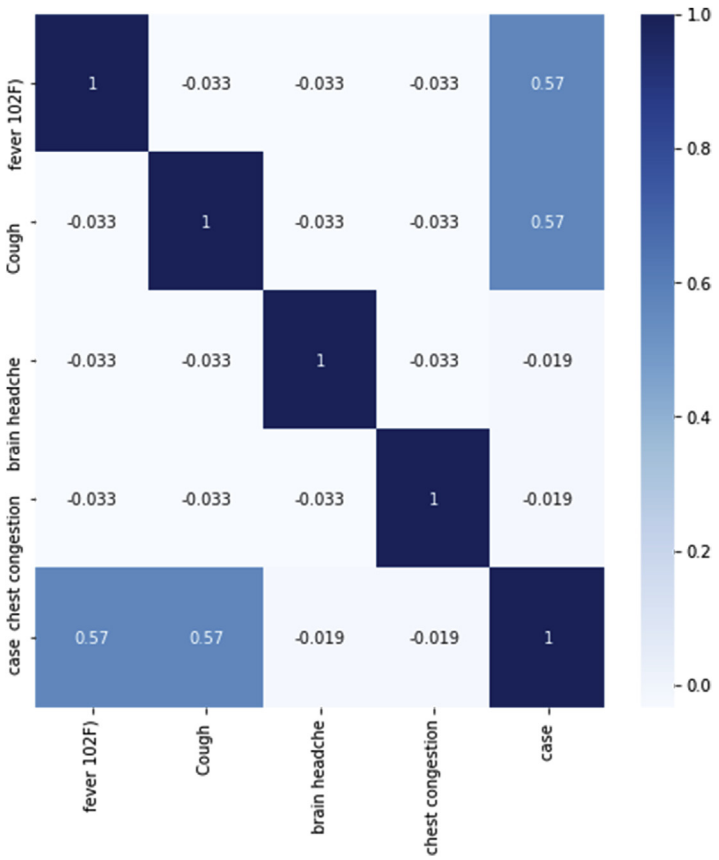

FIGURE 25.6

Heat map of the binary datasets.

2.3 Data preprocessing

In the data preparation section, the dataset consists of columns named like date, string, and numeric type as well as some categorical variables. In the data preprocessing section, we convert all the datasets into numeric form, so in the next stage, we can apply the different ML classification techniques. This should be possible by replacing a number to each special unmitigated value in the column. The dataset comprises various missing values supplanted by "NA." In certain patient datasets, "death" and "recovery" columns contain missing values; they have been isolated from the primary dataset and used as a test dataset, while the rest of the records have been used as training dataset for designing the ML classifier. The dataset also consists of data columns, which are not for direct use, and feature engineering has been applied.

2.4 Evaluation metrics

The analytic part of any classifier has most of the parts controlled by the receiver operating characteristic (ROC) curve and confusion matrix [10]. In the field of AI and ML, the confusion matrix is also known as the error matrix. The fundamental diagram of the confusion matrix has been shown in Fig. 25.7A, where true positive (TP) and true negative (TN) are the positive and negative cases where the classifier correctly identified them. False positive (FP) are the negative cases where the classifier incorrectly identified them as positive, and the false negative (FN) are the positive cases where the classifier incorrectly identified them as negative. By using these properties, supervised ML measures performance of a classifier as recall, precision, accuracy, and f1-score, represented in Eqs. (25.1)–(25.4).

TP: Cases when classifier predicted TRUE (they have the disease), and the correct class was TRUE (patient has a disease).

TN: Cases when classifier predicted FALSE (no disease), and the correct class was FALSE (patient does not have the disease).

FP (Type 1 error): Cases when classifier predicted TRUE (they have the disease), and the correct class was FALSE (patient does not have the disease).

FN (Type 2 error): Cases when classifier predicted FALSE (no disease), but the correct class was TRUE (patient does have the disease).

$$\text{Classification accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{25.1}$$

$$\text{Misclassication rate(Error Rate)} = \frac{(FP + FN)}{(TP + TN + FP + FN)} \tag{25.2}$$

