

Article

Diagnosis of COVID-19 Using Chest X-ray Images and Disease Symptoms Based on Stacking Ensemble Deep Learning

Abdulaziz AlMohimeed ¹, Hager Saleh ^{2,*}, Nora El-Rashidy ³, Redhwan M. A. Saad ⁴, Shaker El-Sappagh ^{5,6,*} and Sherif Mostafa ¹

- ¹ College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh 13318, Saudi Arabia; aialmohimeed@imamu.edu.sa (A.A.); sherif.mostafa@fcih.svu.edu.eg (S.M.)
² Faculty of Computers and Artificial Intelligence, South Valley University, Hurghada 84511, Egypt
³ Machine Learning and Information Retrieval Department, Faculty of Artificial Intelligence, Kafrelsheikh University, Kafrelsheikh 13518, Egypt; noura.alrashidy@ai.kfs.edu.eg
⁴ College of Informatics, Midocean University, Moroni 8722, Comoros; redhwan@midocean.edu.km
⁵ Faculty of Computer Science and Engineering, Galala University, Suez 435611, Egypt
⁶ Information Systems Department, Faculty of Computers and Artificial Intelligence, Benha University, Banha 13518, Egypt
* Correspondence: hager.saleh@fcih.svu.edu.eg (H.S.); shaker.elsappagh@gu.edu.eg (S.E.-S.)

Abstract: The COVID-19 virus is one of the most devastating illnesses humanity has ever faced. COVID-19 is an infection that is hard to diagnose until it has caused lung damage or blood clots. As a result, it is one of the most insidious diseases due to the lack of knowledge of its symptoms. Artificial intelligence (AI) technologies are being investigated for the early detection of COVID-19 using symptoms and chest X-ray images. Therefore, this work proposes stacking ensemble models using two types of COVID-19 datasets, symptoms and chest X-ray scans, to identify COVID-19. The first proposed model is a stacking ensemble model that is merged from the outputs of pre-trained models in the stacking: multi-layer perceptron (MLP), recurrent neural network (RNN), long short-term memory (LSTM), and gated recurrent unit (GRU). Stacking trains and evaluates the meta-learner as a support vector machine (SVM) to predict the final decision. Two datasets of COVID-19 symptoms are used to compare the first proposed model with MLP, RNN, LSTM, and GRU models. The second proposed model is a stacking ensemble model that is merged from the outputs of pre-trained DL models in the stacking: VGG16, InceptionV3, Resnet50, and DenseNet121; it uses stacking to train and evaluate the meta-learner (SVM) to identify the final prediction. Two datasets of COVID-19 chest X-ray images are used to compare the second proposed model with other DL models. The result has shown that the proposed models achieve the highest performance compared to other models for each dataset.

Keywords: machine learning; deep learning; ensemble learning; COVID-19; diagnosis; symptoms; stacking



Citation: AlMohimeed, A.; Saleh, H.; El-Rashidy, N.; Saad, R.M.A.; El-Sappagh, S.; Mostafa, S. Diagnosis of COVID-19 using Chest X-ray Images and Disease Symptoms Based on Stacking Ensemble Deep Learning. *Diagnostics* **2023**, *13*, 1968. <https://doi.org/10.3390/diagnostics13111968>

Academic Editors: Philippe A. Grenier and Chiara Martini

Received: 10 April 2023

Revised: 17 May 2023

Accepted: 30 May 2023

Published: 5 June 2023



Copyright: © 2023 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The leading cause of the global COVID-19 pandemic is the SARS-CoV-2 virus. Therefore, it has become necessary to find means that would effectively achieve early detection of people with COVID-19 and provide them with the care needed on time. In addition, all medical measures and precautions must be taken to separate patients infected with COVID-19 from other patients to reduce the spread of the disease or its fatal symptoms. The number of deaths due to coronavirus reached 6,517,058 based on global measures [1]. Furthermore, COVID-19 poses a severe challenge due to its ease of transmission and global lack of definitively viable therapies [2]. Many vaccines have been proven to expose users to many complications, including blood clots.

COVID-19 infection goes through three stages: the incubation period, acute COVID-19, and finally, COVID-19 recovery. The incubation period is the period between

the actual infection with the disease and the onset of symptoms in the patient. Acute COVID-19 is the time when the symptoms appear, such as fever, cough, fatigue, headache, congestion, or runny nose, among other COVID-19 symptoms. In addition, an essential step in the fight against this fatal illness would be a successful screening and diagnosis procedure to treat affected patients. In addition, an effective strategy in the fight against COVID-19 may be early detection utilizing chest X-ray pictures [3].

Therefore, effective and early COVID-19 diagnosis based on the symptoms and chest X-ray images will help to mitigate the coronavirus outbreak. Moreover, it will assist healthcare systems, including doctors, nurses, and medical staff, in protecting vulnerable patients. Artificial intelligence (AI) has been instrumental in the steady transition from laboratory to clinical and public health applications.

AI provides a wide range of approaches for analyzing complex data to advance understanding of the subject of COVID-19 [4–7]. AI employs machine learning (ML) and deep learning (DL) to produce algorithms that can be used in the clinical and biomedical fields for patient classification and stratification based on the pairing and processing of a wide range of available data sources, such as heart disease detection [8], polycystic ovary syndrome detection [9], and chronic kidney disease detection [10]. The most significant contribution is using AI to detect patients at higher risk early to treat those patients and control disease transmission. Furthermore, AI can help governments to manage the pandemic by early notification of COVID-19 outbreaks [11]. An ensemble classifier combines the results of several classifiers in a way that enables component models to balance out the deficiencies of each other. Ensemble learning has three types, stacking [12,13], bagging [14], and boosting, which use a generic meta-approach in predictive performance by integrating predictions from different models, improving the general prediction of DL. Stacking involves combining weak algorithms into a meta-model that can make better predictions [12,13].

There are two types of stacking ensembles: heterogeneous and homogeneous. The heterogeneous ensemble uses a variety of classifiers, while the homogeneous ensemble uses the same base model repeatedly. An ensemble of heterogeneous agents may perform better than an ensemble of homogeneous agents because of the combination of their biased decisions. In our work, we develop homogeneous stacking ensemble models to detect COVID-19.

Therefore, rapid diagnosis based on symptoms with accurate prediction is the essential AI-based solution to control the spread of the pandemic. Some related research work has been done on COVID-19 diagnosis. However, experimental research still needs to be done using ensemble learning for COVID-19 diagnosis. On the other hand, COVID-19 can induce pneumonia, which is caused by lung inflammation triggered by bacterial or viral infection. Consequently, researchers, specialists, and companies have used medical images (i.e., chest X-ray and computed tomography (CT)) for early diagnosis of COVID-19 patients. Hundreds of chest X-ray have been used to investigate the nature of pneumonia due to COVID-19 infection (see Figure 1). According to the context of this paper, DL models have been proposed to study chest X-ray images to benefit the detection of COVID-19. Researchers have used DL classifiers, in particular, to classify COVID-19 using chest X-rays images. For example, CNN models have been used to learn the pattern of COVID-19 infection from radiological X-ray images [15,16]. In particular, CNN models help to draw a clear distinction between non-tangible elements in the X-ray that can expose COVID-19 infection.

In our work, we develop homogeneous stacking ensemble models to detect COVID-19 based on symptoms and chest X-ray images. Our contributions can be summarized as follows:

- We propose two stacking ensemble DL models to detect COVID-19 using symptoms and chest X-ray images.
- The first proposed model is merged from the outputs of pre-trained DL models in the stacking: MLP, RNN, LSTM, and GRU; it uses stacking to train and evaluate the meta-learner (SVM) to identify the final prediction based on symptoms.
- The second proposed model is merged from the outputs of pre-trained models in the stacking: ResNet152V2, DenseNet201, VGG16, MobileNetV2, and inception_v3i; it uses stacking to train and evaluate the meta-learner (SVM) to identify the final prediction based on chest X-ray images.
- The first proposed model is evaluated against MLP, RNN, LSTM, and GRU using two COVID-19 symptom datasets and different assessment techniques: accuracy (A), recall (R), precision (P), and f1-score (F1).
- The second presented model is compared to ResNet152V2, DenseNet201, VGG16, MobileNetV2, and inceptionv3i utilizing COVID-19 chest X-ray images and different assessment techniques.
- A comparison of the proposed models with other models shows that the proposed models have the highest performance.

