Analyzing Covid-19 Data Using Various Algorithms

Ayah Krajah
Department of Computer Science, King Abdullah II School of Information Technology (KASIT)
University of Jordan
Amman,Jordan
ayh8180643@ju.edu.jo

Yara Feras Almadani
Faculty of medicine
University of Jordan
Amman,Jordan
yar0182406@ju.edu.jo

Heba Saadeh
Department of Computer Science, King Abdullah II School of Information Technology (KASIT)
University of Jordan
Amman,Jordan
heba.saadeh@ju.edu.jo

Azzam Sleit
Department of Computer Science, King Abdullah II School of Information Technology (KASIT)
University of Jordan
Amman,Jordan
azzam.sleit@ju.edu.jo

2021 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT) | 978-1-6654-4293-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/JEEIT53412.2021.9634124

***Abstract*— The world strives to combat Covid-19, which has spread very quickly across the world after the disease was first confirmed in Dec 2019 in Wuhan District, China. This disease has infected millions and kills thousands in most countries until today. Machine learning and artificial intelligence are promising technologies that have an effective role in developing business in many sectors. Recent studies have shown the importance of using artificial intelligence and machine learning in the health sector. The results show that they lead to increased processing capability, reliability, and even superiority over the human's performance in particular healthcare tasks. In this research, we applied several machine learning algorithms such as Logistic Regression (LR), Linear Discriminant Analysis (LDA), Classification and Regression Trees (CART), Support Vector Machines (SVM), Gaussian Naive Bayes (NB), and k-Nearest Neighbors (KNN) on Covid-19 dataset provided via Kaggle website to predict the patient's death or survival depending on the patient's health status and some other factors. To evaluate the performance of these algorithms, we used the confusion matrix.**

***Keywords*—covid-19, machine learning, artificial intelligence, classification algorithms.**

I. INTRODUCTION

Covid-19 was declared a global pandemic by the World Health Organization (WHO) and was reported as an international concern for public health. After infection, symptoms begin to appear in a period ranging from (2-14) days. The popular symptoms of Covid-19 are dry cough, tiredness, and fever [1]. In general, the early symptoms of Covid-19 are anosmia and ageusia. Furthermore, several symptoms can appear, such as dyspnea, polymyalgia, chills, sore throat, nasal discharge, and headaches [1]. Because of the quick spread of this pandemic and its socio-economic impact in our countries, it is necessary to discover solutions and strategies that reduce the burden on health systems.

Machine learning can contribute to restricting the spread of this disease by predicting risks [2]. Early ML experiments are promising and can provide an effective method to alleviate the disease burdens. There are many powerful machine learning applications, such as classification, which is widely applied in prediction.

Classification is one of the machine learning applications requiring particular algorithms to analyze and classify certain data depending on shared or similar properties. Classification problems require a training dataset for prediction with several examples of suitable inputs and outputs to start the learning process. The training dataset must adequately represent the problem, and each class label should have multiple examples [2]. The values of class labels are usually strings, e.g., disease is “infectious”, or “not-infectious”. Before entering these class labels into an algorithm for the modeling process, they must be converted or mapped to numeric values, e.g., “infected =0” and “not-infected=1”[3]. The performance of classification algorithms is measured based on the results, one of the measures used to evaluate performance based on the predicted classes is accuracy.

In this research, several algorithms were applied to solve this classification problem. The eight algorithms selected include:

A) Linear Algorithms: Logistic Regression (LR) and Linear Discriminant Analysis (LDA).

B) Nonlinear Algorithms: Classification and Regression Trees (CART), Support Vector Machines (SVM), Gaussian Naive Bayes (NB), and k-Nearest Neighbors (KNN).

A confusion matrix is an N x N matrix applied for measuring the performance of a classification model where the output can be two or more classes, and N indicates the number of target classes [4]. This matrix compares the actual target values with those predicted by the machine learning classifier. See Fig.1.

