COVID

MDPI

Article

Utilizing Machine Learning Techniques for Computer-Aided COVID-19 Screening Based on Clinical Data

Honglun Xu [1], Andrews T. Anum [2], Michael Pokojovy [3], Sreenath Chalil Madathil [4], Yuxin Wen [5], Md Fashiar Rahman [1], Tzu-Liang (Bill) Tseng [1,*], Scott Moen [6] and Eric Walser [7]

[1] Department of Industrial, Manufacturing and Systems Engineering, The University of Texas at El Paso, El Paso, TX 79968, USA; hxu4@utep.edu (H.X.); mrahman13@utep.edu (M.F.R.)
[2] Department of Mathematical Sciences, The University of Memphis, Memphis, TN 38152, USA; atanum@memphis.edu
[3] Department of Mathematics & Statistics, School of Data Science, Old Dominion University, Norfolk, VA 23529, USA; mpokojovy@odu.edu
[4] Department of Systems Science and Industrial Engineering, Binghamton University, Binghamton, NY 13902, USA; schalil@binghamton.edu
[5] Dale E. and Sarah Ann Fowler School of Engineering, Chapman University, Orange, CA 92866, USA; yuwen@chapman.edu
[6] Department of Microbiology & Immunology, The University of Texas Medical Branch, Galveston, TX 77550, USA
[7] Department of Radiology Prostate, The University of Texas Medical Branch, Galveston, TX 77550, USA; emwalser@utmb.edu
* Correspondence: btseng@utep.edu

Abstract

The COVID-19 pandemic has highlighted the importance of rapid clinical decision-making to facilitate the efficient usage of healthcare resources. Over the past decade, machine learning (ML) has caused a tectonic shift in healthcare, empowering data-driven prediction and decision-making. Recent research demonstrates how ML was used to respond to the COVID-19 pandemic. This paper puts forth new computer-aided COVID-19 disease screening techniques using six classes of ML algorithms (including penalized logistic regression, random forest, artificial neural networks, and support vector machines) and evaluates their performance when applied to a real-world clinical dataset containing patients' demographic information and vital indices (such as sex, ethnicity, age, pulse, pulse oximetry, respirations, temperature, BP systolic, BP diastolic, and BMI), as well as ICD-10 codes of existing comorbidities, as attributes to predict the risk of having COVID-19 for given patient(s). Variable importance metrics computed using a random forest model were used to reduce the number of important predictors to thirteen. Using prediction accuracy, sensitivity, specificity, and AUC as performance metrics, the performance of various ML methods was assessed, and the best model was selected. Our proposed model can be used in clinical settings as a rapid and accessible COVID-19 screening technique.

Keywords: machine learning; COVID-19 screening; patient data; dimensionality reduction

Check for updates

Academic Editor: Somenath Chakraborty

Received: 29 November 2025
Revised: 6 January 2026
Accepted: 7 January 2026
Published: 9 January 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

The COVID-19 pandemic has exacerbated the urgent need for rapid, scalable, and low-cost screening tools that can support early identification of infected individuals and reduce transmission risk, particularly in resource-constrained and high-throughput settings such as emergency departments, outpatient clinics, and community screening sites [1,2]. Although molecular testing methods, including RT-PCR, remain the diagnostic gold standard,

their limited availability, processing delays, logistical complexity, and reliance on specialized laboratory infrastructure restrict their effectiveness as large-scale screening solutions. These limitations are especially pronounced during infection surges, when testing demand exceeds capacity and timely isolation decisions are critical.

In response, there has been growing interest in machine learning-based screening approaches that leverage routinely collected clinical, physiological, and demographic data to provide rapid, pre-diagnostic risk stratification prior to confirmatory testing [3–5]. Such approaches have the potential to complement laboratory diagnostics by enabling early triage, prioritizing testing resources, and supporting real-time decision-making without additional clinical burden. However, the practical utility of these models depends not only on predictive accuracy, but also on their robustness, interpretability, and performance under the class imbalance typical of screening scenarios. Consequently, developing and systematically evaluating multiple machine learning models using consistent datasets and evaluation criteria is essential for understanding their relative strengths, limitations, and trade-offs between sensitivity, specificity, and operational feasibility in real-world screening workflows [6,7].

Advanced AI techniques can be used to identify COVID-19 patients and those who are less likely to be exposed to the virus [8]. Machine learning is a subfield of artificial intelligence that focuses on data-driven model learning and prediction. Recent developments in AI-driven diagnosis and screening of diseases have pushed the frontiers of AI to include what was once an exclusive human prerogative [9]. Healthcare systems and medicine are among the most promising application fields of AI, probably dating back to as early as mid-twentieth century [10,11]. Researchers successfully proposed and developed a number of decision systems, such as rule-based systems and AI diagnosis systems [12]. Rule-based systems proved successful in the late seventies and were beneficial for detecting disease [13,14], selecting proper treatment, and generating diagnosis hypotheses for medical doctors [15]. Unlike early generations of knowledge-based AI diagnosis systems, which depend on prior medical knowledge of experts and construction based-rules, modern AI applies ML algorithms to search for associations and patterns in data [16–18]. A modern deep learning artificial network typically involves a large number of hidden layers [19]. Current developments in AI have resulted in the natural question as to whether AI physicians can potentially replace human doctors in the foreseeable future. Leaving this question undecided, many scientists believe that AI-driven intelligent systems can meaningfully assist and guide human doctors in making quick and better decisions, and even occasionally eliminate the requisite of human decisions in such fields as radiology. The ever-increasing amounts of data in healthcare, resulting from the growing use of digital technology and the rapid development of Big Data analytics methods, have been attributed to the present success of AI in healthcare [20]. Even though the use of AI in healthcare is still in a developmental phase, most research is focused on diseases in three categories: cardiology, neurology, and oncology [21]. Guided by discovery and evidence, robust AI can leverage medical data and be applied for prediction and decision-making support [22].

With AI having proved valuable in healthcare, one naturally expects it to be potentially useful when applied in COVID-19 screening diagnosis. From designing anti-viral-replication molecules to forecasting the dynamics of a pandemic, AI has already made a significant impact on healthcare [23]. Recent research on COVID-19 suggests that AI can also be used in detecting COVID-19 infection in patients [24]. Fast detection of COVID-19 is vital, because a false negative result can not only postpone treatment but increase the risk of viral transmission to others. AI has the potential to facilitate and improve earlier detection of SARS-CoV-2 infection. The goal of this paper is to analyze and compare several leading machine learning techniques when applied in COVID-19 screening in order to discriminate

between COVID-19 patients and non-COVID-19 patients. The six types of ML models studied in this paper are logistic regression, classification and regression trees (CARTs), bootstrap aggregation (bagging), random forests, support vector machines (SVMs) and artificial neural networks (ANNs). The clinical dataset we used for this investigation was provided by The University of Texas Medical Branch (UTMB) and includes non-image clinical data collected from COVID and non-COVID-19 patients. A full description of the data is given in the next section. With each patient having a unique identifier, the dataset is spread across several files. As a result, we performed extensive data preprocessing to compile a single data frame for this study. The resulting data frame contained, however, multiple entries (across variables and observations) with missing information. We used the well-known MICE technique [25] to impute missing information. Further, we performed variable/feature selection based on variable importance measures produced with non-parametric random forests to reduce the number of variables. Splitting the dataset into training and test sets (with an 80:20 ratio), the six ML models were applied to the original dataset with all features, as well as the "reduced" dataset with all irrelevant features removed. Various performance metrics, including the area under the receiver operating curve (AUC), were recorded and compared.