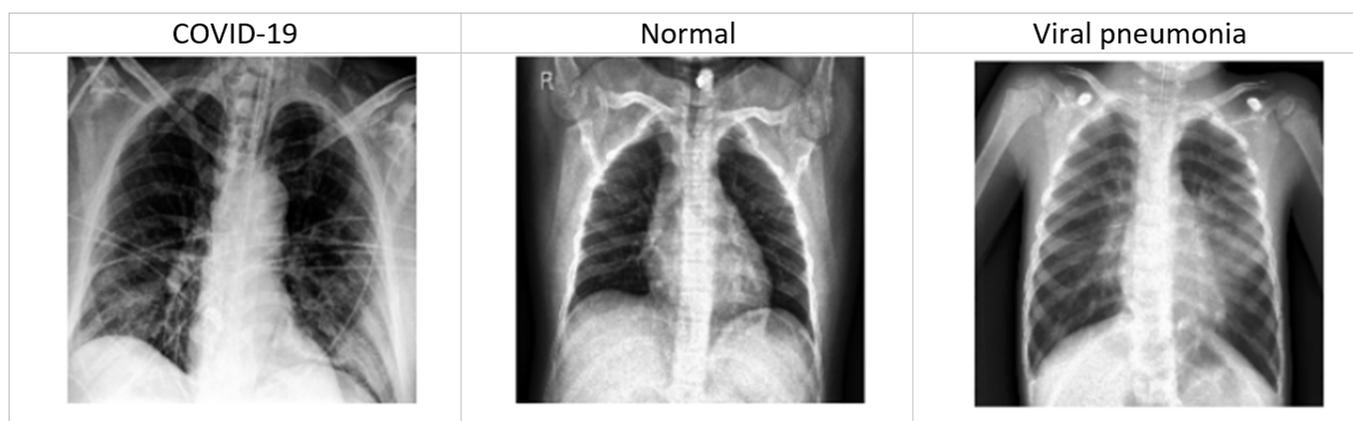


Figure 1. Example of chest X-ray images.

The study's structure is as follows: Section 2 discusses COVID-19 detection based on symptoms and chest X-ray images. The methodology and proposed models are discussed in Section 3. In Section 4, the experimental results are depicted. Conclusions are presented in Section 6.

2. Related Work

This section presents recent studies on the subject of detecting COVID-19 using symptoms and chest X-ray images.

2.1. Detecting COVID-19 Using Symptoms

The authors used ML and DL algorithms to detect COVID-19. For example, in [6], the authors used the gradient-boosting (GBoost) model for COVID-19 patient detection. They evaluated the model using AUC. In [17], the authors proposed a DL model technique called gray level co-occurrence matrix (GLCM) based on CNN. The authors in [18] contrasted widely employed feature extraction techniques for COVID-19 automatic categorization based on DL. The authors applied a group of deep CNNs, including InceptionV3, InceptionResNetV2, MobileNet, DenseNet, Xception, ResNet, VGGNet, and NASNet. In [19], the authors developed a predictive algorithm based on a trained DL model using 8427 COVID-19 patient records. In [20], the authors used the ML models: XBoost, AdaBoost, RF, and ExtraTrees with 337 COVID-19 patients. Jamshidi et al. [21] summarized

different models, including hybrid DL approaches and ML approaches, for calculating and forecasting complicated occurrences focused on the spread of COVID-19. In [22], the authors used ML techniques to detect mortality risks in COVID-19 using a dataset collected from the UK Biobank. The authors of [23] used KNN, SVM, LR, multilayer perceptual neural networks (MLP), LSTM, and GRU for COVID-19 diagnosis. They used the COVID-19 dataset from Kaggle [24] that includes some features and symptoms for their experiment. In [25], the authors used LR, NB, RF, DT, and gradient boosters for COVID-19 diagnosis based on some symptoms. They used the COVID-19 dataset from Kaggle [24] that includes some features and symptoms. The results showed that KNN achieved the highest accuracy. In [26], the authors Used RF, SVM, MLP and XGB, and LR to predict COVID-19 for children based on collected data that include some of the symptoms.

Previous studies used regular ML and DL models. However, they did not use ensemble stacking based on LSTM and GRU. In our study, we proposed stacking ensemble DL models for detecting COVID-19. The proposed model combined LSTM and GRU with SVM as a meta-learner for detecting COVID-19.

2.2. The Detection of COVID-19 Using Chest X-rays

Several studies have used transfer learning on chest X-ray images to identify COVID-19 patients. Here, we focus only on issues directly relevant to our suggestion.

In [27], X-ray images of the chest were analyzed using three pre-trained models for extracting features and detecting COVID-19. A variety of data augmentation techniques, such as random rotation and noise, were employed. VGG16 achieved the best results.

In [28], A total of 100 chest X-ray images was analyzed by the author to detect COVID-19 using three pre-trained CNNs, Inception-ResNetV2, InceptionV3, and ResNetV2. ResNet50 registered the highest result. In [29], the authors proposed CNN models (COVID-Net) and proposed a new design pattern called residual projection extension-projection extension (PEPX).

The authors of [30] proposed a concatenation-based CNN (Concat_CNN) model to detect COVID-19 from chest X-rays images. A comparison was made between Concat_CNN and the following transfer models: VGG16, InceptionV3, Resnet50, and DenseNet121. Concat_CNN registered the best results.

In [31], the authors suggested a CNN employing Softmax classifier and ML (SVM and RF).

In [32], the authors presented a hybrid CNN model using Xception and ResNet101 to extract COVID-19 characteristics from chest X-rays.

In [33], the authors proposed new ML models to detect COVID-19 from chest X-ray images. They used fractional multichannel exponent moments to extract features from images. In [34], the authors presented a DL model and employed SqueezeNet with a modified output layer to categorize X-ray pictures into COVID-19, normal, and pneumonia. In [35], the authors developed deep CNN (DCNN) to detect COVID-19 with five classes and compared it with eight pre-trained models. Based on the results, DCNN had the highest accuracy.

In [36], the authors used VGG16, VGG19, DenseNet201, Inception_ResNet_V2, Inception_V3, Resnet50, and MobileNet_V2 with five classes.

Table 1 provides an overview of previous studies on COVID-19 detection.

Table 1. Summary of previous studies on the detection of COVID-19.

Papers	Methods	Datasets
[6]	GBoost	51,831 tested individuals from Israeli Ministry of Health public dataset
[17]	GLCM based on CNN	Chest X-rays (273 X-rays, frontal view)
[18]	Different types of pretrained CNN	Kaggle's chest X-ray images (pneumonia)

Table 1. Cont.

Papers	Methods	Datasets
[20]	X_GBoost, AdaBoost, RF, and ExtraTrees	337 COVID-19-positive patients at Cheikh Zaid Hospital
[22]	Data-driven RF	11,245 participants in UK from Biobank dataset
[23]	KNN, SVM, LR, MLP, LSTM, and GRU	COVID-19-Symptoms-2
[25]	RF, SVM, MLP, and XGB	COVID-19-Symptoms-2

3. Materials and Methods

Our work aims to develop the proposed stacking ensemble models for detecting COVID-19 using two data types: chest X-ray images and some of the symptoms. This section describes the methodology and the framework of stacking ensemble models based on two data types.