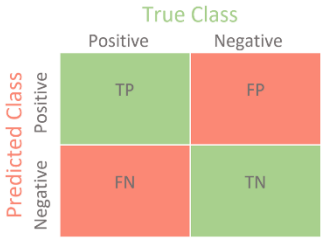

Figure 1. The confusion matrix for binary classification

The columns represent the actual values of the target variable, and the rows represent the predicted values of the target variable[4]. In this matrix, the prediction is divided into two categories: the correct prediction and the wrong prediction. The correct predictions are categorized into True Positive (TP), and True Negative (TN). The wrong predictions are categorized into False Positive (FP), and False Negative (FN). TP shows that the predicted value matches the actual value, TN indicates that the actual value was negative while the model predicted a negative value, FP shows that the actual value was negative, but the model predicted a positive value, and FN indicates that the actual value was positive, but the model predicted a negative value. The Confusion Matrix provides several metrics to evaluate the model's performance, such as accuracy, precision, recall, and f1-Score.

Accuracy is a suitable basic metric to estimate the model's performance concerning how many correct predictions are classified by the model for the entire dataset used in the classification[4]. Accuracy is an effective measure of balanced data but does not perform well in unbalanced data. It is calculated by the following formula:

$$\text{Accuracy} = \frac{TN+TP}{TN+FP+TP+FN} \quad (1)$$

Precision is utilized as a measure to show the accuracy of a positive prediction [4]. It is calculated by the following formula:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

The recall demonstrates how many of the real positive cases we have been able to predict with our model correctly [4]. It is calculated by the following formula:

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

Two models with low accuracy and high recall or vice versa are difficult to compare. So, we use the F-score to make them comparable. The F1-score helps to measure recall and precision simultaneously [4]. It is calculated by the following formula:

$$\text{F1-Score} = \frac{2*Recall*Precision}{Recall+Precision} \quad (4)$$

II. Related Works

A. The use of machine learning to classify COVID-19 patients

In this research [5], they applied machine learning classifiers such as Logistic Regression, Multinomial Naive Bayes, Support Vector Machine (SVM), and Decision Trees on textual clinical reports that contain symptoms of corona virus and other viruses, they classifiers these reports into four classes COVID, ARDS, SARS and Both (COVID, ARDS).The best results were obtained from the Multinomial Naive Bayes algorithm and Logistic Regression algorithm, with a testing accuracy of (96.2%). Also in [6] they applied a random Forest model boosted by the AdaBoost algorithm. The model aims to predict the seriousness of the case and the potential outcome by using the geographical, travel, health, and demographic details of COVID-19 patients. The model showed accuracy (94%) and F1-score (0.86%).

Moreover, authors in [7] implemented machine learning to reanalyzed "COVID-19" data from 151 published studies, they proposed to construct a more precise diagnostic model of "COVID-19" based on patients' symptoms and routine test outcomes. Furthermore, based on clinical variables, a computational classification model for distinguishing between "COVID-19" patients and influenza patients was produced. Conclusively, the (XGBoost) model trained to get a sensitivity (92.5%) and specificity (97.9%) in the distinguishing process between "COVID-19" patients and influenza patients. In [8] they presented an early-stage detection of "COVID-19" through the Support Vector Machine algorithm. The detection process was performed on abdominal Computed Tomography (CT) images. The classification performance was evaluated using "sensitivity", "precision", "specificity", "accuracy", and "F1-score" metrics. The highest classification accuracy was (99.68%).

In [9], they proposed building effective deep learning models, trained with PA chest X-ray images. The study conducted 25 different types of augmentations on the original images to enhance the dataset size and produce generalized models. In addition, for training and testing of the classification models, the transfer learning approach was used. The incorporation of two best-performing models (each trained on (286) images, rotated by 120° or 140° angle) showed the highest prediction accuracy for normal, "COVID-19", "non-COVID-19", "pneumonia", and tuberculosis images.