2. Data Description and Preprocessing

The University of Texas Medical Branch (UTMB) clinical COVID-19 dataset contains demographic data, vital measurements, and multiple ICD-10 codes of respective comorbidities for each of the 1987 patients [26]. With each patient having a unique anonymized identifier, the raw data are spread over multiple files. For each unique patient ID, we identified the latest PCR (Polymerase Chain Reaction) date and the PCR test result (COVID-19 detected or not) corresponding to this date. Demographics (sex, age, ethnicity) corresponding to this unique ID were added. Further, the five-number summary for each of the "vitals" (pulse, pulse oximetry, respirations, temperature, systolic blood pressure, diastolic blood pressure, and BMI) recorded within a reasonable temporary proximity to the PCR test date (two weeks before and one day after) were appended. Only the 20 most relevant out of a total of 6720 ICD-10 codes (in terms of their frequency in the training dataset) were retained. This resulted in a data frame with 78 predictors (58 demographic/vital and 20 ICD-10 binary variables) and one response variable (COVID-19 status based on the PCR test) recorded in 1987 patients. With each patient labeled as either COVID-19-positive (detected) or -negative (not detected), the data were partitioned into training and test sets at an 80:20 ratio, yielding 1590 training cases and 397 test cases. The test set contained 151 positive and 246 negative cases, and the training set contained 684 positive and 906 negative cases. Both training and test data appeared to be fairly balanced but contained some missing entries. Therefore, we performed imputation using the Multivariate Imputation by Chained Equations (MICE) method [25,27] available from the `mice` package in `R`, based on random forest (RF) as the imputation method. Missing data imputation using MICE was performed separately on the training and test sets, resulting in the full training and test datasets with 78 predictors. Feature selection performed exclusively on the training dataset identified a reduced subset of 13 predictors, yielding the "reduced" training dataset. These predictors were subsequently used to construct the corresponding reduced test dataset. Table 1 provides a brief description of the resulting data frame.

Table 1. Description of non-image COVID-19 data variables.

Variables	Description	Type
sex	Gender (male or female)	Binary
ethnicity	Whether a patient is Hispanic or Latino	Binary
age	Patient's age	Numerical
pulse	Number of pulse beats per minute	Numerical
pulse oximetry	Blood oxygen level (oxygen saturation)	Numerical
respirations	Number of breaths per minute	Numerical
temperature	Body temperature	Numerical
BP systolic	Systolic blood pressure (top number): the force heart exerts on the walls of arteries each time it beats	Numerical
BP diastolic	Diastolic blood pressure (bottom number): the force heart exerts on the walls of arteries in between beats	Numerical
BMI	Measure of under-/overweight	Numerical
ICD-10 codes	Binary indicator (yes/no) for each ICD *-10 code (for diseases, signs and symptoms, abnormal findings, complaints, social circumstances, and external causes of injury or diseases)	Binary

* International Statistical Classification of Diseases and Related Health Problems.

3. Methodology

Numerous ML methods are known in the literature that had the potential to be used to recognize patterns in the UTMB COVID-19 clinical dataset and classify patients into COVID-19-positive or -negative. In this study, we chose six particular methods, namely, logistic regression, classification and regression trees, bootstrap aggregation (bagging), random forests, support vector machines, and artificial neural networks. These machine learning techniques were chosen due to their versatility, flexibility, and overwhelming popularity. In addition, these six methods are well-understood and widely applied in classification. While logistic regression and SVM with linear and polynomial kernels are parametric, the rest of the methods considered are non-parametric.

3.1. Logistic Regression

The logistic regression model is a parametric statistical approach to model the relationship between multiple predictor variables or attributes/features and a binary response variable. The logistic regression analysis method is closely related to the usual linear regression method. While linear regression estimates the parameters using the ordinary least squares (OLS) method, the logistic regression model employs the nonlinear maximum likelihood estimation (MLE) method to estimate the coefficients [28].

Logistic regression is typically used when the response variable is binary, like 'yes' and 'no', 0 and 1, or true and false, and, therefore, is sometimes called binary logistic regression. Using the usual logit-transform, the logistic model reads as

$$\text{logit}(\pi) = \log(\pi/(1-\pi)) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p + \varepsilon \quad (1)$$

where the error ε is modeled as a Gaussian random variable independent of x_i's. With y denoting the binary response variable, π denotes the probability of $y = 1$ and $1 - \pi$ is the probability of $y = 0$. The ratio $\pi/(1-\pi)$ is referred to as the odds.

3.2. Classification and Regression Trees

Classification and regression trees (CARTs) are a non-parametric statistical method that uses a decision tree to model the response of a continuous or categorical output variable. This method was developed by [29] to create a decision tree for a single response variable, and is thus referred to as univariate CART.

If the dependent variable is categorical, CART is formulated as a classification tree, whereas a continuous response yields a regression tree. In this study, a CART classifier was constructed using a binary recursive partitioning procedure applied to the training dataset. Tree growth began at the root node containing all training observations, with successive binary splits selected by evaluating all predictors and candidate split points to maximize node purity based on the Gini index or entropy [30]. This recursive splitting process continued until a maximal, fully grown tree was obtained, which typically captures complex interactions among predictors but is prone to overfitting. To address this, cost-complexity pruning was applied to balance predictive accuracy and model interpretability by generating a sequence of nested subtrees indexed by a complexity parameter α, which penalizes excessive terminal nodes [31]. The optimal subtree was selected using 10-fold cross-validation on the training data by minimizing the penalized risk $R_\alpha(\check{T})$:

$$R_\alpha(\check{T}) = R(\check{T}) + \alpha|\check{T}| \quad (2)$$

where $R(\check{T})$ is the Gini index or entropy, $|\check{T}|$ is the complexity of the tree, which is the total number of terminal nodes of the subtree, and α is the complexity parameter regulating the penalty for every extra terminal node. The resulting pruned tree was then used for performance evaluation on the test dataset.

3.3. Bootstrap Aggregation (Bagging)

The bagging algorithm, also known as bootstrap aggregation [32], is a voting algorithm involving an ensemble learning method that is commonly used to reduce variance within a noisy dataset [33]. This method classifies an observation based on different training datasets which come from the same distribution as the original training dataset. This method of classification yields a better classification compared to when only the original dataset is used, and also reduces the variance of the resulting prediction [34]. Normally, we are unable to create multiple training datasets that imitate the original dataset. Therefore, bootstrap samples are generated from the training set by resampling with replacement. These bootstrap samples generated from the training dataset come from the bootstrap distribution, which approximates the distribution of the training dataset [32]. The bootstrap samples drawn have the same number of data points as the original training dataset at each sample. Some observations may not be drawn at each sample or be drawn multiple times as a result of sampling with replacement. By majority voting on classifiers obtained from these different bootstrap samples, the class assigned to an observation is the class with most votes. The method also takes into consideration cases where there are ties in voting. In such cases, any one class of the classes involved is randomly selected. The classifiers obtained are then put together to form a classifier that is used to classify the test set. Selecting CART as basic classifier, the algorithm reads as follows:

1. Draw a bootstrap sample from the training set;
2. Apply CART to the bootstrap sample;
3. Repeat these steps for a preselected number of times (say, 500);
4. Based on majority voting, combine basic CART classifiers to produce the final decision criterion.

3.4. Random Forests

Random forest is an ensemble classifier that aggregates predictions from multiple decision trees using majority voting [35]. The random forest model can be thought of as a modification of the bagging model. Random forest deals with recursively constructing binary trees. With the random forest model, the average of a large collection of trees is considered such that the correlation among individual trees is not "too strong" [34]. Each

of these trees is grown using a bootstrap sample from the dataset. At each node of the tree, a random forest randomly selects m variables from the original p variables. These variables serve as candidates for obtaining the best split for the node. The random forest model can be used for regression as well as for classification. For classification tasks, the final class label is determined by majority voting across all trees. Since we aim to classify the UTMB's COVID-19 dataset into two classes, we will use a random forest model as a classification tool. The algorithm is as follows:

1. Grow a decision tree for different bootstrap samples of size p' from the training data;
2. When growing the tree, select m variables at random from the p variables at each step;
3. From the m variables randomly selected, choose the best split variable;
4. Split the node into two nodes until the minimum node size is reached;
5. Output the ensemble of the trees.