3.1. Detecting COVID-19 Based on Symptoms

This subsection describes the proposed methods to detect COVID-19 based on symptoms, as shown in Figure 2. First, two symptoms included in the COVID-19 datasets are described. Second, the data splitting is presented. Third, the DL model architecture, including MLP, RNN, LSTM, and GRU, and optimization methods for DL models are presented. Finally, we discuss how the DL pre-trained models were combined using stacking ensemble learning techniques.

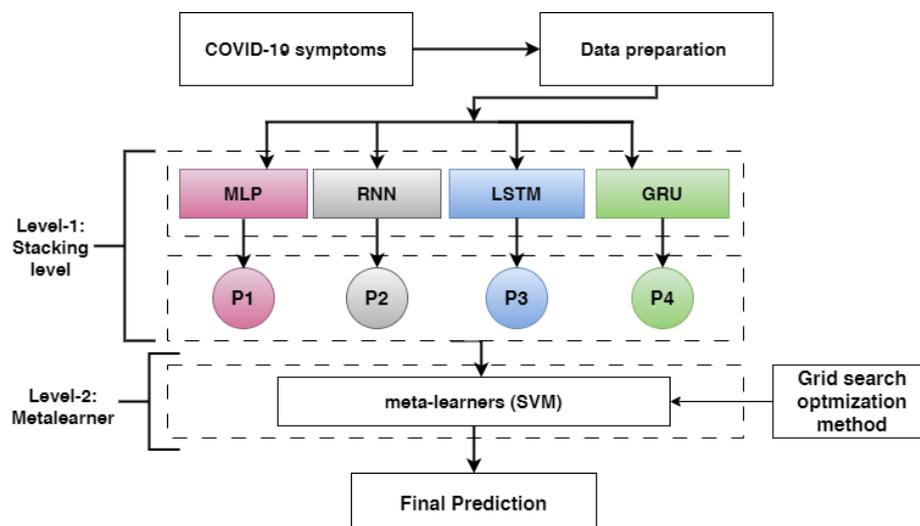


Figure 2. The proposed method for detecting COVID-19 based on symptoms (P1, P2, P3, and P4 refer to the probability outputs of each model).

3.1.1. COVID-19 Symptoms Dataset Description

Two datasets of COVID-19 symptoms are used to conduct our experiment.

- The first dataset of symptoms of COVID-19 (COVID-19-Symptoms-1) is downloaded from GitHub [37], and it includes 13 features and one class label. The class label has 755 recovered (0) and 108 deaths (1); 250 rows for class 0 are selected. Based on some pre-defined standard symptoms, the data will help determine whether a person will recover from COVID-19 symptoms. WHO guidelines are used to determine these symptoms. An explanation of the features of COVID-19-Symptoms-1 is described in Table 2.

Table 2. Description of the features of COVID-19-Symptoms-1.

Features	Descriptions
location	What region of the country
country	The place where the person lives
gender	Male or female
age	Age of patient
vis_wuhan	Indication of whether the person has visited Wuhan
from_wuhan	Whether the person is from Wuhan, China, or not
symptom1, symptom2, symptom3, symptom4, symptom5 and symptom6	Six features of symptoms
diff_sym_hos	Time before symptoms appear
result	Recovered or death

Figure 3 shows the correlation matrix of the COVID-19-Symptoms dataset. We can see that the symptoms are highly correlated with each other. Age and diff_sym have the highest correlation with the results.

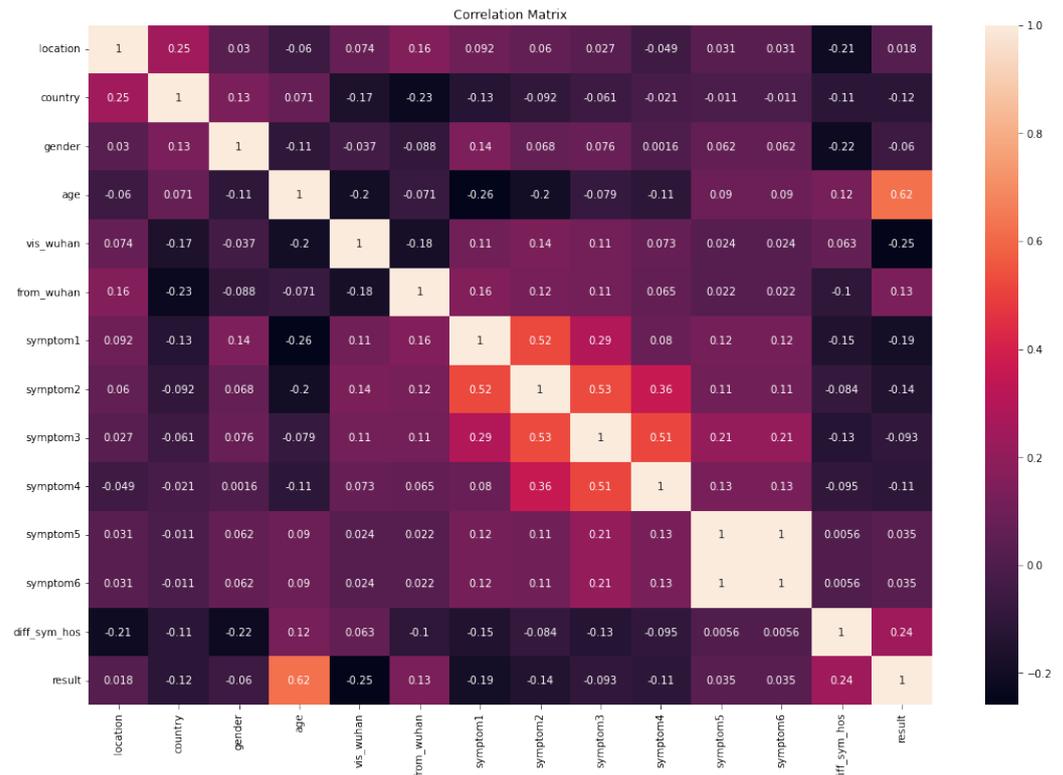


Figure 3. Correlation matrix of COVID-19-Symptoms-1.

- The second dataset of COVID-19 symptoms (COVID-19-Symptoms-2) [24] covers the presence of several features (mask use, trip overseas, and interaction with a COVID patient), as well as multiple symptoms (fever, dry cough, and breathing issues); in addition, the class label refers to whether the person has COVID or not. There are 4347 rows for the training set, and 1087 for the testing set. A description of the features of COVID-19-Symptoms-2 is shown in Table 3.

Figure 4 shows the correlation matrix of the COVID-19-Symptoms dataset2. We can see that the WM and SF features have one value; therefore, they are removed from the dataset. In addition, we transformed categorical features into numerical data using LabelEncoder in Python.



Figure 4. Correlation matrix of COVID-19-Symptoms-2.

Table 3. Description of the features of COVID-19-Symptoms-2.

Features	Description
BP	Difficulty in breathing due to breathing problems Ranges from mild, to moderate, to severe
F	Increase in patient temperature (commonly over 38)
DC	Type of cough that usually does not produce any phlegm
ST	A common symptom that is mainly characterized by pain or itchiness in the throat
RN	Discharge of fluid due to viral or bacterial causes
AS	Chronic respiratory diseases usually narrow the airway path and cause breathing problems
CLD	A medical condition that causes long-term problems with breathing
HD	Felling pain or discomfort in the face region, ranges from mild, to moderate, to severe
Heart	A medical condition that affects blood vessels and heart status
DI	Chronic diseases in which the patient becomes unable to produce sugar at a regular level due to pancreas problems
HT	Chronic diseases in which the force of the blood against the direction of the walls is higher than normal
FA	The feeling of pain or illness due to extreme effort or tiredness

Table 3. *Cont.*

Features	Description
GA	This refers to the digestive system where all processing and absorbing of food occurs
AL	Check if the patient traveled in the last 14 days
CW	Direct connection with positive cases of COVID-19
AL	Check if the patient has attended recent gatherings (i.e., festival, party)
VP	Check if the patient has visited an exposed place
FW	If any one of the patient's family works in one of the exposed places
WM	Check if the patient's mask wearing continues when outside
SF	Check if the patient visits a place for sanitization
COVID-19	The final decision (yes for positive, no for negative)

3.1.2. Data Splitting

Datasets are divided 80/20 into training and testing sets. A training set was used for training and optimizing models, and a testing set was used for evaluating models. Additionally, 10% of training sets are used as validation sets.