B. Hybridizing different classification approaches for better results

They combined the strengths of Convolution Neural Networks (CNN) with a swarm-based features selection algorithm to extract the most relevant features [10]. The authors evaluated their proposed approach on two public datasets of Covid-19 X-ray images. The sources of these images are cardiothoracic radiologists, researchers, and the Kaggle repository. They obtained the best results in the datasets when compared to a set of algorithms used in the extraction features. Moreover, the proposed approach outperforms many traditional CNN approaches and recent research on "COVID-19" images. In [11] they implemented

many classification algorithms such as Naive Bayes, KNN, Support Vector Machine, and Decision Tree. They aimed to determine how strong the impact of the "Coronavirus" is on some of the patient's health circumstances. The Support Vector Machine algorithm showed the best performance with accuracy (100%).

III.METHODOLOGY

This study applied several machine learning algorithms such as Logistic Regression (LR), Linear Discriminant Analysis (LDA), Classification and Regression Trees (CART), Support Vector Machines (SVM), Gaussian Naive Bayes (NB), and k-Nearest Neighbors (KNN), to predict if the patient will die or live depending on its health status and some other circumstances. The steps that applied are listed as follows:

1.Problem Definition

The dataset used in this study called "COVID-19 patient pre-condition dataset" obtained from Kaggle site, reported by the Mexican government [12]. This dataset includes (566,602) cases related to Covid-19. For more details about the dataset see Table. I.

2. Dataset Pre-processing

A new column named (Alive) was added to indicate the patient's death or survival. The feature type of this column is Numeric where (Dead= 0, Alive=1). The unspecified values (97,98,99) were removed from rows since these values do not provide any useful information, and are therefore considered null values. After that, a function called dropna() applied to remove rows and columns with Null/NaN values., As a result, the new dimension of the dataset is equal (59,225 x 23).

The first graph of Fig.2 shows that many patients tested positive on the same day that the symptoms occurred. The second graph shows that half of the positive patients with 0-8 days of symptoms were entered into the ICU. Fig.3 shows that most people who tested positive are between the ages of 40 and 80. Half of the tested-positive patients were entered into ICU.

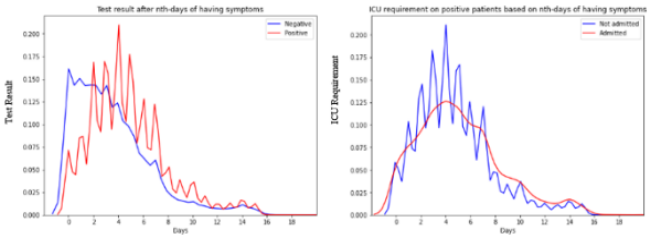

Figure 2. Test result and ICU requirement based on the number of days passed since the symptoms occurred.

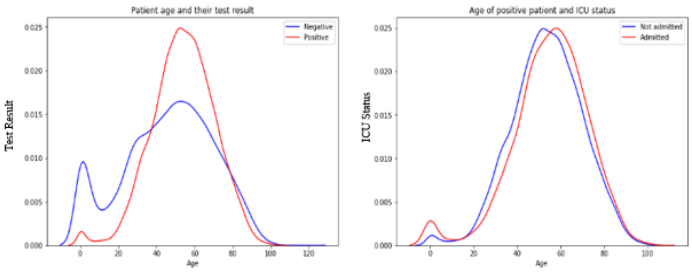

Figure 3. Test result and ICU requirement based on age

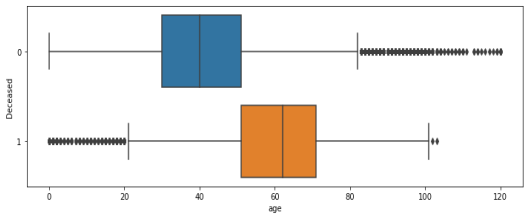

Figure 4. Distribution of patient's age in the two classes (Alive: Blue and Dead: Orange).