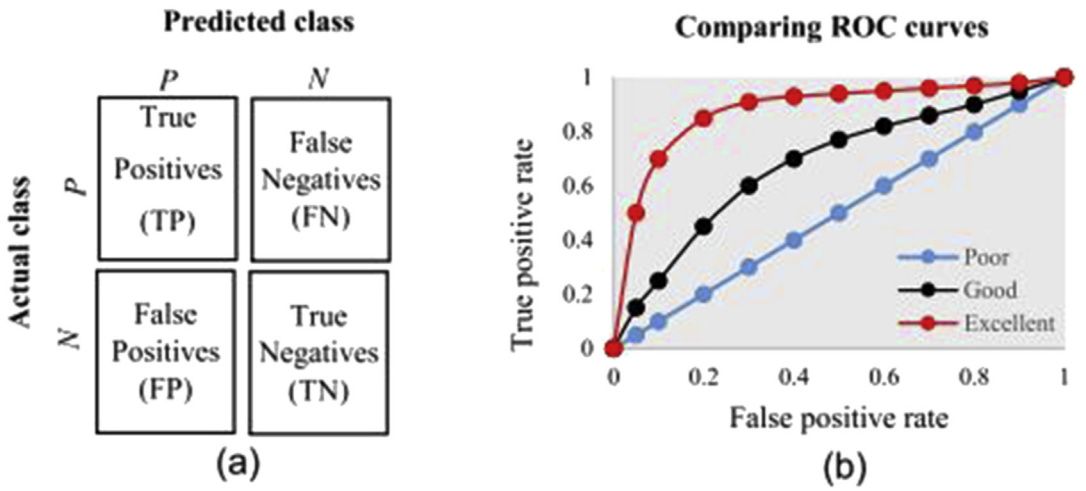

FIGURE 25.7

Basic framework of confusion matrix and ROC [11].

$$\text{Precision} = \text{TP}/\text{Total TRUE Predictions} = \frac{TP}{(TP + FP)} \quad (25.3)$$

(When model predicted TRUE class, how often was it right?)

$$\text{Recall} = \text{TP}/\text{Actual TRUE} = \frac{TP}{(FN + FP)} \quad (25.4)$$

(When the class was actually TRUE, how often did the classifier get it right?)

The values of the *TP*, *TN*, *FP*, and *FN* can be collected from the heat map and can be substituted to get the table for precision, recall, F1-score, and support (misclassification rate).

A ROC is a graphic measuring tool plotting the *TP* rate against the *FP* rate at different threshold points [10]. The predictability of any classifier is determined by the area under the curve (AUC) of ROC. A higher value of AUC indicates more superiority of a classifier. Fig. 25.7B represents ROC curves based on a theoretical dataset. The area under the blue ROC curve is half of the concealed square. Thus, the AUC under the blue ROC curve is 0.5. The area under the black ROC curve is occupied more prominently than half of the shaded rectangle, so it is better than the blue ROC. Red ROC occupied the maximum area of the shaded rectangle; that is why red ROC is superior to the other two previous classifiers.

3. ML

For the most part, two kinds of AI are applied in various fields: supervised and unsupervised ML. In the supervised learning algorithm, both the calculated and target output datasets are given. Info and yield information are marked for the arrangement to give a learning premise to future information preparation. Supervised ML can be further categorized into two parts: regression and classification issues. A regression analysis dataset may have discrete or continuous values, while in classification problem, the output variable is two discrete values, 0 or 1. There is a notable ML algorithm used in different medical fields, as show in Table 25.2.

In an unsupervised ML algorithm, the datasets are not perfectly organized and allow the computation to catch up on that information without any direction. In this research, the outcome variable is a dependent variable: either COVID-19 positive or negative. That is two discrete levels, so we utilized a classification algorithm of supervised learning. In the present research, we have used three different types of classification algorithms in ML:

1. nearest neighbor
2. random forest classification
3. bagging algorithm

3.1 kNN (k-nearest neighbor)

k-Nearest neighbor classification is the closest neighbor classifiers that depends on the possibility that an object ought to be anticipated to have a place with a similar class as the items in the preparation set with the greatest likeness [32,33]. To utilize kNN grouping, it is required to have an appropriate comparability search framework in which the preparation information is put away. The arrangement is

Table 25.2 Notable ML methods for outbreak prediction.

Si No	Year published	Paper	Outbreak infection	Algorithm used
1	June2020	[12]	COVID-19	Logistic regression, decision tree, random forest, support vector machine (SVM), AdaBoost and stochastic gradient boosting multinomial Naïve Bayes (MNB), bagging
2	April 2020	[13]	COVID-19	Decision tree, random forest, XGBoost, AdaBoost, bagging, and light GBM
3	July 2020	[14]	COVID-19	random forest classification
4	June 2020	[15]	COVID-19	random forest, k-nearest neighbor, Naïve Bayes, logistic regression, decision tree, support vector machine (SVM)
5	December 2019	[16]	Breast cancer	Support vector machine, neural network, logistic regression, linear discriminant analysis, Naïve Bayes, decision tree
6	2017	[17]	Cerebral infarction	Decision tree, k-nearest neighbor, Naïve Bayes
7	2018	[18]	Diabetes	Decision tree, Naïve Bayes, support vector machine
8	2018	[19]	Heart disease	k-Nearest neighbor, random forest, support vector machine
9	2018	[20]	Heart disease	Linear regression, random forest
10	2017	[21]	Heart disease	Logistic regression, random forest
11	2018	[22]	Heart disease	Logistic regression, support vector machine
12	2017	[23]	Heart disease	k-Nearest neighbor, Naïve Bayes, random forest, support vector machine
13	2017	[24]	Heart disease	Logistic regression, random forest
14	2017	[25]	Heart disease	Decision tree, k-nearest neighbor, Naïve Bayes, random forest, support vector machine
15	2018	[26]	Heart disease	Neural network, Naïve Bayes, support vector machine
16	2017	[27]	Heart disease	Decision tree, k-nearest neighbor, Naïve Bayes,
17	2018	[28]	Liver disease	Neural network, support vector machine, logistics regression, random forest
18	2017	[29]	Lung cancer	Support vector machine, decision tree, random forest
19	2018	[30]	Prostate cancer	Naïve Bayes, decision tree, support vector machine
20	2013	[31]	Type 2 diabetes	Support vector machine, multifactor dimensionality reduction, k-nearest neighbor, logistic regression