Table 4 presents the number of rows for each class in COVID-19-Symptoms-1 and COVID-19-Symptoms-2, respectively.

Table 4. The number of rows in COVID-19 datasets.

Datasets	Classes	Training Set	Testing Set
COVID-19-Symptoms-1	Recovered	198	52
	Death	88	20
	Total	286	72
COVID-19-Symptoms-2	Yes	3501	882
	No	846	205
	Total	4347	1087

3.1.3. DL Models

The MLP, RNN, LSTM, and GRU are trained and evaluated in accordance with our objective datasets. The final layer of each model includes three neurons and a softmax function; the loss function is categorical cross-entropy, and the optimizer is Adam [38].

- A multilayer perceptron (MLP) is a neural network that complements forward neural networks. It has three layers: input, output, and hidden. The input layer receives input signals [39].
- Recurrent neural networks (RNN) keep a state vector in their hidden units that indirectly provides information about the history of all previous items in an input sequence [40]. A basic RNN contains three layers: input, recurrent hidden, and output. N input units are present in the input layer. This layer's inputs are a series of vectors traversing time t [41]. The input units in the hidden layer are fully linked to the hidden units in the hidden layer, with the connections determined by a weight matrix. The hidden layer includes M hidden units, which are linked together in time via recurrent connections [42].
- Long short-term memory (LSTM) architecture is applied to DL algorithms as an attention-based RNN. LSTMs have feedback connections. A complete data sequence can be analyzed, as well as single data points. In LSTM mode, one of the most crucial components is the "cell state" of the memory cell, which maintains its state over time [43].

- Gated recurrent units (GRUs), a type of RNN, use gate units to control information flow rather than separate memory cells. GRUs contain two gate operating mechanisms to solve the challenge posed by standard RNNs: an update gate and a reset gate [44,45]. The update gate ensures that the necessary memory is retained in order to go to the next stage. In order to advance to the next stage, the update gate ensures that enough memory is retained. The reset gate controls how previously stored data are updated with a new input. After the reset gate engages, a newer memory content is created for the details of the preceding time step [46].

3.1.4. Optimization Methods

Hyperparameter tuning is the process of adapting hyperparameters to obtain the right set of values that optimizes the performance of a DL. A hyperparameter is a variable that determines the training process and model topology for DL models. These variables directly impact DL performance throughout the training process. KerasTuner [47] is a Python library explicitly developed for tuning DL hyperparameters. KerasTuner supports different types of algorithms, namely Bayesian optimization, hyperband, Sklearn, and random search [47]. Some hyperparameters are adapted, such as the number of units (ranging between 20 and 800) and the width of hidden layers.

3.1.5. The Proposed Model

Stacking is combining the different models' output with training other models to produce the best result. Heterogeneous stacked ensemble is a strategy for blending many heterogeneous models by learning by meta-learner to predict the final results [48]. The idea behind stacking is that some models will fit the categories of a test observation properly while others will not [49]. The algorithm learns from the variety of predictions and seeks to integrate the models to improve the performance of the basic models [50].

Two levels are proposed in our model: level-1 and level-2.

- In level-1, each base-learner (MLP, RNN, LSTM, and GRU) is trained separately and saved. Then, the pre-trained models (RNN, LSTM, and GRU) are loaded, and all layers are frozen without the output layer. Each model takes a training set and predicts the training output of a probability (p1, p2, and p3). Then, the training outputs are combined in stacking, which is called training stacking.
- In level-2, the meta-learner (SVM) is trained and optimized using training stacking. The meta-learner (SVM) is evaluated and tested using testing stacking to predict the final results. The meta-learner is optimized using a grid search with different parameter values.

3.2. Detecting COVID-19 Based on the Dataset of Chest X-ray Images

This subsection describes the proposed methods to detect COVID-19 based on symptoms. First, chest X-ray images are described. Second, the data preparation procedure involving data augmentation and image resizing is presented. Third, the pre-trained models ResNet152V2, DenseNet201, VGG16, MobileNetV2, and inception_v3i are presented. Finally, we discuss how the pre-trained models were combined using stacking ensemble learning techniques. Figure 5 shows the proposed methodology's overall workflow in detail.

3.2.1. COVID-19 Chest X-ray Images Description

- COVID-19 -chest-X-ray-1

Kaggle provided 317 chest X-ray images [51] in three classes: 137 images with COVID-19, 90 images with normal imaging, and 90 images with viral pneumonia. A total of 251 images are available for training and 66 images are available for testing.

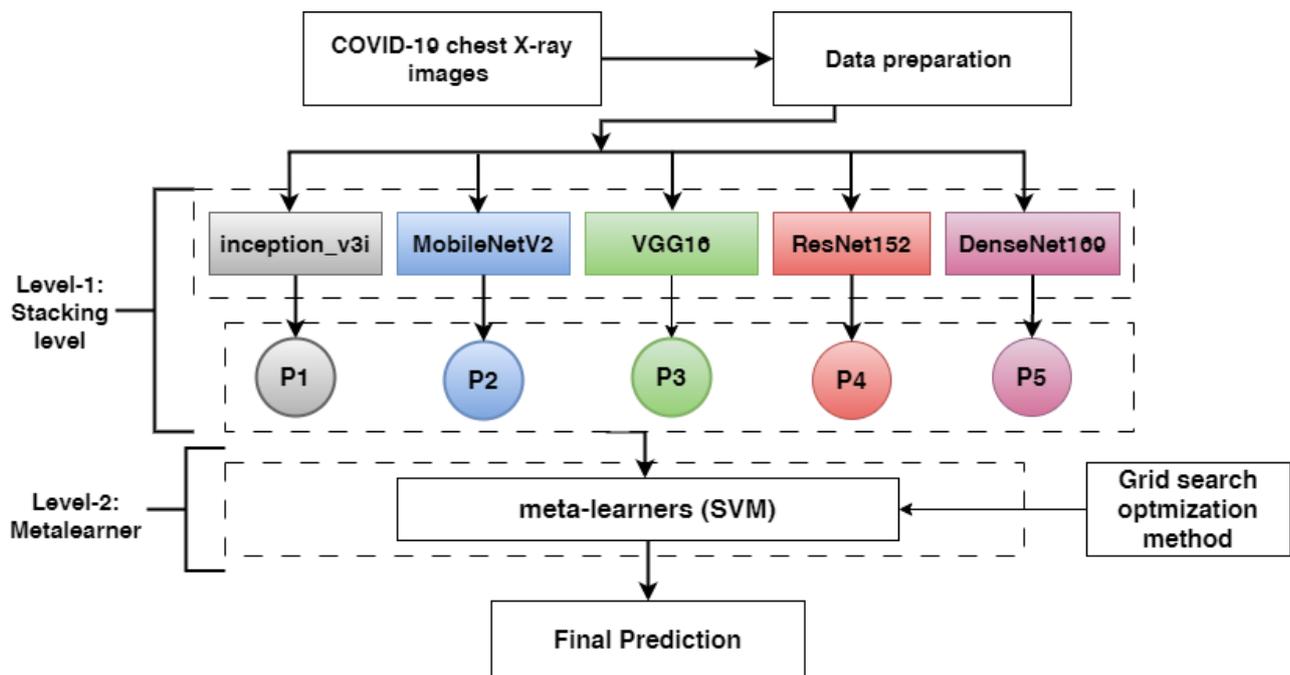


Figure 5. The proposed method for detecting COVID-19 based on chest X-ray images (P1, P2, P3, P4, and P5 refer to the probability outputs of each model).