3. Features Selection

Feature selection is a method of selecting the features in your data that contribute the most to the prediction variable or output [13]. There are advantages to performing feature selection before modeling your data such as reduce data redundancy which means there is less chance of making decisions based on noise and this will reduce overfitting, and help the algorithms to train faster and thus reducing the time and complexity. There are many feature selection techniques that are easy to use and give good results such as:

3.1. Feature Importance

Feature importance is an inbuilt class that is included with Tree-Based Classifiers. This method is used to estimate the importance of the features related to the prediction variable.

As shown in Fig.5, the intubed, covid-res, pneumonia, and ICU are the most features that are significant to the prediction variable (Alive).

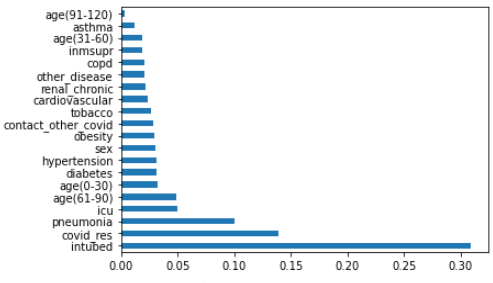

Figure 5. Bar chart of feature importance.

3.2 Univariate Selection

Statistical tests can be applied to select those features that have the strongest relationship with the prediction variable. Table. II shows the Score of relation with the prediction variable for each feature.

Table I. Dataset Description

No	Feature Name	Description	No	Feature Name	Description
1	Id	ID number of patient	13	Asthma	1 = YES, 2 = NO, 98/97/99= not specified
2	Sex	1= Women, 2=Man	14	Immunosuppressin	1 = YES, 2 = NO, 98/97/99= not specified
3	Patient type	1= not hospitalized, 2=hospitalized	15	Hypertension	1 = YES, 2 = NO,98/97/99= not specified
4	entry date	entry date to the hospital	16	Other_diseases	1 = YES, 2 = NO, 98/97/99= not specified
5	datesymptoms	The date of show symptoms	17	Cardiovascular	1 = YES,2 = NO, 98/97/99= not specified
6	date_died	"9999-99-99" = recovered	18	Obesity	1 = YES, 2 = NO, 98/97/99= not specified
7	Intubated	1 = YES, 2 = NO 97/98/99= not specified	19	renal_chronic	1 = YES,2 = NO, 98/97/99= not specified
8	Pneumonia	1 = YES, 2 = NO 97/98/99= not specified	20	tobacco (Smoker)	1 = YES, 2 = NO, 98/97/99= not specified
9	Age	age of the patient	21	contact_other_covid	Connect another covid19 patient
10	Pregnant	1 = YES, 2 = NO 98/97/99= not specified	22	Icu	1 = YES, 2 = NO, 98/97/99= not specified
11	Diabetes	1 = YES, 2 = NO 98/97/99= not specified	23	Covid-res	1 = POSITIVE, 2 = NEGATIVE, 3 = in awaiting process
12	COPD	1 = YES, 2 = NO, 98/97/99= not specified			

Table II. Score of relation with prediction variable for each feature.

No	Features	Score	No	Features	Score
1	**intubed**	6451.11	**11**	**obesity**	**120.68**
2	**icu**	1388.29	12	renal_chronic	85.40
3	**age(61-90)**	1250.58	13	cardiovascular	52.69
4	**age(0-30)**	1099.03	14	copd	34.83
5	**pneumonia**	983.62	15	sex	34.70
6	**covid_res**	829.31	16	asthma	23.71
7	**diabetes**	519.44	17	age(91-120)	12.18
8	**hypertension**	487.72	18	inmsupr	6.97
9	**contact_other_covid**	189.842	19	tobacco	4.43
10	**age(31-60)**	122.93	20	other_disease	4.42

11 features (marked as bold in Table II) were selected that have the highest scores.

4. Dataset Validation

The machine learning algorithms must be evaluated on data that is not used to train the algorithms (un-seen data). So, the validation hold-out set method was applied where (80%) for training for training and (20%) for and test and evaluate the accuracy of algorithms.