finished by investigating the consequences of a kNN inquiry. The least complex approach to decide a characterization after the effect of a kNN classifier is the dominant part rule. The objects in the question result are meant each class and the class having the dominant part check is anticipated to be the class of the test object. Another technique is to think about the separations of the item to gauge the effectiveness of each neighboring object. In this manner, a nearby object contributes more to the choice than an object having a huge separation. kNN classifiers utilize the class fringes of the items inside the preparation set, and along these lines, we need not bother with any preparation or model structure apriority. Therefore, kNN classifiers cannot be utilized to increase unequivocal class information to

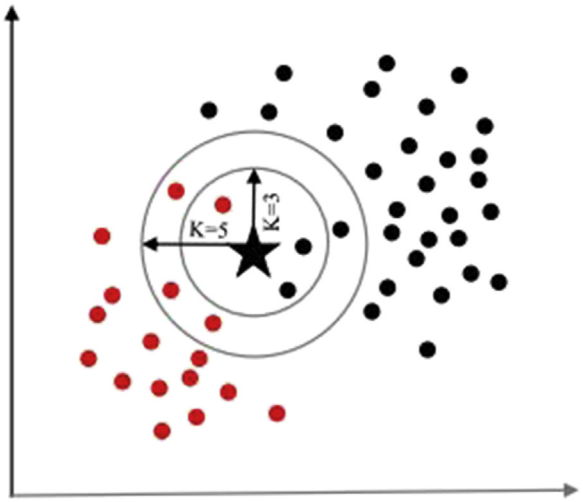

FIGURE 25.8

A simplified illustration of the kNN [11].

dissect the structure of the classes. kNN grouping is otherwise called a lazy learning algorithm. The boundary that decides the neighborhood size, k, is critical to the order precision accomplished by a kNN classifier. If k is picked excessively small, the order will in general be extremely delicate to clamor and anomalies. Then again, a too huge incentive for k may expand the outcome set of the k closest neighbor by objects that are excessively far away to be like the classified object. Since kNN classification works on preparation information, the characterization time relies upon the proficiency of the basic closeness search framework. On account of enormous preparing datasets, a direct hunt turns out to be wasteful. Appropriate record structures can offer a superior solution for the model [32, 34]. Erasing irrelevant objects from the preparation dataset likewise helps in accelerating the kNN arrangement calculation [32]. kNN classifier algorithm can be accelerated by building the centroid for the objects of each class and utilizing just the centroids and closest neighbors characterization [35]. This methodology is fairly basic. However, it gives a precise order for text information. In Fig. 25.8, it is seen that $K = 3$, and $K = 5$ are classifiers in which a sample object ("star") is classified as "black" or "red" depending upon the number of votes it gets.

3.2 Random forest (RF)

A random forest is a supervised ML classifier that comprises a treelike structure $\{h(x, (k)\ k = 1, 2, \ldots.\}$, unique independent vector $\{\theta(k)\}$, and input for most famous class of x [36–38]. In random forest, to produce each single tree, researcher Breiman followed the following advances. In the bootstrap test, in N number of preparation datasets, N number of records are examined aimlessly by using substitution from the first information. This is the first stage for developing the tree. On the off chance that there are M input factors, a number $m << M$ is chosen with the end goal that at every node, m factors are chosen at random out of M, and the best split on these m credits is utilized to part of

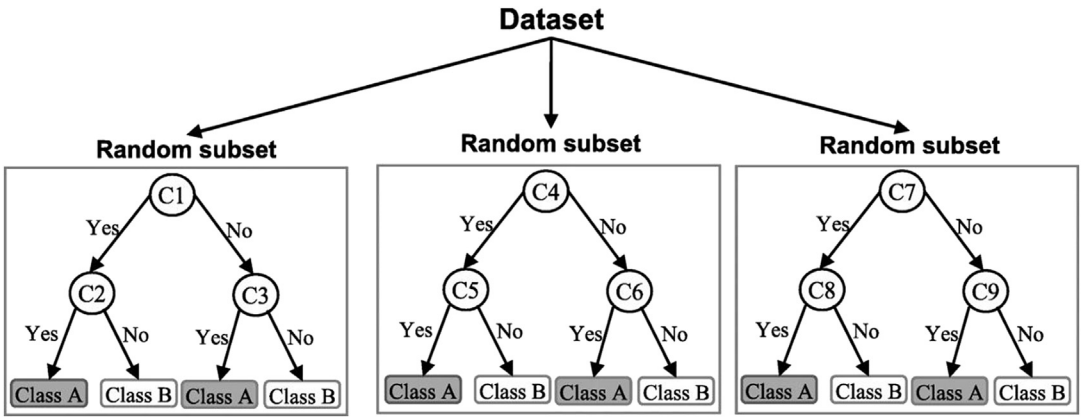

FIGURE 25.9

An illustration of a random forest [11].

the node. The estimation of m is held steady during forest development. Each tree is developed to the largest degree of divide. Along these lines, various trees are actuated in the forest; the quantity of trees is prechosen by the boundary Ntree. The number of factors (m) chosen at every hub is likewise alluded to as mtry or k in the literature (Fig. 25.9).