3.5. Support Vector Machines

Support vector machines (SVMs) were employed as supervised classifiers for COVID-19 screening using the training dataset [36–38]. To accommodate potential linear and nonlinear relationships among predictors, SVM models with linear, radial basis function (RBF), and polynomial kernels were evaluated to capture different linear and nonlinear decision boundaries. Model training focused on identifying an optimal separating hyperplane by maximizing the margin while allowing for controlled misclassification through a cost parameter [39–41]. Hyperparameters, including the cost parameter and kernel-specific parameters (e.g., bandwidth for the RBF kernel and degree and scale for the polynomial kernel), were tuned using grid search combined with bootstrap or cross-validation on the training data. The optimal configurations were selected based on cross-validated performance, and the finalized SVM models were subsequently evaluated on the held-out test set using metrics including accuracy, sensitivity, specificity, and AUC.

3.6. Artificial Neural Networks

The artificial neural network (ANN) is a data processing system which consists of a series of interconnected processing elements akin to a biological neural system. It has the ability to learn from an experimental or real dataset to determine nonlinear and interaction effects. ANN models are among the most widely applied artificial intelligence methods that are used for prediction and forecasting purposes. An ANN consists of three types of layers: the input layer, one or multiple hidden layers, and the output layer. The number of units in the input and output layers depends on the data size. Input units pass the raw data to the network without performing any computation. Then, these units will pass information into the hidden units. The hidden nodes process data and transfer information from the input units to the output units [42,43].

The ANN model is then trained by adjusting the weights as to reduce the prediction error. To compute errors in these ANN systems, generated ANN outputs need to be compared to ground truth targets. Stochastic gradient descent can be used to update the weights and compute errors at each iteration until the optimal value is found.

3.7. Summary

Based on the six methods described, a schematic illustration of our COVID-19 non-image-based diagnosis procedure is displayed in Figure 1. First, the data are preprocessed using MICE [25] to impute missing records. Then, the full model and reduced models are created separately for each of the methods. Full models use all features from the dataset, while reduced models only use variables deemed to be main influence factors based on variable importance measures computed using a random forest approach to reduce the

number of features from the original dataset. Finally, these two models are deployed separately, and simulation results are compared based on performance metrics including accuracy and AUC.

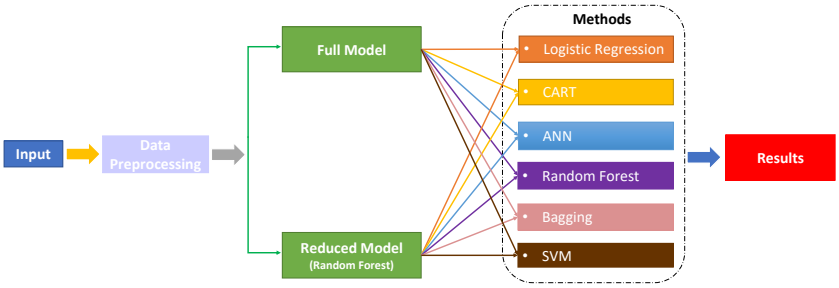


Figure 1. Illustration of COVID-19 non-image-based diagnosis procedure.

4. Empirical Results and Discussion

We randomly split the data into training (80% for fitting a predictive model) and testing (20% for evaluating the model) datasets. Based on the training dataset, the six ML models are trained and used to predict whether a patient in the test data is diagnosed with COVID-19 (positive vs. not detected based on PCR test). We report the classification accuracy, sensitivity, specificity, F1 score, and area under the receiver operating characteristic (ROC) curve for all the methods considered. The numerical simulation [44] is performed using the `R` statistical (version 4.5.2) software.

4.1. Full Dataset/Model

The full model involves all features, namely, sex, ethnicity, age, pulse, pulse oximetry, respirations, temperature, BP systolic, BP diastolic and BMI, PCR, and top 20 ICD-10 codes. All methods described in Section 3 are applied to the data.

First, we apply the logistic regression technique to build a predictive model using the training data. The training data size is 1590 and has 78 predictors. The binary responses are COVID-19 positive (detected) and COVID-19 negative (not detected). In this study, LASSO regularization is applied [45]. The penalty can be used for variable selection and shrinkage because it has the effect of forcing some of the coefficient estimates to be zero. Table 2 shows important predictors using the best predictive model with ℓ_1 penalty. For the ℓ_1-penalized logistic regression model, predictors retained after regularization are interpreted as important features rather than statistically significant variables. Coefficient magnitude reflects the relative contribution of predictors within the penalized model, which does not imply statistical significance because penalized models prioritize feature selection and shrinkage.We then apply the trained model to the test data using this predictive model and compute the performance metrics, including accuracy. A plot of the ROC curve for the logistic regression model is given in Figure 2. The AUC is 0.7621 and accuracy is 72.5% for test data.

The CART methodology generates trees with several levels and may result in overfitting. So, tree pruning is used to improve generalization of the outcomes [46]. The pruning step also identifies influential predictors retained in the final tree for subsequent analysis. Node splitting is performed using the Gini impurity criterion, which is computationally efficient and promotes class homogeneity [47]. Model selection is carried out using 10-fold cross-validation, resulting in an optimal tree with 10 terminal nodes. The final tree has a sensitivity of 58.6% and specificity of 73.2%, resulting in an AUC of 0.6821 for the test data. Figure 2 displays the ROC curve for the CART model.

Table 2. Coefficients of important predictors using logistic regression model for COVID-19 data.

Variable	Coefficients
sex	4.884×10^{-2}
ethnicity	1.300×10^{-12}
I10 code	1.230×10^{-3}
pulse	7.300×10^{-4}
pulse oximetry	5.110×10^{-3}
R52 code	3.119×10^{-2}
temperature	6.208×10^{-2}
E11.9 code	4.212×10^{-2}
BP diastolic	4.310×10^{-3}
BMI	1.868×10^{-2}
I50.9 code	2.643×10^{-2}

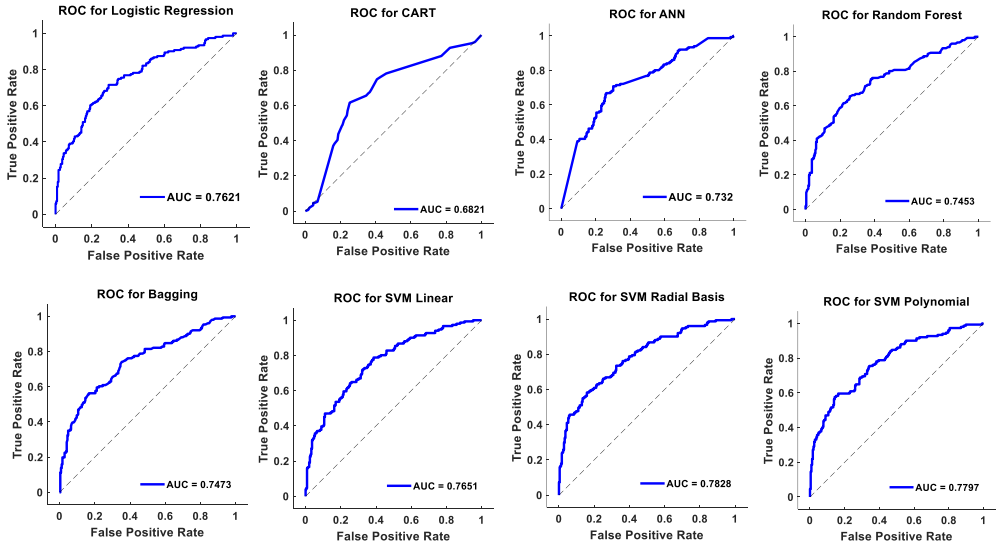


Figure 2. Test ROC curves for the full model.

Several random forest models were trained in `R` using the training dataset with various `ntree` values (number of trees to grow) uniformly chosen between 100 and 2500 and `mtry` (number of variables randomly sampled as candidates at each split) chosen to be $\sqrt{p}$. Each of these models were then used to classify the test data and the classification accuracies were recorded. The best model was selected based on highest classification accuracy. Figure 2 displays the respective ROC curve for test data. The test AUC for the random forest was 0.7453.