- COVID-19-chest-X-ray-2

A total of 2060 CHX-Ray images were downloaded from Kaggle [52]. Of these, 696 images were selected for testing and 2060 for training.

3.2.2. Data Augmentation

Preprocessing the first X-ray chest images is required to enhance image features and improve image data quality. First, RGB is modified for the image channel sequence. Second, these images are resized to $224 \times 224 \times 3$. Third, image augmentation is performed, which is a method of producing additional dataset points from existing data by developing changed copies of a dataset [53–55]. A variety of augmentation strategies are applied: rescale:1./255, zoom_range:0.1, rotation_range:20, width_shift_range:0.1, height_shift_range:0.1, and horizontal_flip:True.

3.2.3. Fine-Tuning the Pre-Trained

The pre-trained ResNet152V2, DenseNet201, VGG16, MobileNetV2, and inception_v3i are picked and fine-tuned in accordance with our objective datasets. The final layer of each model includes three neurons and a softmax function; the loss function is categorical cross-entropy, and the optimizer is Adam.

- Visual geometry group (VGG): In a convolutional neural network architecture, Zisserman and Simonyan proposed VGG in 2014 [56]. The essential part of this architecture is that rather than depending on a huge number of hyperparameters, it concentrates on fundamental size kernels in the convolutional layers and kernels in the max-pooling layers. In the end, there are two fully connected layers, followed by a softmax for output [57,58]. VGG19 differs from VGG16 in that it contains an extra layer in the three convolutional blocks [59].
- Densely connected convolutional networks (DenseNet): The dense convolutional network recognizes the input image size, which uses dense connections across layers with dense blocks. The network spans 201 layers of depth while connecting all layers directly with each other with feed-forward using matching feature-map sizes [60]. Each layer receives extra inputs from all previous levels and relays its feature maps

to all previous layers to maintain the system's feed-forward structure. Compared to conventional networks, DenseNet can outperform ordinary networks by increasing processing needs, reducing parameter count, increasing feature reuse, and maintaining feature propagation [61].

- Deep residual networks (ResNet) employ residual blocks to increase model accuracy for image classification. The skip connections are crucial to the residual blocks and the strength of this form of neural network [62]. One residual block consists of a convolution layer preceded by a batch normalization layer that adjusts to retain a mean outcome closer to 0. The output standard deviation is near one, and a ReLU activation function is used. This is followed by a convolution layer and a batch normalizing layer [63]. The skip connection bypasses both levels and is added immediately before the ReLU activation function. Such residual blocks are repeated to construct a residual network. ResNet comes in a variety of forms that all follow the same basic idea but employ different numbers of layers [64]. It has five stages, each with a convolution and identity block, and each convolution and identity block has three convolution layers [65].
- The inception network, a significant landmark in the creation of CNN classifiers, incorporates a block of parallel convolutional layers with three distinct filter sizes [66]. In addition, max pooling is conducted. Because of the varying filter sizes, the network has the ability to learn multiple variabilities at different scales using convolutions [67]. Concatenated results are forwarded to the following inception module [68]. The max-pooling layer in an inception module may benefit from padding to keep its height and breadth consistent with the other outputs (feature maps) of the convolutional layers in the same inception module [69].
- Xception is a 71-layer deep convolutional neural network that has an input image size of 299 upon swapping the normal inception modules with depthwise separable convolutions [66,70]. Depthwise separable convolution layers are based on the principle that convolutional neural network feature maps resulting from such cross-channel and spatial correlation translation could be entirely independent [71].
- MobileNet is a simplified design that employs depthwise separable convolutions created by mixing two 1D convolutions with two kernels to generate lightweight deep convolutional neural networks [72]. This means that less memory and fewer parameters are required for training, resulting in a more efficient model for mobile and embedded vision applications [73].

3.2.4. The Proposed Model

Two levels are proposed in our model: level-1 and level-2.

- In level-1, each base-learner (ResNet152V2, DenseNet201, VGG16, MobileNetV2, and inception_v3i) is trained separately and saved. Then, the pre-trained models (ResNet152V2, DenseNet201, VGG16, MobileNetV2, and inception_v3i) are loaded, and all layers are frozen without the output layer. Each model takes a training set and predicts a training probability output (p1, p2, p3, p4, and p5). Then, the training outputs are combined in stacking, which is called training stacking.
- In level-2, the meta-learner (SVM) is trained and optimized using training stacking. The meta-learner (SVM) is evaluated and tested using testing stacking to predict the final results. The meta-learner is optimized using a grid search with different parameter values.

4. Experiments Results

This section describes the results of testing DL models and the proposed models using two COVID-19 symptom datasets and two chest X-ray image datasets to detect COVID-19.

4.1. Experiment Setup

The experiments were conducted with Python using Google Colab. The Scikit-learn package was used for ML, while the Keras library was used for DL.

4.2. Evaluation

The evaluation metrics were applied to assess the learning algorithms. The following four metrics were used to assess classification performance: accuracy (A), precision (P), recall (R), and F1-score (F1).

- Accuracy is a popular evaluation parameter for classification problems. It is the proportion of correct forecasts relative to total predictions [74].

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

- Precision is a measure for determining categorization accuracy. The equation represents the proportion of correct positive classifications relative to total anticipated positive classifications [74].

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

- Recall is the number of accurately detected positive cases out of the total number of positive cases. Returning to the fraud issue, the recall value will be quite valuable in fraud scenarios. A high recall value indicates that a significant number of fraud cases are recognized in comparison to the total number of frauds [74].

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

- The F1-score measures the mean of the model's precision and recall [74].

$$F1\text{-score} = \frac{2 \cdot precision \cdot recall}{precision + recall} \quad (4)$$

True positive (TP), true negative (TN), false positive (FP), and false negative (FN) values were recorded. A TP indicates the set of correctly formed positive values, a FP indicates the number of negative values generated incorrectly, a TN indicates the number of negative values generated correctly, and a FN indicates the number of positively predicted values that were correctly created.

4.3. Results COVID-19 Symptoms Datasets

This section explores the ability of our proposed model to detect COVID-19 based on symptom datasets.

4.3.1. Parameters Configuration

A training set is used to optimize and train DL models. Some parameters were adopted in RNN, LSTM, and GRU to conduct experiments, such as `batch_size = 200` and `epoch = 50` with a learning rate of 0.0001, and Adam optimizer. In addition, we used KerasTuner to optimize some parameters in RNN, LSTM, and GRU. The final values of parameters for each model are shown in Table 5.

Table 5. In order to detect COVID-19 based on symptoms, the best parameter values were selected for RNN, LSTM, and GRU.

Datasets	Models	Number of Units
COVID-19-Symptoms-1	RNN-Layer1	[490]
	RNN-Layer2	[690,450]
	LSTM-Layer1	650
	LSTM-Layer2	[350,370]
	GRU-Layer1	530
	GRU-Layer1	[530,310]
COVID-19-Symptoms-2	RNN-Layer1	[570]
	RNN-Layer2	[430,330]
	LSTM-Layer1	[230]
	LSTM-Layer2	[570,350]
	GRU-Layer1	[250]
	GRU-Layer1	[610,570]

4.3.2. COVID-19-Symptoms-1

Table 6 shows the results of DL models, as well as the proposed model, using COVID-19-Symptoms-1. We can observe that Proposed-Layer2 achieved the highest performance compared to other models.

Regarding DL models with one layer, Among the four metrics, RNN-Layer1 scored the lowest: 84.72, 84.31, 84.72, and 83.98, respectively. According to the evaluation metrics, the second best results were obtained by MPL-Layer1: 93.06, 93.66, 93.06, and 92.72.

Regarding DL models with two layers, RNN-Layer2 recorded the lowest results in several metrics: 87.5, 87.43, 87.5, and 86.89 in terms of A, P, R, and f1, respectively. The second-best results were obtained from MLP-Layer2 according to different evaluation metrics: 94.44, 94.77, 94.44, and 94.52 in terms of A, P, R, and f1, respectively. Proposed-Layer2 improved performance in different metrics: A by 2.45, P by 2.38, R by 2.45, and f1 by 2.54 compared to MLP-Layer2.