The profundity of the tree can be constrained by a boundary node size (for example, number of occurrences in the leaf node), which is generally set to 1. When the woodland is prepared or worked as clarified above, to arrange another occasion, it is stumbled into all the trees developed in the backwoods. Each tree is divided into several cases that are recorded as a vote. The votes from all trees are accumulated, and the class that has the maximum number of votes is assigned as the grouping of the new case.

3.3 Bagging algorithm

Bagging is a technique for making classifier gatherings, comparable by the idea, yet with essential contrasts [40]. Packing was proposed by Breiman [41] and stretched out further by Arcing [42,43] to oblige the versatile steady development of the group that underlies the boosting technique. Bagging makes the classifiers in the gathering by taking irregular samples with substitution [44] from the informational index and building one classifier on each bootstrap test. The last arrangement choice for an unlabeled information point x is made by taking the majority share vote over the class names delivered by the L classifiers. The genuine quality of bagging is for precarious classifiers, for example, neural systems and choice trees. Shaky classifiers are sensitive to little changes in the informational index. In this way, preparing a similar classifier model on two somewhat extraordinary prepared sets may bring about considerably various classifiers. The classifiers may be comparable by and large in exactnesses, yet the boundaries like weights of the neural network will vary, prompting a characteristic troupe decent variety. Ideally, this diversity will show up by the two classifiers perceiving effectively various articles from the informational collection, i.e., having “ability” in various locales in the component space. Bootstrap testing is utilized to give the irregular little adjustments of the

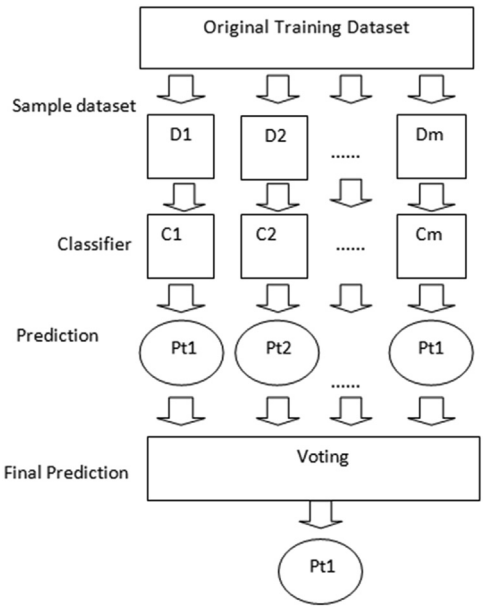

FIGURE 25.10

An illustration of a bagging algorithm.

informational collection. Stowing has been seen as wasteful for direct classifiers prepared on enormous informational collections, as these are steady classifiers [45, 46]. This implies if direct classifiers (e.g., the closest mean classifier, NMC) are prepared on two fundamentally same enormous informational indexes (e.g., bootstrap tests), the two classifiers will be indistinguishable. Little contrasts in the information will not prompt a lot of distinction in the assessments of the class implies. So the wastefulness of bagging for this case can be ascribed to the absence of assorted variety in the group. Direct classifiers may likewise become unsteady if the preparation size is small (Fig. 25.10).

At that point, any modification in the informational collection will majorly affect the outcome, subsequently making the classifiers not the same as one another. The threat here is that, in the event that we resort to little preparing test sizes (as we do in this examination), the general exactness of the individuals from the outfit will be low, thus will be the consolidated one. Hence the mix probably will not arrive at the exhibition of a solitary straight classifier prepared all in all preparation informational indexes.

4. Result analysis

4.1 kNN model

The total time for training the model is 0.001 s and the prediction time taken by the model is 0.003 s. The model thus gives an accuracy of 0.7, i.e., 70% (Fig. 25.11).

```
Training time: 0.001 s
Prediction time: 0.003 s

Report:

Accuracy: 0.7

              precision    recall  f1-score   support

           0       0.62      1.00      0.77         5
           1       1.00      0.40      0.57         5

    accuracy                           0.70        10
   macro avg       0.81      0.70      0.67        10
weighted avg       0.81      0.70      0.67        10

[[5 0]
 [3 2]]
```

FIGURE 25.11

Confusion matrix for kNN.

```
[[5 0]
 [2 0]]
              precision    recall  f1-score   support

           0       0.71      1.00      0.83         5
           1       0.00      0.00      0.00         2

    accuracy                           0.71         7
   macro avg       0.36      0.50      0.42         7
weighted avg       0.51      0.71      0.60         7

Training time: 0.032 s
Prediction time: 0.004 s
```

FIGURE 25.12

Confusion matrix for random forest.