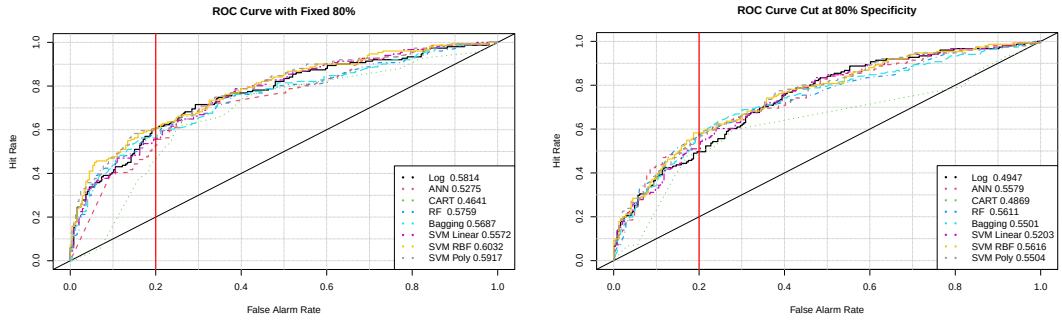
Also, several bagging models were trained using the training dataset, with different values of `ntree` uniformly chosen between 99 and 501 and a `mtry` value of $(p-1)$. The model with the best test classification accuracy was selected.

To obtain the best parameters for the support vector machine model, hyperparameters were tuned using 10-fold cross-validation on the training set. For the linear SVM, the cost parameter was selected from the grid (0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2).

From cross-validation, the best `cost` value obtained was 0.01. Figure 2 shows the ROC curve for the SVM with linear kernel for test data.

The `cost` and `sigma` hyperparameters were tuned for the SVM with radial basis kernel on the grids (0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2) and (0.001, 0.01, 0.1, 1), respectively. The best model in the parameter range is obtained using `cost` $= 0.5$ and `sigma` $= 0.01$. The model is used to classify the test data.

The best model for SVM with polynomial kernel is obtained using `degree` = 2, `cost` = 0.5 and `sigma` = 0.01. The parameters (`degree` of polynomial, `cost` and `scale`) are tuned on the grids (2.0, 3.0), (0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1.0, 1.5, 2.0), and (0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1.0), respectively. Figure 2 displays the test ROC curve for SVM with polynomial kernel. In our simulation studies, we assessed the classification performance of the machine learning techniques by the area under the receiver operating characteristic curve of the test dataset. The corresponding ROC plots obtained from these methods are shown in Figure 2. While ROC curves provide a threshold-independent measure of discrimination, they may not fully reflect performance in screening settings with class imbalance. Accordingly, sensitivity at a fixed specificity was used to represent a screening-oriented operating point that emphasizes minimizing false negatives. A specificity of 0.8 was chosen to enable consistent comparison across models rather than a prescriptive clinical cutoff, as optimal operating points may vary depending on institutional screening priorities and available resources. From Figure 3a, SVM radial basis has the largest sensitivity of 0.6032 at a fixed operating point.



(**a**) Full Model (**b**) Reduced Model

Figure 3. Test ROC curves compared at 80% specificity for the full and reduced models.

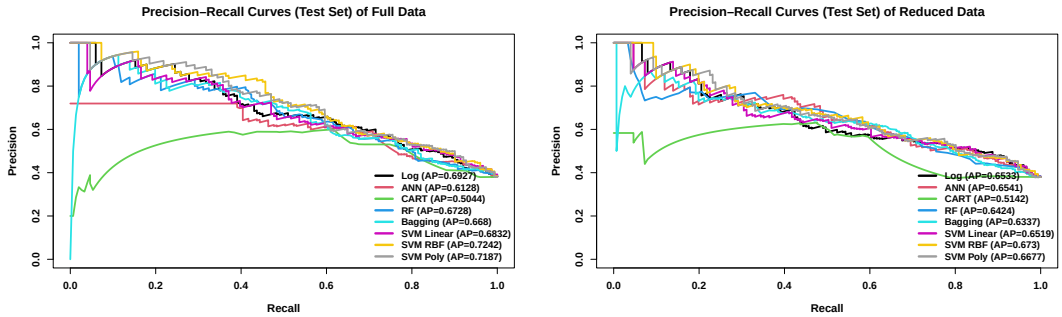
Next, we discuss artificial neural network models. We used the resilient backpropagation without weight backtracking to train the model with 2 hidden layers made by 5 and 3 neurons, respectively. After training the model, the accuracy on the test dataset was 70.8%, with a sensitivity of 74.8% and specificity of 64.2%, leading to an AUC of 0.7295.

Table 3 gives a summary of the resulting test AUC obtained by using all the methods mentioned above to classify the test dataset. From Table 3, SVM radial basis also produced better classification results based on the AUC of 0.7828 as compared to other methods. For the full-feature models, kernel-based SVMs demonstrate the strongest overall screening performance, as reflected by their comparatively higher F1 scores. The SVM with polynomial kernel achieves the highest F1 score (0.7049), followed closely by the SVM with radial basis kernel (0.7014) and logistic regression (0.6886), indicating a favorable balance between case detection and error control at the selected operating point. Linear SVM (0.6759) and ANN (0.6732) also show competitive performance, suggesting reasonable screening utility despite simpler or more constrained modeling assumptions. Among tree-based approaches, random forest (0.6585) and bagging (0.6443) provide moderate screening effectiveness, while CART exhibits the weakest performance (0.6038), reflecting limited robustness in balancing sensitivity and specificity. Collectively, these results highlight the advantage of kernel-based and margin-based classifiers for COVID-19 screening when evaluated using decision-threshold–dependent metrics that more closely reflect real-world deployment considerations.

Table 3. Performance results on the full test dataset.

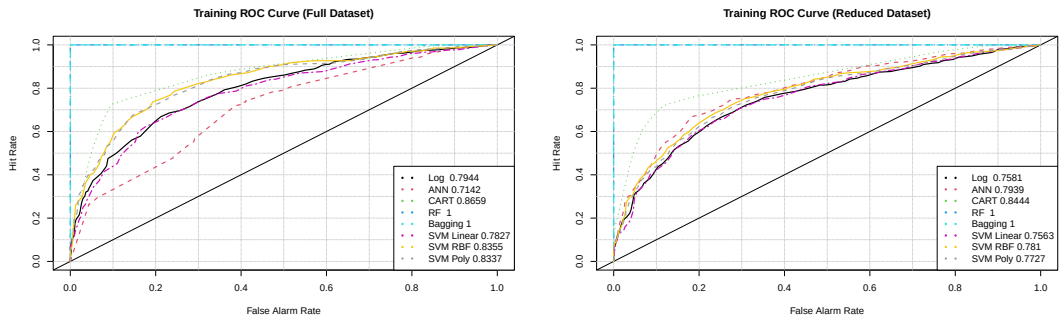
Method	Accuracy	Sensitivity	Specificity	AUC	F1 Score
random forest	0.718	0.656	0.746	0.7529	0.6585
bagging	0.710	0.632	0.751	0.7473	0.6443
SVM linear	0.715	0.837	0.517	0.7651	0.6759
SVM radial basis	0.743	0.862	0.550	0.7828	0.7014
SVM polynomial	0.746	0.850	0.576	0.7797	0.7049
ANN	0.708	0.748	0.642	0.7295	0.6732
CART	0.680	0.586	0.732	0.6821	0.6038
logistic regression	0.725	0.797	0.609	0.7621	0.6886

Given the class imbalance inherent in COVID-19 screening and the clinical importance of minimizing false negatives, we additionally evaluated all models using precision–recall curves and average precision, which better reflect positive-class performance across operating thresholds. Figure 4a shows the precision–recall (PR) curves for all models evaluated on the full test dataset. Kernel-based SVM and ensemble models achieve the strongest PR performance, with the SVM-RBF model yielding the highest average precision (AP) (0.7242), followed by SVM-Poly (0.7187) and Logistics Regression (0.6927). SVM-Linear (0.6832) and RF (0.6728) exhibit lower precision at higher recall levels, indicating increased false positives when sensitivity is prioritized. Overall, models with higher AP maintain superior precision across clinically relevant recall ranges, supporting their suitability for COVID-19 screening where false negatives must be minimized. Figure 5a illustrates the training ROC curves for all models using the full feature set, showing near-perfect AUC for ensemble methods and high but non-saturated performance for nonlinear SVM and ANN models, which provides a clear baseline for comparison with test-set results and assessment of potential overfitting.