Table 6. The A, P, R, and F1 of applying models to COVID-19-Symptoms-1.

Models	Models	Accuracy	Precision	Recall	F1-Score
DL models	MPL-Layer1	93.06	93.66	93.06	92.72
	MPL-Layer2	95.83	96.06	95.83	95.72
	RNN-Layer1	84.72	84.31	84.72	83.98
	RNN-Layer2	87.5	87.43	87.5	86.89
	LSTM-Layer1	91.67	91.56	91.67	91.53
	LSTM-Layer2	94.44	94.77	94.44	94.52
	GRU-Layer1	90.28	90.55	90.28	89.81
	GRU-Layer2	90.28	90.17	90.28	90.02
The proposed model	Proposed-Layer1	96.83	96.81	96.83	96.80
	Proposed-Layer2	98.28	98.44	98.28	98.26

4.3.3. COVID-19-Symptoms-2

Table 7 shows the results of DL models, as well as the proposed model, using COVID-19-Symptoms-2. We can observe that Proposed-Layer2 achieves the highest performance compared to other models.

Regarding DL models with one layer, LSTM-Layer1 recorded the lowest results in several metrics: 94.44, 94.77, 94.44, and 94.52. The second-best results were obtained from MPL-Layer1 according to different evaluation metrics: 97.52, 97.56, 97.52, and 97.53. Proposed-Layer1 improved performance by several metrics: A by 0.78, P by 0.76, R by 0.78, and f1 by 0.77 compared to MPL-Layer1.

Regarding DL models with two layers, LSTM-Layer2 records the lowest results in the different metrics: 96.87, 96.84, 96.87, and 96.85. The second-best results were obtained from MPL-Layer2 according to different evaluation metrics: 98.10, 98.10, 98.10, and 98.10. Proposed-Layer2 improved performance in several metrics: A by 1.20, P by 1.20, R by 1.20, and f1 by 1.21 compared to MPL-Layer2.

Table 7. The A, P, R, and F1 of applying models to COVID-19-Symptoms-2.

Approaches	Models	A	P	R	F1
DL models	MPL-Layer1	97.52	97.56	97.52	97.53
	MPL-Layer2	98.10	98.10	98.10	98.10
	RNN-Layer1	97.33	97.31	97.33	97.31
	RNN-Layer2	97.61	97.62	97.61	97.61
	LSTM-Layer1	94.44	94.77	94.44	94.52
	LSTM-Layer2	96.87	96.84	96.87	96.85
	GRU-Layer1	96.6	96.55	96.6	96.55
	GRU-Layer2	97.24	97.25	97.24	97.25
The proposed model	Proposed-Layer1	98.30	98.32	98.30	98.30
	Proposed-Layer2	99.30	99.30	99.30	99.31

4.4. Results of Chest X-ray Images Datasets

This section explores the ability of our proposed model to detect COVID-19 based on chest X-ray datasets.

4.4.1. Parameters Configuration

For training ResNet152V2, DenseNet201, VGG16, MobileNetV2, and inception_v3i, some parameters were adopted to conduct experiments, such as batch_size = 64 and epoch = 100 with a learning rate of 0.001 and Adam optimizer. The activation function is softmax, and the loss function is categorical cross-entropy.

4.4.2. COVID-19-Chest-X-ray-1

Table 8 shows the results of models, including the proposed model, using COVID-19-chest-X-ray-1. Comparing the proposed model to other models, it was the most efficient. The proposed model improved performance in several metrics: A by 1.38, P by 1.4, R by 1.38, and F1 by 1.38 compared to ResNet152V2 and VGG16.

ResNet152V2 and VGG16 recorded similar performance in terms of different metrics: 98.24, 98.24, 98.26, and 98.24. The third-highest results were obtained from MobileNetV2 according to evaluation metrics: 96.97, 96.97, 96.97, and 96.97. Finally, inception_v3i recorded the lowest scores: 93.94, 94.08, 93.94, and 93.98.

Table 8. The A, P, R, and F1 of applying models to COVID-19-chest-X-ray-1.

Models	A	P	R	F1
ResNet152V2	98.24	98.26	98.24	98.24
DenseNet201	96	96	96	96
VGG16	98.24	98.26	98.24	98.24
MobileNetV2	96.97	96.97	96.97	96.97
inception_v3i	93.94	94.08	93.94	93.98
The proposed model	99.62	99.66	99.62	99.62

4.4.3. COVID-19-Chest-X-ray-2

Table 9 shows the results of models, including the proposed model, using COVID-19-chest-X-ray-2. Comparing the proposed model to other models, it was the most efficient. The proposed model improves performance in different metrics: A by 2.22, P by 2.29, R by 2.22, and F1 by 2.21 compared to MobileNetV2.

MobileNetV2 recorded the second-highest performance in different metrics: 96.26, 96.27, 96.26, and 96.27 in terms of A, P, R, and F1, respectively. The third-highest model was obtained from VGG16 according to the evaluation metrics: 95.55, 95.56, 95.55, and 95.55. Finally, inception_v3i recorded the lowest scores: 93.25, 93.25, 93.25, and 93.24.

Table 9. The A, P, R, and F1 of applying models to COVID-19-chest-X-ray-2.

Models	A	P	R	F1
ResNet152V2	95.11	95.13	95.11	95.11
DenseNet201	94.97	94.98	94.97	94.97
VGG16	95.55	95.56	95.55	95.55
MobileNetV2	96.26	96.27	96.26	96.27
inception_v3i	93.25	93.25	93.25	93.24
The proposed model	98.48	98.56	98.48	98.48

5. Discussion

This section presents the best models of the COVID-19 symptoms dataset and COVID-19 chest X-ray images. It also shows a comparison between the proposed model and recent studies.

5.1. COVID-19 Symptoms Datasets

5.1.1. The Best Models for COVID-19 Symptoms Datasets

A stacking ensemble model was proposed to detect COVID-19 sickness by combining the pre-trained models MLP, RNN, LSTM and GRU with the meta-learner model SVM. The proposed model achieved the highest performance in the two datasets compared to other models.

Figure 6 presents the best models for COVID-19-Symptoms-1. We can see that Proposed-Layer2 recorded the highest scores using different matrices: A = 98.28, P = 98.44, R = 98.28, and F1 = 98.26. MLP-Layer2 recorded the second-highest scores using different matrices: A = 95.83, P = 96.06, R = 95.83, and F1 = 95.72. RNN-Layer2 recorded the lowest A, P, R, and F1 at 87.5, 87.43, 87.5, and 86.89, respectively.

Figure 7 presents the best models for COVID-19-Symptoms-2. We can see that Proposed-Layer2 recorded the highest scores using different matrices: A = 99.30, P = 99.30, R = 99.30, and F1 = 99.31. MLP-Layer2 recorded the second-highest scores using different matrices: A = 98.10, P = 98.10, R = 98.10, and F1 = 98.10. LSTM-Layer2 recorded the lowest A, P, R, and F1 at 96.87, 96.84, 96.87, and 96.85, respectively.