5. Algorithms Evaluation: Baseline

The (K-fold cross-validation) is a good standard test harness configuration in this situation, 10-fold cross-validation is used. As shown in Table. III, Algorithms were compared by showing the accuracy of mean and standard deviation.

Table III . Accuracy of mean and standard deviation

No	Algorithm	Accuracy	No	Algorithm	Accuracy
1	LR	0.828	5	NB	0.763
2	LDA	0.811	6	RF	0.820
3	KNN	0.810	7	SVM	0.829
4	CART	0.812	8	MLP	0.810

These are just mean accuracy values. It is always best to consider the distribution of accuracy values computed across cross-validation folds. See Fig.6

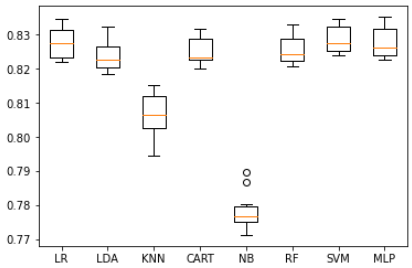

Figure 6.Distribution of accuracy values computed across cross-validation folds.

The results show a tight distribution for all algorithms except the KNN algorithm, which is encouraging.

6. Algorithms Evaluation: Normalized Data

It is preferable to test the same algorithms using a standardized copy of the dataset. The data is transformed to have a mean value of zero and a standard deviation of one for each feature. When data is transformed, it must also evade data leakage, an excellent approach to minimize leaking is using pipelines that standardize the data and construct the model for each fold in the cross-validation test harness. That way, we will have a good idea and fair evaluation of how each model will perform with standardized data on unseen data.

As Shown in (Table. IV) LDA, KNN, and CART are still doing well. NB provide the same result after scaled. RF and MLP performance affected with a little different. Moreover, the standardization of the data has lifted the skill of (SVM) and (LR) to be the most accurate algorithms tested so far.Fig.7 shows the distribution of accuracy values after standardizing the data.

Table IV. Results after Standardize Data

No	Algorithm	Accuracy	No	Algorithm	Accuracy
1	ScaledLR	0.828225	5	ScaledNB	0.763033
2	ScaledLDA	0.811090	6	ScaledRF	0.810611
3	ScaledKNN	0.810182	7	ScaledSVM	0.827757
4	ScaledCART	0.812421	8	ScaledMLP	0.802865

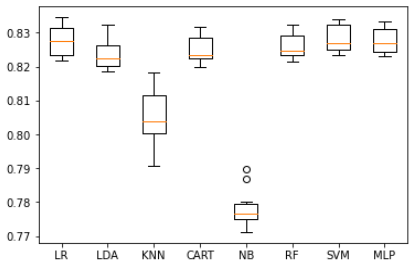

Figure 7. Distribution of accuracy values after standardizing the data.

7. Algorithms Tuning

In this section, tuning the parameters are investigated for two algorithms that showed the best results in the previous section (SVM and LR), grid search is an approach used to parameter tuning that finds the best hyperparameters of a model which provide the highest accuracy.

Logistic Regression (LR)
Best: 0.83 using {'C': 0.01, 'penalty': 'l2', 'solver': 'liblinear'}
Support Vector machine (SVM)
Best :0.82 using {kernel="linear", C=0.025}

8. Models Finalization

In this section, we will finalize the models by training them on the entire training dataset and make predictions for the hold-out validation dataset to confirm the results. As shown in Table. V, We summarize the performance result of the two machine learning methods. In terms of accuracy, recall, and f1-score, we can notice that the logistic regression is the most accurate with (83%)(96%)(90%)respectively, the performance of the LR was the most simple, fastest algorithm and provide satisfying results, while the Support vector machine has the highest precision but it was slowest in implementation.