(**a**) Full dataset (**b**) Reduced dataset

Figure 4. Precision-recall curves for all models on full and reduced test dataset.



(**a**) Full training dataset (**b**) Reduced training dataset

Figure 5. Training ROC curves for all models on full and reduced training dataset.

4.2. Reduced Dataset/Model

Using the full model, i.e., keeping all predictors, running simulations can be slow and the accuracy may decrease for some of the methods, because some predictors may not be related to the PCR test outcomes. Thus, feature selection may be desirable to improve computation speed and mitigate the problem of overfitting. Using fewer predictors reduces training time and computational cost and may improve model performance by achieving a better bias–variance trade-off. Feature selection and dimensionality reduction also help reduce noise and multicollinearity in the data. To this end, we performed variable selection before building and comparing predictive models.

4.2.1. Variable Selection

There are numerous variable selection techniques, which are based on computing the importance of variables or features. Ref. [48] proposed an SVM score criterion in terms of weight vectors or generalization error that bounds sensitivity of individual variables. Ref. [49] put forward a variable selection procedure by a stepwise strategy that includes successive applications of the CART method. Ref. [50] proposed the application of chi-squared scores and p-values in selecting variables to retain in logistic regression models. The best predictors are selected in the model with the highest chi-squared value [50]. In contrast, being non-parametric, random forests can successfully compute how much every variable reduces the node impurity providing a useful important measure. The most significant variable is the one that yields the largest reduction in mean impurity. The eventual importance of the variables is the average of the mean decrease for each variable across all trees.

For the reduced-feature models, variable selection is guided by random forest–based feature importance rankings derived from the full feature set. Predictors with higher importance scores are retained to construct a parsimonious reduced model, allowing evaluation of model robustness under feature constraints. This approach provides a data-driven mechanism for feature reduction while avoiding the introduction of post hoc explanation methods, and it facilitates comparison of model performance when only a limited subset of clinically available variables is used.

The main purpose of this subsection is to describe how random forests were applied for variable selection for our study. The predictor variables to be considered for inclusion in our models were ranked in the order of their importance. The variable importance index based on random forests, which is also known as mean decrease accuracy index (MDA), deliberates relationships between variables. This makes it a robust method for discovering important variables that will be used in predictive models. This involves ranking of explanatory (independent) variables using the random forests score of importance. The MDA (scaled by default)

$$\text{MDA} = \frac{\text{Mean(Decreases in Accuracy of Trees)}}{\text{Standard Deviation(Decreases in Accuracy of Trees)}} \tag{3}$$

mimics a test statistic and neither represents a percentage nor count of observations.

The Out-of-bag (OOB) misclassification rate serves as a guide to determine the ideal number of trees for aggregation in a random forest model. Initially, as depicted in Figure 6, the misclassification rate fluctuates significantly but eventually stabilizes, showing less pronounced changes. The primary focus is on the black OOB misclassification rate curve, where a minimum is noted at the 9th grid point, indicating one possible preferred choice. For parsimony reasons, we also presented the curves for plus/minus one standard error of the OOB (shown as red dashed lines). Due to the big standard error, our analysis is predominantly based on these red dashed curves. From the lowest point on the black

curve, we identify where it intersects with the curve for misclassification error minus 1× standard error. This intersection determines the optimal number of trees for the random forest, which is found to be 650.

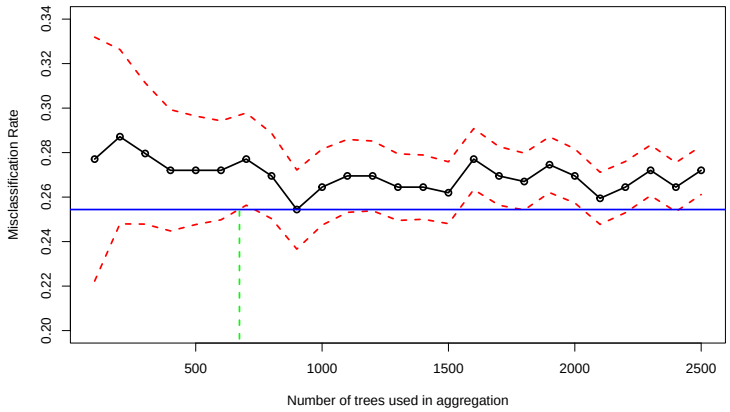


Figure 6. Out-of-bag (OOB) misclassification rate.

Based on the optimized random forest model, Figure 7 illustrates the top 20 variables by importance for classification. The Mean Decrease Accuracy plot expresses how much accuracy the model losses by excluding each variable. The more the accuracy suffers, the more important the variable is for the successful classification. The variables are presented in descending order of importance. Although the elbow method can be used to select the number of important variables, we instead adopted a cross-validation–based approach. Specifically, 10-fold cross-validation was used to determine the optimal number of variables by minimizing the mean cross-validation error (RMSE of cross-validation). The cross-validation curve (Figure 8) shows the relationship between model error and the number of predictors.

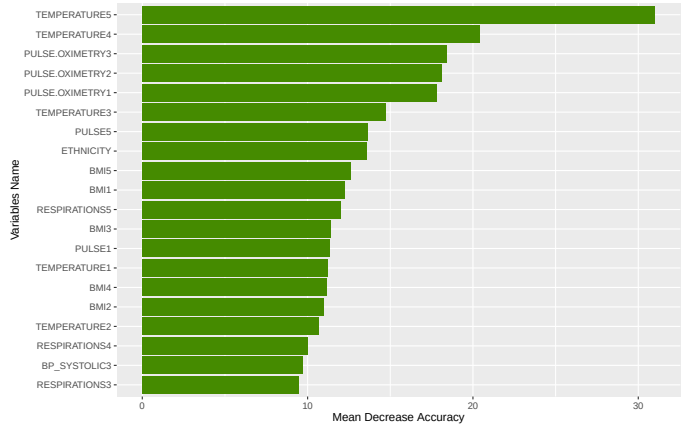


Figure 7. Variables Importance.

Figure 8 shows a plot of the CV errors for the number of predictors. Generally, the errors decrease with an increase in predictors and the minimum error was attained with 49 predictors. The plus/minus 1-standard error-curves (red dashed) are displayed in plot. For model parsimony considerations, the cross intersection point of the mean plus 1× standard error curve with the minimum CV error (from black curve) is selected as the optimal number of variables. From Figure 8, the intersection occurs at 13 variables. Thus, we choose the top 13 variables.

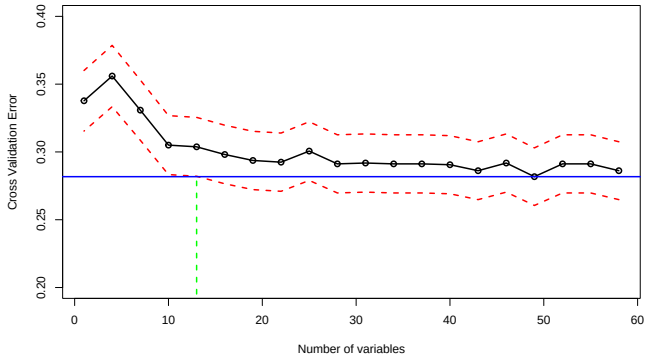


Figure 8. Variable selection with cross-validation.

4.2.2. Results for the Reduced Model

After reducing the number of variables, we used the top 13 predictors from Figure 8 to obtain a new "reduced" dataset. Then, the six ML methods were separately applied to this reduced dataset. Firstly, the logistic regression technique was used to build a model using the training set. We used LASSO as regularization technique.