4.2 Random forest

On training the datasheet in the random forest algorithm, the accuracy of the model increases to 85.71%, The confusion matrix for the random forest classifier is given next, which will provide us with a more clear idea (Fig. 25.12).

The total training time to train the model is 0.032 s and time taken to predict the results is 0.004 s. The calculation for the precision, recall, f1-score, and support has already been discussed (Fig. 25.13).

4.3 Bagging algorithm

On training the datasheet in the bagging algorithm, the accuracy of the model increases to 71.42%, which is quite good compared to the previously used kNN algorithm. The confusion matrix for the bagging algorithm classifier is given next, which will provide us with a more clear idea: the total training time to train the model is 0.05 s, and the time taken to predict the results is 0.003 s (Fig. 25.14). The calculation for the precision, recall, f1-score, and support has already been discussed (Table 25.3).

```
[[5 0]
 [2 0]]
              precision    recall  f1-score   support

           0       0.71      1.00      0.83         5
           1       0.00      0.00      0.00         2

    accuracy                           0.71         7
   macro avg       0.36      0.50      0.42         7
weighted avg       0.51      0.71      0.60         7

Training time: 0.05 s
Prediction time: 0.003 s
```

FIGURE 25.13

Confusion matrix for bagging algorithm.

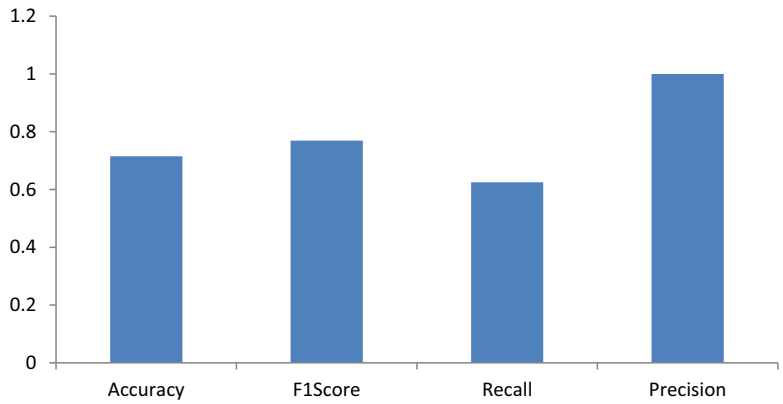

FIGURE 25.14

Evaluation metrics for kNN.

Table 25.3 Confusion matrix parameters.

	kNN	Random forest	Bagging algorithm
Accuracy (%)	70%	85.71%	71.42%
Sensitivity (%)	0.625	0.714	0.714
Specificity (%)	1	0	0
Precision (%)	1	1	1

5. Conclusion and future work

Classification models in ML algorithms were tested to make the best use of the clinical data provided online to be able to predict the severity of the COVID-19 cases. In this classification, we used five attributes and other relevant information of a patient to achieve the minimum classification error, which proves the feasibility of the proposed approach.

To conclude, we have made use of three basic algorithms, starting from kNN, random forest, and the bagging algorithms. The reason we chose these algorithms is in context to the target that is classifying the result as COVID-19 positive or negative, i.e., in binary form 0 or 1 (0 = negative, 1 = positive). A brief discussion of the obtained result is done after each prediction, highlighting all the accuracy obtained from individual models. The accuracy of the models could be increased by normalizing the datasheet more precisely. So finally, we have built three ML classification models, and we can see that the random forest classification algorithm gives the best results interns of accuracy and prediction time compared to kNN and random forest algorithm in this dataset. From the result, it has been seen that the patient who is more than 50 years old with a high fever after 5 days is confirmed with a case of COVID-19 (Fig. 25.15).

From Fig. 25.17, it is also seen that F1 score and recall both are better for bagging algorithm, while from Fig. 25.18, it is seen random forest algorithm has a maximum AUC score of 0.957, which is much higher than bagging (0.9366) and kNN (0.561). Hence, random forest shows higher predictive accuracy for the test datasets (Fig. 25.16).

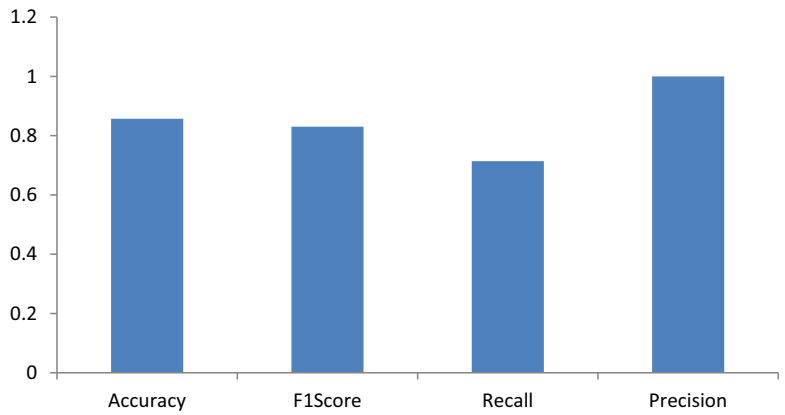

FIGURE 25.15

Evaluation metrics for random forest.