Table V.The Results After Finalize The Models

Algorithm	**Result for each class**				**Accuracy**
LR		Precision	recall	f1-score	0.83
	0	0.69	0.30	0.42	
	1	0.84	0.96	0.90	
	Avg	0.84	0.96	0.90	
SVM	0	0.62	0.35	0.45	0.82
	1	0.85	0.94	0.89	
	Avg	0.85	0.94	0.89	

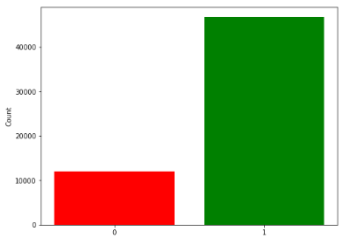

Figure8. Distribution of a predictive variable (Alive)

The class imbalance is a common problem in machine learning, especially in classification problems, in which the distribution of instances across classes is uneven or biased (see Fig.8). Imbalance data can hinder our model accuracy big time. There are several methods for dealing with the imbalanced dataset such as Re-sampling [14], which is one of the most often used and popular approaches. In this technique, before entering the data as input to the machine learning algorithm, we must focus on balancing the classes in the training data. The primary purpose of class balancing is to either raise the minority class's frequency or reduce the dominant class's frequency. This is done to ensure that each class has roughly the same number of instances. There are two resampling techniques the oversampling and undersampling. Over-Sampling is a technique for increasing the number of examples in the minority class by randomly repeating them to provide a more representative sample of the minority class. Under-sampling seeks to achieve a more balanced class distribution by removing majority class examples at random. This is repeated until the number of instances of the majority and minority classes is equal. In this research, we implemented oversampling and undersampling to handle the imbalanced dataset problem. One of the oversampling techniques is SMOTENC.

SMOTE (Synthetic Minority Oversampling Technique) works by selects a point from the minority class randomly and computes its k-nearest neighbors[14], creating a new point in the middle of the two current points, and adding it to the sample., Since our dataset is binary, It is difficult to compute a (midpoint) between two binary or categorical data points, so there is an updated version of SMOTE that allows for use of binary or categorical data called SMOTENC (Synthetic Minority Oversampling Technique for Nominal and Continuous). SMOTENC is taking the most often occurring category of nearest neighbors to a minority class point.

Before Smotenc: Counter ({1: 37397, 0: 9618})
After Smotenc: Counter ({1: 46709, 0: 46709})

After applied the SMOTE-NC technique, the class distribution of a data set was adjusted to be balanced. As a result, introducing bias to the model was avoided. Furthermore, as shown in (Table. VI) the accuracy for each model (LR, SVM) became lower than before (0.72%),(0.71%), which means the SMOTE-NC decreased the performance of the classifiers(LR, SVM).

Other methods to handle this problem are Random Under Sampler and Near-Miss [14]. These methods are examples of popular under sampling techniques. Random Under Sampler is a simple and quick technique to balance data by picking a portion of data at random for the targeted classes. Under-sample the majority class by randomly selecting samples with or without replacement.

Before undersample: Counter ({1: 37658, 0: 9722})
after undersample: Counter ({0: 12165, 1: 12165})

While the NearMiss using distance instead of resampling the minority class, by the distance the majority class equal to the minority class, when two points in the distribution from different classes are relatively near to each other, this technique removes the data point from the majority class, attempting to balance the distribution.

Before NearMiss: Counter ({1: 37658, 0: 9722})
After NearMiss: Counter ({0: 12165, 1: 12165})

The under-sampling techniques (Random Under Sample and Near-Miss) solved the imbalanced dataset problem by randomly selecting examples from the majority class and deleting them from the training dataset. Moreover, as shown in (Table. VI) the two techniques decreased the accuracy of the two models like SMOTE NC. The Re-sampling strategies applied in this research provide an effective solution to the imbalance problem, but decreased the performance of the classifiers and increase the computational effort.