Table 4 shows important variables using the logistic model with ℓ_1-penalty. Pulse, Pulse Oximetry and Ethnicity appear to be among the important predictors. Test data were used to assess the predictive model and obtain test ~~the~~ performance metrics such as accuracy. Figure 9 displays the ROC curve for the test dataset. The test AUC is 0.7416 and the test accuracy is 68.3% for logistic regression model.

Table 4. Estimated coefficients for selected variables using LASSO.

Variable	Coefficient
ethnicity	5.180×10^{-14}
pulse oximetry	8.180×10^{-9}
pulse	4.480×10^{-6}
temperature	8.910×10^{-3}
BMI	3.295×10^{-2}

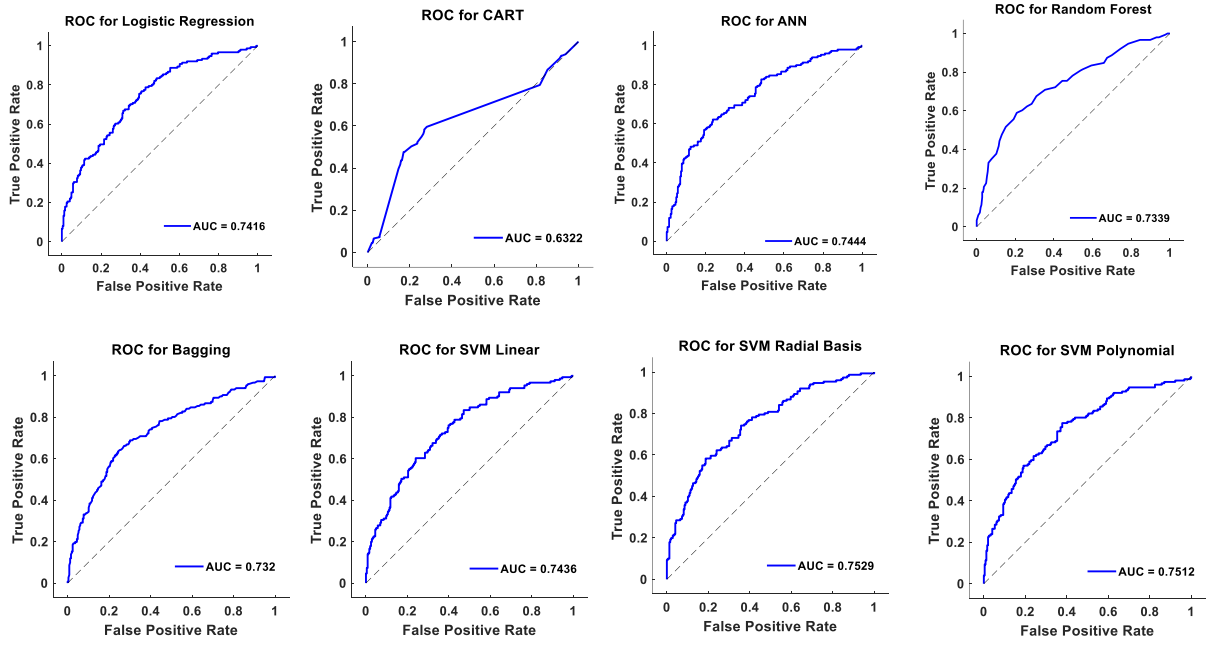


Figure 9. ROC plots for the models applied to the reduced dataset.

Next, the top 13 important attributes were used to train a CART model. CART models were constructed and pruned to improve generalization, with node splitting based on the Gini impurity criterion. Model selection via 10-fold cross-validation yielded an optimal tree with 10 terminal nodes. The resulting CART tree had an accuracy of 67.3% and AUC of 0.6322 for test data.

Based on the reduced training dataset, we trained multiple random forest models with several chosen `ntree` and `mtry` values (cf. Section 4.1). Each of these models were then used to classify the test data and the model producing the highest classification accuracy was chosen. Figure 9 displays the ROC curve for the random forest model with AUC value of 0.7339 for the test data. We built several bagging models based on the reduced training data using the same setting in the case of the full data. The best model is selected based on the highest test classification accuracy.

Figure 9 shows the ROC curve for the bagging model with AUC of 0.7320. SVM hyperparameters were tuned using 10-fold cross-validation; for the linear kernel, the cost parameter was selected from the grid $(0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0)$. The best `cost` value obtained was 0.1. We used this model as the final model to classify the test dataset. Figure 9 presents the ROC curve for the SVM with a linear kernel. For the SVM with a radial basis kernel, the cost and sigma parameters were jointly tuned via 10-fold cross-validation over the grids $(0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2)$ and $(0.001, 0.01, 0.1, 1, 10)$, respectively. Among the evaluated hyperparameter configurations, the optimal model was achieved with `cost` = 0.1 and `sigma` = 0.1. Figure 9 also shows the ROC curve obtained from the test data for the SVM with a radial basis kernel. For the SVM with a polynomial kernel, we tuned the degree of polynomial, `cost`, and `scale` hyperparameters on the grids $(2.0, 3.0)$, $(0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1.0, 1.5, 2.0)$, and $(0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1.0)$, respectively. The best values for the hyperparameters `degree`, `cost`, and `scale` obtained were 2.0, 1.0, and 0.01, respectively. Table 5 shows the AUC values for the models for the test dataset. Finally, we trained an ANN model on the reduced dataset. The cross-validation test accuracy was 70.8%, with a specificity of 53.0% and sensitivity of 81.7%, resulting in an AUC of 0.7444. Under feature reduction, overall screening performance declines across all models, as reflected by lower F1 scores, and changes in relative model behavior become apparent. The SVM with a radial basis kernel achieves the highest F1 score (0.6383), followed closely by the ANN (0.6327) and the SVM with a polynomial kernel (0.6252), indicating comparatively better balance between case detection and error control under limited feature availability. Bagging (0.6203) and random forest (0.6183) show moderate robustness, while logistic regression (0.6113) and the SVM with a linear kernel (0.6034) experience greater degradation. CART exhibits the lowest F1 score (0.5995), suggesting reduced effectiveness in the reduced-feature setting. Overall, these results indicate that although feature sparsification negatively impacts all models, kernel-based SVMs and ANNs retain relatively stronger screening performance, whereas tree-based and linear models are more sensitive to the loss of input information.

Figure 4b presents the PR curves obtained using the reduced feature set. Although overall AP values decrease slightly compared with the full feature models, the relative ranking of methods remains consistent. The SVM-RBF model continues to demonstrate the highest AP performance (0.6730), followed by SVM-Poly (0.6677) and ANN (0.6541). Together, Figure 4a,b demonstrate that precision–recall analysis provides an additional clinically relevant assessment to ROC curves by explicitly quantifying sensitivity–precision trade-offs under class imbalance. Figure 5b presents the training ROC curves for the reduced-feature models, where ensemble methods (Random Forest and Bagging) again achieve near-perfect training AUC, SVM and ANN models exhibit lower but non-saturated training performance.

Table 5. Performance results on reduced test dataset.

Method	Accuracy	Sensitivity	Specificity	AUC	F1 Score
Random forest	0.710	0.632	0.747	0.7339	0.6183
Bagging	0.710	0.629	0.755	0.7320	0.6203
SVM linear	0.698	0.615	0.740	0.7436	0.6034
SVM radial basis	0.723	0.652	0.760	0.7529	0.6383
SVM polynomial	0.713	0.643	0.746	0.7512	0.6252
ANN	0.708	0.817	0.530	0.7444	0.6327
CART	0.673	0.723	0.577	0.6322	0.5993
Logistic regression	0.683	0.768	0.543	0.7416	0.6114

The classification performance of the six machine learning techniques was assessed using metrics including the area under the curve (AUC), accuracy and F1 score on both the full and reduced test datasets.

We choose the best model based on AUC and sensitivity at a fixed specificity. This is due to the fact that AUC is a statistic summarized over all possible thresholds. In addition, by fixing the test size (Type I error rate), we aimed to maximize statistical power (1—Type II error rate). Therefore, we also compare the performance based on sensitivity by fixing the specificity. Thus, the analysis incorporates two performance metrics, of which one is evaluated across all possible decision thresholds, while the other is measured at a fixed threshold.