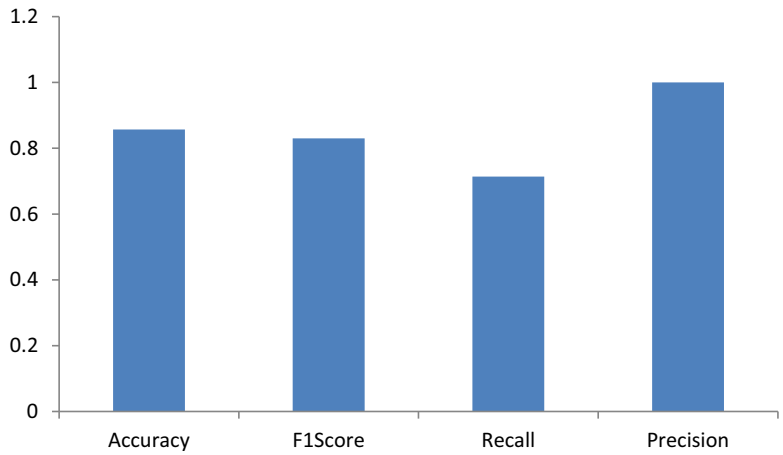

FIGURE 25.16

Evaluation metrics for bagging algorithm.

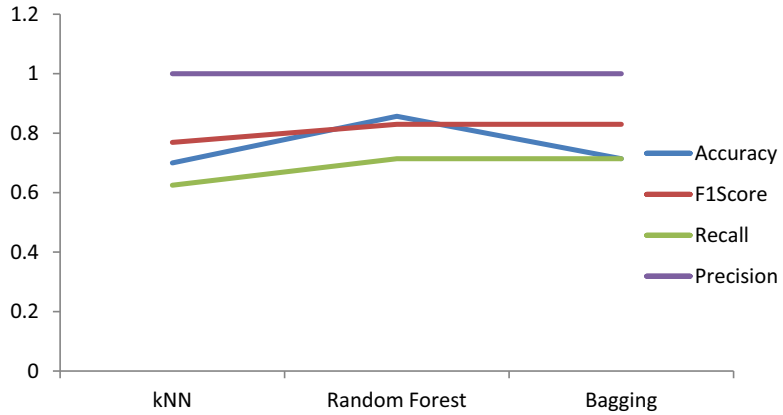

FIGURE 25.17

Comparison of models performance.

For future work, if data sets can be gathered by researchers or WHO organization or based on personal efforts to include symptoms and other information of suspects of COVID-19, we can be able to diagnose that new Coronavirus. Secondly, this classification can be further improved if other symptoms like loss of taste and smell, a rash on the skin, etc., are considered.

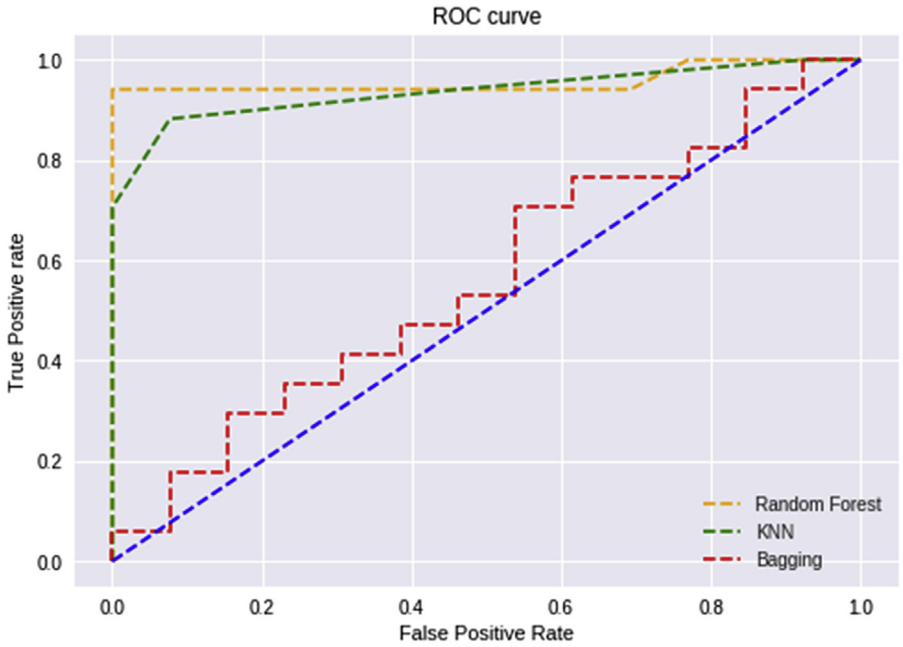

FIGURE 25.18

ROC for analysis for kNN, random forest, and bagging.