Table VI. Experimental Results

Resampling Techniques	Algorithm	Accuracy	F1-Score	Precision	Recall
SMOTENC	*LR*	0.72	0.72	0.72	0.72
	SVM	0.71	0.72	0.70	0.74
Random Under Sampler	*LR*	0.72	0.73	0.71	0.74
	SVM	0.70	0.72	0.68	0.75
NearMiss	*LR*	0.73	0.75	0.70	0.80
	SVM	0.72	0.77	0.65	0.93

IV. Conclusion

Machine learning and artificial intelligence are promising technologies that have an effective role in developing business in many sectors. In this research, we applied several machine learning algorithms on the Covid-19 dataset provided via the Kaggle website. The results showed the importance of employing machine learning algorithms to reduce the burdens caused by covid-19 on the health sector. It also demonstrated the capacity of a trained model to predict the covid-19 patient's death or survival depending on the patient's health status.

REFERENCES

[1] Fung, To Sing and Liu, Ding Xiang*," Similarities and Dissimilarities of COVID-19 and Other Coronavirus Diseases*", Annual Review of Microbiology,Vol.75,Jan 2021.

[2] Samuel Lalmuanawmaa,∗, Jamal Hussaina, Lalrinfela Chhakchhuak b*" Applications of machine learning and artificial intelligence for Covid-19 (SARS-CoV-2) pandemic: Areview"* Chaos Solitons Fractals,Vol.139,June 2020 .

[3] Kotsiantis, S.. "*Supervised Machine Learning: A Review of Classification Techniques."* Informatica (Slovenia) 31 (2007): 249-268.

[4] Vakili, M., Ghamsari, M., & Rezaei, M. (2020*)." Performance Analysis and Comparison of Machine and Deep Learning Algorithms for IoT Data Classification".* ArXiv, abs/2001.09636..

[5] Khanday, A.M.U.D., Rabani, S.T., Khan, Q.R. et al*." Machine learning based approaches for detecting COVID-19 using clinical text data",* International Journal of Information TechnologyVol.12,pp 731-739, Sep 2020.

[6] Celestine Iwendi 1 *, Ali Kashif Bashir 2 , Atharva Peshkar 3 , R. Sujatha4 , Jyotir Moy Chatterjee5 , Swetha Pasupuleti 6 , Rishita Mishra7 , Sofia Pillai 8 and Ohyun Jo9 *,"COVID-19 Patient Health Prediction Using Boosted Random Forest Algorithm",* Front. Public Health, Vol. 8, July2020

[7] Wei Tse Li, Jiayan Ma*," Using machine learning of clinical data to diagnose COVID-19: a systematic review and meta-analysis",* Li et al. BMC Medical Informatics and Decision Making,Vol 20,Sep 2020.

[8] Mucahid Barstugan , Umut Ozkaya , Saban Ozturk, "*Coronavirus (COVID-19) Classification using CT Images by Machine Learning Methods",* arXiv.org,Mar 2020.

[9] Arun Sharma , Sheeba Rani , and Dinesh Gupta*,"Artificial Intelligence-Based Classification of Chest X-Ray Images into COVID-19 and Other Infectious Diseases",* International Journal of Biomedical Imaging,Vol.2020, 2020.

[10] AhmedT. Sahlol1 , DaliaYousri2 *," COVID 19 image classifcation using deep features and fractional order marine predators algorithm*",Scientific Reports,Vol.10,Sep 2020.

[11] Yavuz ÜNAL, Muhammet Nuri DUDAK*," Classification of Covid-19 Dataset with Some Machine Learning Methods",* journal of amasya university the institute of sciences and technology ,Vol.1,36-44, Jun 2020.

[12] Tanmoy Mukherjee*,"COVID-19 patient pre-condition dataset"*.(2020)byKaggle.com.https://www.kaggle.com/tanmoyx/covid19-patient-precondition-dataset?select=covid.csv

[13] S. G. Devi and M. Sabrigiriraj, "*Feature Selection, Online Feature Selection Techniques for Big Data Classification: - A Review,"* 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT), 2018, pp. 1-9, doi: 10.1109/ICCTCT.2018.8550928.

[14] Mohammed, Roweida & Rawashdeh, Jumanah & Abdullah, Malak. (2020). *"Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results*". 243-248. 10.1109/ICICS49469.2020.