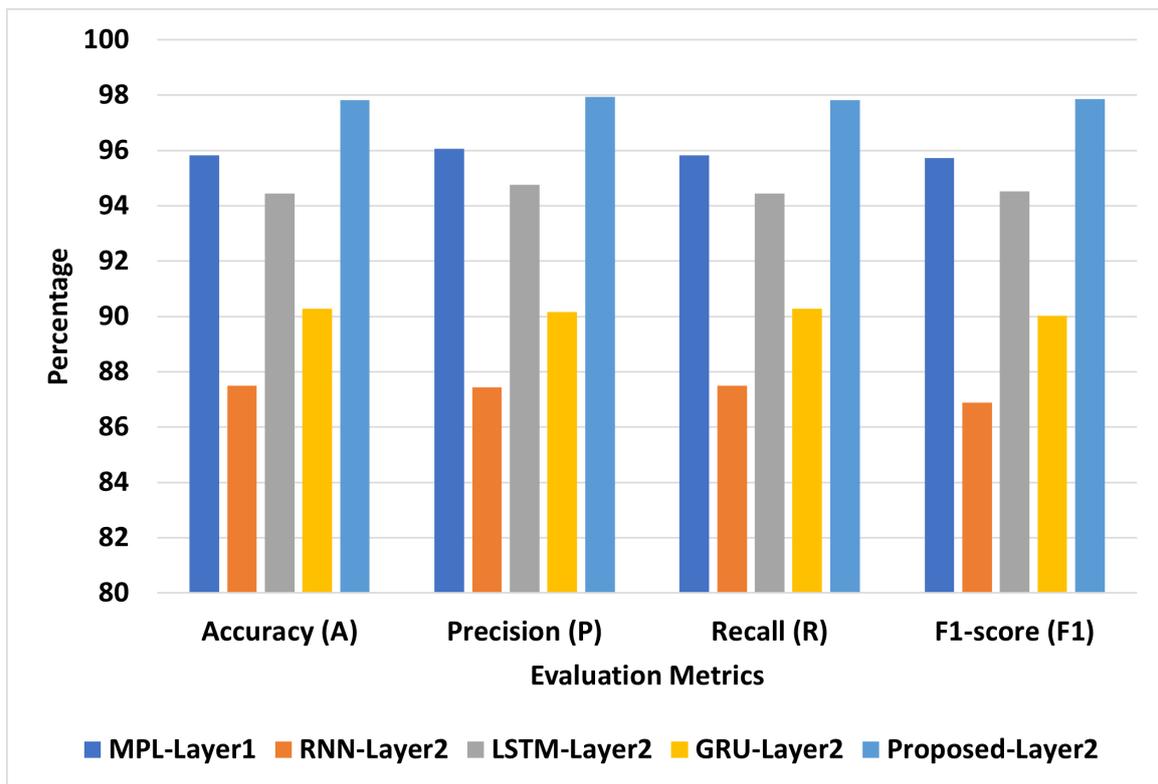


Figure 6. The best models for COVID-19-Symptoms-1.

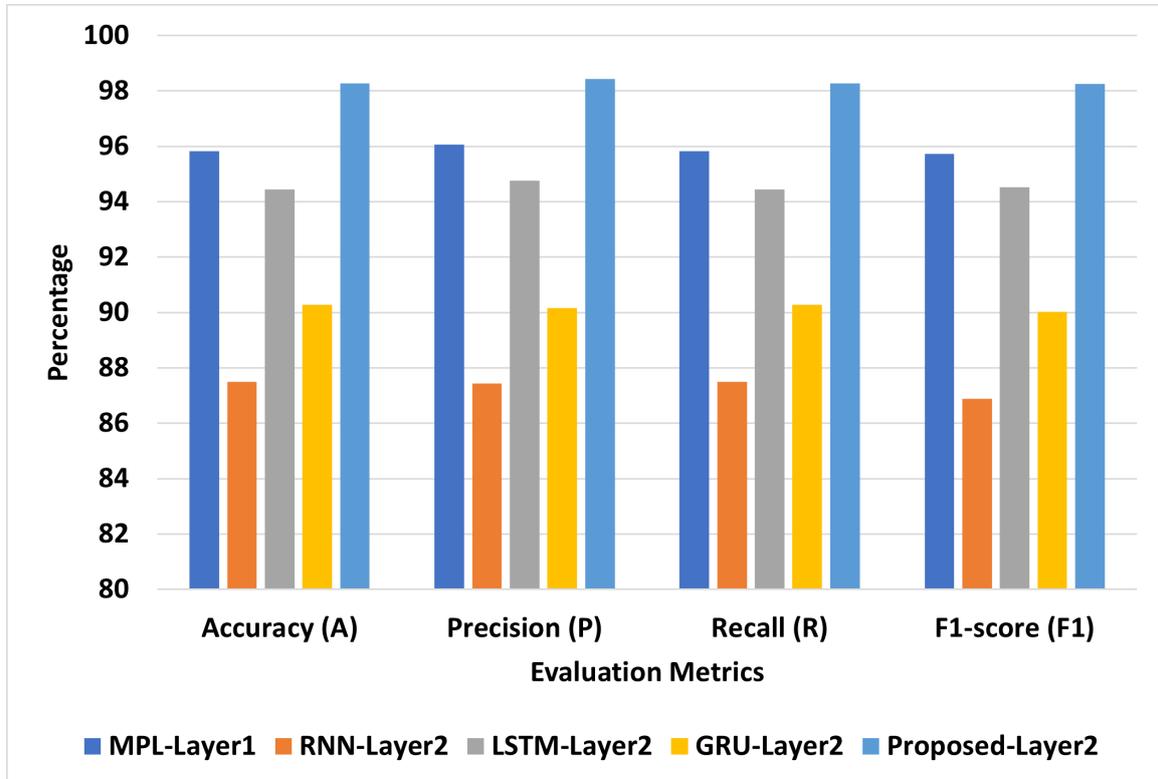


Figure 7. The best models for COVID-19-Symptoms-2.

5.1.2. Comparison with Literature Studies for COVID-19 Symptoms Dataset

Table 10 shows a comparison of previous studies that used COVID-19-Symptoms-2 [24] with the proposed models. In [23], the authors used GRU, which recorded A = 98.65, R = 98.6, P = 99.2, and F1 = 99.2. In addition, in [25], the authors used KNN, which registered A = 97.97, R = 97.97, P = 97.97, and F1 = 97.97. The proposed model achieved the highest performance compared to [23,25].

Table 10. Comparing previous studies with the proposed model using COVID-19-Symptoms-2.

Papers	The Best Models	Performance
[23]	GRU	A = 98.65, R = 98.6, P = 99.2, and F1 = 99.2
[25]	KNN	A = 97.97, R = 97.97, P = 97.97, and F1 = 97.97
The proposed model	Stacking SVM	A = 99.30, R = 99.30, P = 99.30, and F1 = 99.31

5.2. COVID-19 Chest X-ray Images Datasets

A stacking ensemble model was proposed to detect COVID-19 sickness by combining the pre-trained models ResNet152V2, DenseNet201, VGG16, MobileNetV2, and inception_v3i with the meta-learner model SVM. The proposed model achieved the highest performance in the two datasets compared to other models.

5.2.1. The Best Models for Chest X-ray Image Datasets

This section presents the best models used with chest X-ray image datasets.

Figure 8 presents the best models for COVID-19-chest-X-ray-1. We can see that the proposed model recorded the highest scores using different matrices: A = 99.62, P = 99.66, R = 99.62, and F1 = 99.62. VGG16 and ResNet152V2 recorded the second-highest scores using different matrices: A = 98.24, P = 98.26, R = 98.24, and F1 = 98.24.

Figure 9 represents the best models for COVID-19-chest-X-ray-2. We can see that the proposed model recorded the highest scores using different matrices: A = 98.48, P = 98.56, R = 98.48, and F1 = 98.48. MobileNetV2 record the second-highest scores using different matrices: A = 96.26, P = 96.27, R = 96.26, and F1 = 96.27.