References

[1] W.C. Roda, et al., Why is it difficult to accurately predict the COVID-19 epidemic? Infect. Dis. Modell. 5 (2020) 271–281, https://doi.org/10.1016/j.idm.2020.03.001.

[2] M.M. Sajadi, et al., Temperature, humidity and latitude analysis to predict potential spread and seasonality for COVID-19, JAMA Net. Open (2020) 3550308, https://doi.org/10.2139/ssrn.3550308. SSRN.

[3] F.A.B. Hamzah, et al., CoronaTracker: World-wide COVID-19 outbreak data analysis and prediction, preprint. nCoV. (2020), https://doi.org/10.2471/BLT.20.255695.

[4] F. Petropoulos, S. Makridakis, Forecasting the novel coronavirus COVID-19, PloS One 15 (3) (2020) e0231236, https://doi.org/10.1371/journal.pone.0231236. Edited by L. A. Braunstein.

[5] D. Fanelli, F. Piazza, Analysis and forecast of COVID-19 spreading in China, Italy and France, Chaos, Solit. Fractals 134 (2020) 109761, https://doi.org/10.1016/j.chaos.2020.109761.

[6] X. Mei, et al., Artificial intelligence–enabled rapid diagnosis of patients with COVID-19, Nat. Med. 26 (8) (2020) 1224–1228, https://doi.org/10.1038/s41591-020-0931-3.

[7] S. rekha Hanumanthu, Role of intelligent computing in COVID-19 prognosis: a state-of-the-art review, Chaos, Solit. Fractals (2020), https://doi.org/10.1016/j.chaos.2020.109947.

[8] X. Chen, Z. Liu, Early prediction of mortality risk among severe COVID-19 patients using machine learning, preprint, Epidemiology (2020), https://doi.org/10.1101/2020.04.13.20064329.

[9] A. Mosavi, COVID-19 Outbreak Prediction with Machine Learning, 2020, p. 38.

[10] T. Fawcett, An introduction to ROC analysis, Pattern Recogn. Lett. 27 (8) (2006) 861−874, https://doi.org/10.1016/j.patrec.2005.10.010.
[11] S. Uddin, et al., Comparing different supervised machine learning algorithms for disease prediction, BMC Med. Inf. Decis. Making 19 (1) (2019) 281, https://doi.org/10.1186/s12911-019-1004-8.
[12] A.M.U.D. Khanday, et al., Machine learning based approaches for detecting COVID-19 using clinical text data, Int. J. Inf. Technol. (2020), https://doi.org/10.1007/s41870-020-00495-9.
[13] S.H. Kassani, et al., Automatic Detection of Coronavirus Disease (COVID-19) in X-Ray and CT Images: A Machine Learning-Based Approach, 2020 arXiv:2004.10641 [cs, eess]. Available at: http://arxiv.org/abs/2004.10641. (Accessed 19 August 2020).
[14] C. Iwendi, et al., COVID-19 patient health prediction using boosted random forest algorithm, Front. Public Health 8 (2020), https://doi.org/10.3389/fpubh.2020.00357.
[15] L.J. Muhammad, et al., Predictive data mining models for novel coronavirus (COVID-19) infected patients' recovery, SN Comp. Sci. 1 (4) (2020) 206, https://doi.org/10.1007/s42979-020-00216-w.
[16] G.F. Stark, et al., Predicting breast cancer risk using personal health data and machine learning models, PloS One 14 (12) (2019) e0226765, https://doi.org/10.1371/journal.pone.0226765.
[17] M. Chen, et al., Disease prediction by machine learning over big data from healthcare communities, IEEE Access 5 (2017) 8869−8879, https://doi.org/10.1109/ACCESS.2017.2694446.
[18] D. Sisodia, D.S. Sisodia, Prediction of diabetes using classification algorithms 132 (2018) 1578−1585.
[19] M. Juhola, et al., Detection of genetic cardiac diseases by Ca2+ transient profiles using machine learning methods, Sci. Rep. 8 (1) (2018) 9355, https://doi.org/10.1038/s41598-018-27695-5.
[20] B. Jin, et al., Predicting the risk of heart failure with EHR sequential data modeling, IEEE Access 6 (2018) 9256−9261, https://doi.org/10.1109/ACCESS.2017.2789324.
[21] H. Forssen, et al., Evaluation of machine learning methods to predict coronary artery disease using metabolomic data, Stud. Health Technol. Inf. 235 (2017) 111−115.
[22] D. Haro Alonso, et al., Prediction of cardiac death after adenosine myocardial perfusion SPECT based on machine learning, J. Nucl. Cardiol.: Offic. Publicat. Am. Soc. Nuc. Cardiol. 26 (5) (2019) 1746−1754, https://doi.org/10.1007/s12350-018-1250-7.
[23] A. Mustaqeem, et al., Wrapper method for feature selection to classify cardiac arrhythmia, 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), in: 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, Seogwipo, 2017, pp. 3656−3659, https://doi.org/10.1109/EMBC.2017.8037650.
[24] H. Mansoor, et al., Risk prediction model for in-hospital mortality in women with ST-elevation myocardial infarction: a machine learning approach, Heart Lung: J. Crit. Care 46 (6) (2017) 405−411, https://doi.org/10.1016/j.hrtlng.2017.09.003.
[25] T. Marikani, K. Shyamala, Prediction of heart disease using supervised learning algorithms, Int. J. Comput. Appl. 165 (5) (2017) 41−44.
[26] P. Lu, et al., Research on improved depth belief network-based prediction of cardiovascular diseases, J. Healthcare Eng. 2018 (2018) 1−9, https://doi.org/10.1155/2018/8954878.
[27] N. Khateeb, M. Usman, Efficient heart disease prediction system using K-nearest neighbor classification technique, in: Proceedings of the International Conference on Big Data and Internet of Thing, Association for Computing Machinery (BDIOT2017), New York, NY, USA, 2017, pp. 21−26, https://doi.org/10.1145/3175684.3175703.
[28] M.M. Islam, et al., Applications of machine learning in fatty live disease prediction, Stud. Health Technol. Inf. 247 (2018) 166−170.
[29] C.M. Lynch, et al., Prediction of lung cancer patient survival via supervised machine learning classification techniques, Int. J. Med. Inf. 108 (2017) 1−8, https://doi.org/10.1016/j.ijmedinf.2017.09.013.