Figure 3b shows the ROC curves and a vertical line fixed at 0.8 specificity. The point of intersection between the line and the curve corresponds to the model's sensitivity value. The specificity was fixed at 0.8 so that it is fair to compare all techniques based on their sensitivity in detecting COVID-19. Table 6 presents a summary of the simulation results for test performance, as measured by AUC and sensitivity at fixed specificity. The SVM radial basis model has the best classification performance in terms of the test AUC of 0.7529 for reduced data. From Table 6 and Figure 3b SVM radial basis has sensitivity of 0.5616 at a fixed operating point. Although reduced-feature models do not exceed the performance of full-feature models, their inclusion provides important insight into model robustness. Certain methods, such as ANN and logistic regression, retain comparatively stable performance under feature reduction, whereas kernel-based SVMs exhibit greater sensitivity to the loss of input information. These findings highlight trade-offs between predictive performance and operational simplicity that are critical for real-world screening deployment.

Table 6. Summary of performance results for full and reduced test datasets.

	Full Model		Reduced Model	
Method	**Sensitivity (Specificity = 0.8)**	**AUC**	**Sensitivity (Specificity = 0.8)**	**AUC**
Random forest	0.5759	0.7529	0.5611	0.7339
Bagging	0.5687	0.7473	0.5501	0.7219
SVM linear	0.5572	0.7651	0.5203	0.7436
SVM radial basis	0.6032	0.7828	0.5616	0.7529
SVM polynomial	0.5917	0.7797	0.5504	0.7512
ANN	0.5275	0.7295	0.5579	0.7444
CART	0.4641	0.6821	0.4869	0.6322
Logistic regression	0.4863	0.7621	0.4947	0.7416

To assess potential overfitting, we additionally evaluated model performance on the training data. Figure 5a,b present the training ROC curves for the full-feature and

reduced-feature datasets, respectively. Ensemble-based models, including random forest and bagging, achieve near-perfect training AUC values, reflecting their high capacity to fit the training data. In contrast, kernel-based SVM models and the ANN exhibit moderately high but non-saturated training performance, indicating more constrained model complexity.

Comparison between training- and test-set results reveals the expected degradation in AUC across all models, particularly for high-capacity classifiers. This gap between training and test performance suggests controlled generalization rather than severe overfitting. Together, the inclusion of training ROC analyses improves transparency in model evaluation and supports the robustness of the reported test-set performance for clinical screening applications.

5. Conclusions and Future Work

A false negative result for COVID-19 can not only delay treatment, but also increase the risk of viral transmission. In this study, we investigated six different machine learning models for COVID-19 diagnosis using the UTMB dataset [26]. The random forest methodology was used as a variable selection method to reduce the full data dimensionality based on the variable importance of models. Predictive modeling with and without feature selection/model reduction was performed separately. The case study results indicate that the SVM with a radial basis kernel achieved the best overall predictive performance and offers a low-cost screening tool to support clinical diagnosis and treatment planning. Our results also indicate that most of the methods considered, except for CART, also performed well. Even though our paper provides an automated procedure to screen for COVID-19 using non-image data, the accuracy and AUC are not very high due to the possible exclusion of other important predictors, which would require further investigation. As a robustness check, future work will examine model performance across a range of clinically plausible specificity constraints (e.g., 0.7–0.9) to quantify how conclusions vary with the operating point and to support site-specific threshold selection. We also intend to explore the inclusion of image data from chest X-ray scans in addition to the non-image data analyzed in this paper to improve COVID-19 detection accuracy. Although this study focuses on predictive benchmarking rather than explainability, future work will incorporate post hoc interpretability methods such as SHAP or LIME to provide instance-level explanations for kernel-based models, thereby enhancing clinical transparency and trust in real-world screening settings.

Author Contributions: Conceptualization, H.X., A.T.A. and M.P.; methodology, M.P.; software, H.X. and A.T.A.; validation, H.X., A.T.A. and M.P.; resources, M.P., T.-L.T. and E.W.; writing—original draft preparation, H.X. and A.T.A.; writing—review and editing, M.P., S.C.M., Y.W., M.F.R., T.-L.T. and S.M.; supervision, T.-L.T. and E.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the National Science Foundation [DMS-2402544 and DUE-2216396], the Department of Education [Award #P116S210004], and the National Institute on Minority Health and Health Disparities (NIMHD) [#U54MD007592].

Institutional Review Board Statement: As this study was conducted using publicly available data and all data were anonymized, ethical approval and informed consent were not required.

Data Availability Statement: The raw UTMB dataset, the `R` codes for data wrangling and imputation, and the complete set of analyses reported in this paper are available online [26,27,44].

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Mei, X.; Lee, H.C.; Diao, K.y.; Huang, M.; Lin, B.; Liu, C.; Xie, Z.; Ma, Y.; Robson, P.M.; Chung, M.; et al. Artificial intelligence–enabled rapid diagnosis of patients with COVID-19. *Nat. Med.* **2020**, *26*, 1224–1228. [CrossRef]
2. Heidari, A.; Jafari Navimipour, N.; Unal, M.; Toumaj, S. Machine learning applications for COVID-19 outbreak management. *Neural Comput. Appl.* **2022**, *34*, 15313–15348. [CrossRef]
3. Ihle-Hansen, H.; Berge, T.; Tveita, A.; Rønning, E.J.; Ernø, P.E.; Andersen, E.L.; Wang, C.H.; Tveit, A.; Myrstad, M. COVID-19: Symptoms, course of illness and use of clinical scoring systems for the first 42 patients admitted to a Norwegian local hospital. *Tidsskr. Nor. Laegeforening* **2020**, *140*.
4. Chow, E.J.; Schwartz, N.G.; Tobolowsky, F.A.; Zacks, R.L.T.; Huntington-Frazier, M.; Reddy, S.C.; Rao, A.K. Symptom screening at illness onset of health care personnel with SARS-CoV-2 infection in King County, Washington. *J. Am. Med. Assoc.* **2020**, *323*, 2087–2089. [CrossRef]
5. Kwekha-Rashid, A.S.; Abduljabbar, H.N.; Alhayani, B. Coronavirus disease (COVID-19) cases analysis using machine-learning applications. *Appl. Nanosci.* **2023**, *13*, 2013–2025. [CrossRef] [PubMed]
6. Luers, J.C.; Rokohl, A.C.; Loreck, N.; Wawer Matos, P.A.; Augustin, M.; Dewald, F.; Klein, F.; Lehmann, C.; Heindl, L.M. Olfactory and gustatory dysfunction in coronavirus disease 2019 (COVID-19). *Clin. Infect. Dis.* **2020**, *71*, 2262–2264. [CrossRef] [PubMed]
7. Moulaei, K.; Shanbehzadeh, M.; Mohammadi-Taghiabad, Z.; Kazemi-Arpanahi, H. Comparing machine learning algorithms for predicting COVID-19 mortality. *BMC Med. Inform. Decis. Mak.* **2022**, *22*, 2. [CrossRef] [PubMed]
8. Zimmerman, R.K.; Nowalk, M.P.; Bear, T.; Taber, R.; Clarke, K.S.; Sax, T.M.; Eng, H.; Clarke, L.G.; Balasubramani, G. Proposed clinical indicators for efficient screening and testing for COVID-19 infection using Classification and Regression Trees (CART) analysis. *Hum. Vaccines Immunother.* **2021**, *17*, 1109–1112. [CrossRef]
9. Russell, S.J.; Norvig, P. *Artificial Intelligence: A Modern Approach*; Pearson Education, Inc.: London, UK, 2010.
10. Ahuja, A.S. The impact of artificial intelligence in medicine on the future role of the physician. *PeerJ* **2019**, *7*, e7702. [CrossRef]
11. Alyasseri, Z.A.A.; Al-Betar, M.A.; Doush, I.A.; Awadallah, M.A.; Abasi, A.K.; Makhadmeh, S.N.; Zitar, R.A. Review on COVID-19 diagnosis models based on machine learning and deep learning approaches. *Expert Syst.* **2022**, *39*, e12759. [CrossRef]
12. Miller, R.A. Medical diagnostic decision support systems—past, present, and future: A threaded bibliography and brief commentary. *J. Am. Med. Inform. Assoc.* **1994**, *1*, 8–27. [CrossRef] [PubMed]
13. Szolovits, P.; Patil, R.S.; Schwartz, W.B. Artificial intelligence in medical diagnosis. *Ann. Intern. Med.* **1988**, *108*, 80–87. [CrossRef] [PubMed]
14. De Dombal, F.T. Computer-aided Diagnosis of Acute Abdominal Pain: The British Experience. In *Professional Judgment: A Reader in Clinical Decision Making*; Dowie, J., Elstein, A., Eds.; Cambridge University Press: Cambridge, UK, 1988; pp. 190–199.
15. Miller, R.A.; McNeil, M.A.; Challinor, S.M.; Masarie, F.E., Jr.; Myers, J.D. The INTERNIST-1/quick medical REFERENCE project—Status report. *West. J. Med.* **1986**, *145*, 816. [PubMed]
16. Rajkomar, A.; Dean, J.; Kohane, I. Machine learning in medicine. *N. Engl. J. Med.* **2019**, *380*, 1347–1358. [CrossRef]
17. Sarker, I.H.; Kayes, A.; Watters, P. Effectiveness analysis of machine learning classification models for predicting personalized context-aware smartphone usage. *J. Big Data* **2019**, *6*, 57 . [CrossRef]
18. Sarker, I.H.; Salim, F.D. Mining user behavioral rules from smartphone data through association analysis. In *Advances in Knowledge Discovery and Data Mining, Proceedings of the 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, 3–6 June 2018, Proceedings, Part I 22*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 450–461.
19. Yu, K.H.; Beam, A.L.; Kohane, I.S. Artificial intelligence in healthcare. *Nat. Biomed. Eng.* **2018**, *2*, 719–731. [CrossRef] [PubMed]
20. Jiang, F.; Jiang, Y.; Zhi, H.; Dong, Y.; Li, H.; Ma, S.; Wang, Y.; Dong, Q.; Shen, H.; Wang, Y. Artificial intelligence in healthcare: Past, present and future. *Stroke Vasc. Neurol.* **2017**, *2*, 230–243. [CrossRef]
21. Murdoch, T.B.; Detsky, A.S. The inevitable application of big data to health care. *J. Am. Med. Assoc.* **2013**, *309*, 1351–1352. [CrossRef]
22. Dilsizian, S.E.; Siegel, E.L. Artificial intelligence in medicine and cardiac imaging: Harnessing big data and advanced computing to provide personalized medical diagnosis and treatment. *Curr. Cardiol. Rep.* **2014**, *16*, 1–8. [CrossRef]
23. Rao, A.S.S.; Vazquez, J.A. Identification of COVID-19 can be quicker through artificial intelligence framework using a mobile phone–based survey when cities and towns are under quarantine. *Infect. Control Hosp. Epidemiol.* **2020**, *41*, 826–830.
24. Wang, S.; Kang, B.; Ma, J.; Zeng, X.; Xiao, M.; Guo, J.; Cai, M.; Yang, J.; Li, Y.; Meng, X.; et al. A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID-19). *Eur. Radiol.* **2021**, *31*, 6096–6104. [CrossRef]
25. Van Buuren, S. *Flexible Imputation of Missing Data*; Chapman & Hall/CRC: Boca Raton, FL, USA, 2012.
26. McCaffrey, P.E. UTMB Non-Image COVID-19 Clinical Dataset. 2024. Available online: https://github.com/pmccaffrey6/COVID-LOS (accessed on 28 November 2025).
27. Pokojovy, M. Data Wrangling and Imputation for the UTMB Non-Image COVID-19 Clinical Dataset. 2024. Available online: https://github.com/mpokojovy/COVID.screening.prep (accessed on 28 November 2025).
28. LaValley, M.P. Logistic regression. *Circulation* **2008**, *117*, 2395–2399. [CrossRef]

29. Breiman, L.; Friedman, J.; Stone, C.J.; Olshen, R.A. *Classification and Regression Trees*; CRC Press: Boca Raton, FL, USA, 1984.
30. Deming, S.; Morgan, S. Handbook of Chemometrics and Qualimetrics: Part A. *Technometrics* **1998**, *40*, 264. [CrossRef]
31. Ziegel, E.R. Handbook of Chemometrics and Qualimetrics, Part B. *Technometrics* **2000**, *42*, 218–219. [CrossRef]
32. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
33. Bauer, E.; Kohavi, R. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Mach. Learn.* **1999**, *36*, 105–139. [CrossRef]
34. Hastie, T.; Tibshirani, R.; Friedman, J.H.; Friedman, J.H. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*; Springer: New York, NY, USA, 2009; Volume 2.
35. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
36. Stitson, M.; Weston, J.; Gammerman, A.; Vovk, V.; Vapnik, V. *Theory of Support Vector Machines*; Technical Report, CSD-TR-96-17; University of London: London, UK, 1996.
37. Furey, T.S.; Cristianini, N.; Duffy, N.; Bednarski, D.W.; Schummer, M.; Haussler, D. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics* **2000**, *16*, 906–914. [CrossRef]
38. Pavlidis, P.; Wapinski, I.; Noble, W.S. Support vector machine classification on the Web. *Bioinformatics* **2004**, *20*, 586–587. [CrossRef] [PubMed]
39. Keerthi, S.S.; Lin, C.J. Asymptotic behaviors of support vector machines with Gaussian kernel. *Neural Comput.* **2003**, *15*, 1667–1689. [CrossRef]
40. Vert, J.P.; Tsuda, K.; Schölkopf, B. A primer on kernel methods. *Kernel Methods Comput. Biol.* **2004**, *47*, 35–70.
41. Musavi, M.T.; Ahmed, W.; Chan, K.H.; Faris, K.B.; Hummels, D.M. On the training of radial basis function classifiers. *Neural Netw.* **1992**, *5*, 595–603. [CrossRef]
42. Jamous, R.; ALRahhal, H.; El-Darieby, M. A new ANN-particle swarm optimization with center of gravity (ANN-PSOCog) prediction model for the stock market under the effect of COVID-19. *Sci. Program.* **2021**, *2021*, 6656150. [CrossRef]
43. Aggarwal, C.C. An Introduction to Neural Networks. In *Neural Networks and Deep Learning: A Textbook*; Aggarwal, C.C., Ed.; Springer: Cham, Switzerland, 2018; pp. 1–52.
44. Xu, H.; Anum, A.T. Utilizing Machine Learning Techniques for Computer-Aided COVID-19 Screening Based on Clinical Data. 2024. Available online: https://github.com/HonglunXu/Machine-Learning-Techniques-for-COVID-19.git (accessed on 28 November 2025).
45. Tibshirani, R. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **1996**, *58*, 267–288. [CrossRef]
46. Krzywinski, M.; Altman, N. Classification and regression trees. *Nat. Methods* **2017**, *14*, 757–758. [CrossRef]
47. Yitzhaki, S.; Schechtman, E. *The Gini Methodology: A Primer on a Statistical Methodology*; Springer: Berlin/Heidelberg, Germany, 2013.
48. Rakotomamonjy, A. Variable selection using SVM-based criteria. *J. Mach. Learn. Res.* **2003**, *3*, 1357–1370.
49. Aminghafari, M.; Cheze, N.; Poggi, J.M. Multivariate denoising using wavelets and principal component analysis. *Comput. Stat. Data Anal.* **2006**, *50*, 2381–2398. [CrossRef]
50. Bruce, P.; Bruce, A.; Gedeck, P. *Practical Statistics for Data Scientists: 50+ Essential Concepts Using R and Python*; O'Reilly Media: Sebastopol, CA, USA, 2020.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.