[30] L. Hussain, et al., Prostate cancer detection using machine learning techniques by employing combination of features extracting strategies, Canc. Biomark.: Section A of Dis. Marker. 21 (2) (2018) 393−413, https://doi.org/10.3233/CBM-170643.
[31] B. Farran, et al., Predictive models to assess risk of type 2 diabetes, hypertension and comorbidity: machine-learning algorithms and validation using national health data from Kuwait–a cohort study, BMJ open 3 (5) (2013), https://doi.org/10.1136/bmjopen-2012-002457.
[32] R. Agrawal, K-nearest neighbor for uncertain data, Int. J. Comput. Appl. 105 (11) (2014) 13−16.
[33] C. Song, Introduction to K-Nearest Neighbors with Red Wines Quality in R, Medium (2018). Available at: https://medium.com/nyu-a3sr-data-science-team/k-nn-with-red-wines-quality-in-r-bd55dcba4fd7. Accessed: 19 August 2020.
[34] I. Okfalisa, M. Gazalba, N.G.I. REZA, "Comparative analysis of k-nearest neighbor and modified k-nearest neighbor algorithm for data classification, in: Information Systems and Electrical Engineering (ICITISEE), 2nd International conferences on Information Technology, Yogyakarta, 2017, pp. 294−298.
[35] X. Wu, et al., Top 10 algorithms in data mining, Knowl. Inf. Syst. 14 (1) (2008) 1−37, https://doi.org/10.1007/s10115-007-0114-2.
[36] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5−32, https://doi.org/10.1023/A:1010933404324.
[37] A. Lineberry, Exploring Random Forest Internal Knowledge and Converting the Model to Table Form, 2018. SpotX, 20 September. Available at: https://www.spotx.tv/resources/blog/developer-blog/exploring-random-forest-internal-knowledge-and-converting-the-model-to-table-form/. (Accessed 19 August 2020).
[38] T.K. Ho, Random decision forests, in: Proceedings of the Third International Conference on Document Analysis and Recognition, vol. 1, IEEE Computer Society (ICDAR '95), USA, 1995, p. 278.
[39] P. Bühlmann, Bagging, boosting and ensemble methods, in: J.E. Gentle, W.K. Härdle, Y. Mori (Eds.), Handbook of Computational Statistics, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 985−1022, https://doi.org/10.1007/978-3-642-21551-3_33.
[40] T. Khoshgoftaar, J.V. Hulse, A. Napolitano, Comparing boosting and bagging techniques with noisy and imbalanced data, IEEE Trans. Syst. Man Cyber. - Part A Syst. Human. 41 (3) (2011) 552−568.
[41] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123−140, https://doi.org/10.1007/BF00058655.
[42] L. Breiman, Arcing classifiers, Ann. Stat. 26 (3) (1998) 801−849.
[43] A.J.C. Sharkey, Combining predictors, in: A.J.C. Sharkey (Ed.), Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems, Springer, London, 1999, pp. 31−50, https://doi.org/10.1007/978-1-4471-0793-4_2 (Perspectives in Neural Computing).
[44] B. Efron, R. Tibshirani, An Introduction to the Bootstrap, Chapman and Hall (Monographs on statistics and applied probability, New York, 1993, p. 57.
[45] T.G. Dietterich, Ensemble methods in machine learning, in: Multiple Classifier Systems, Springer Berlin Heidelberg (Lecture Notes in Computer Science), Berlin, Heidelberg, 2000, pp. 1−15, https://doi.org/10.1007/3-540-45014-9_1.
[46] M. Skurichina, Stabilizing weak classifiers: regularization and combining techniques in discriminant analysis, 2001.