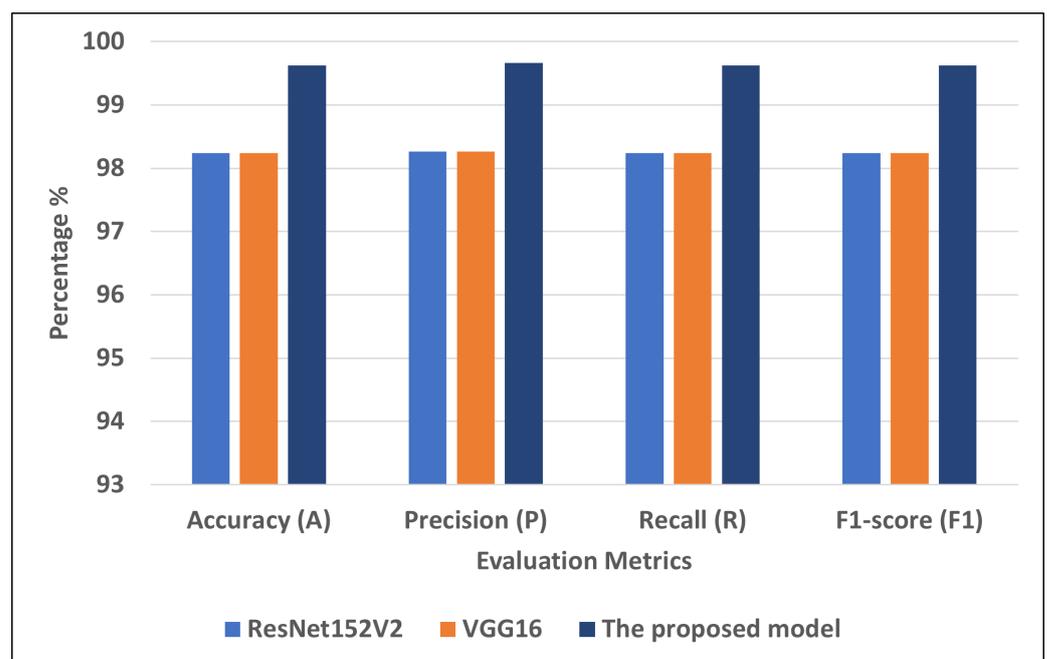


Figure 8. The best models for COVID-19-chest-X-ray-1.

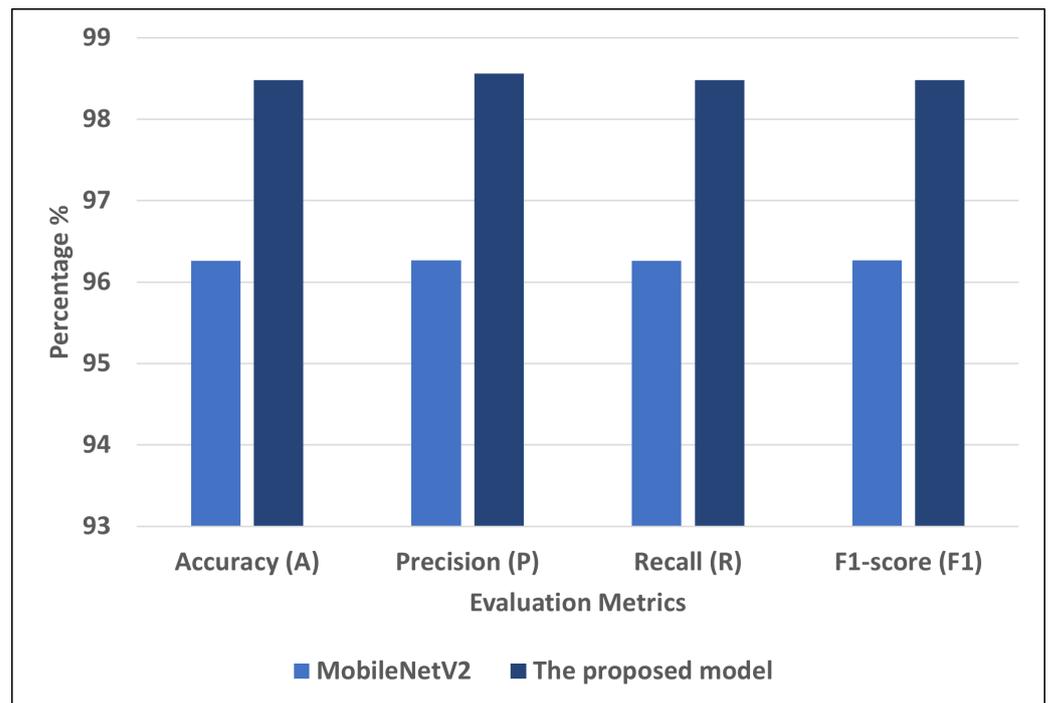


Figure 9. The best models for COVID-19-chest-X-ray-2.

5.2.2. Comparison with Literature Studies

Table 11 shows a comparison between the proposed model and recent studies using COVID-19-chest-X-ray-2 with two or three classes. We can see that the proposed model achieved the highest performance. The authors detected COVID-19 results in three classes: COVID-19, normal, and pneumonia. In [28], ResNet50 recorded A = 98 and R = 96.46. In [29], the authors indicated that COVID-Net has a recorded accuracy at 92.4 A. In [30], the authors proposed Concat_CNN, which recorded A = 96.31, P = 95.8, and R = 92.99. In [32], a concatenated CNN model was proposed and recorded A = 98.02, F1 = 98.24, P = 97.04, and R = y. The authors of [34] used SqueezeNet, with recorded accuracy of A = 95, P = 94.66, R = 94.66, and F1. In [3], XGBoost recorded A = 97.87, P = 97.87, and R.

Table 11. The proposed model is compared to recent studies using COVID-19-chest-X-ray-2.

Papers	Models	Image Classes	Performance
[28]	ResNet50	Normal COVID-19 pneumonia	A = 98 and R = 96.46
[29]	COVID-Net	Normal COVID-19 pneumonia	A = 92.4
[30]	Concat_CNN	Normal COVID-19 pneumonia	A = 96.31, P = 95.8, and R = 92.99
[32]	Concatenated CNN model	Normal COVID-19 pneumonia	A = 98.02, F1 = 98.24, P = 97.04, R = 98.49

Table 11. Cont.

Papers	Models	Image Classes	Performance
[34]	SqueezeNet	Normal COVID-19 pneumonia	A = 95, P = 94.66, R = 94.66, and F1 = 94.66
[3]	XGBoost	Normal COVID-19 pneumonia	A = 97.87, P = 97.87, and R = 98.93
The proposed model	Stacking ensemble DL	Normal COVID-19 pneumonia	A = 99.62, P = 99.66, R = 99.62, and F1 = 99.62
The proposed model	Stacking ensemble DL	Normal COVID-19 pneumonia	A = 98.48, P = 98.56, R = 98.48, and F1 = 98.48

6. Conclusions

This paper proposes a stacking ensemble DL model using COVID-19 symptoms and chest X-ray images to detect the disease. Two models have been proposed for use with the different datasets, including one based on COVID-19 symptoms and one based on chest X-ray images. The first proposed model combines four pre-trained deep learning models, MLP, RNN, LSTM, and GRU, together into a stacking so that a meta-learner is trained and evaluated to identify a final prediction. In comparison to DL models based on two COVID-19 symptom datasets, our proposed model achieved the highest performance (A = 99.30, P = 99.30, R = 99.30, and F1 = 99.31). The second proposed model has merged the outputs of the pre-trained models ResNet152V2, DenseNet201, VGG16, MobileNetV2, and inception_v3i in a stacking and uses stacking to train and evaluate the meta-learner (SVM) to identify the final prediction using chest X-ray datasets. Comparing the proposed model to DL models based on the two COVID-19 chest X-ray datasets, it achieved the best performance (A = 99.62, P = 99.66, R = 99.62, and F1 = 99.62). Our proposed models were applied to two different types of datasets, COVID-19 symptoms and chest X-ray images, and it achieved the highest performance in measuring the generalizability of the proposed model. However, our model needs some enhancements, which will be considered in future work, including (1) testing the model on other datasets and (2) applying explainable AI (XAI).

Author Contributions: Methodology, H.S., A.A., R.M.A.S. and S.M.; Software, H.S.; Validation, H.S.; Data curation, A.A.; Writing—original draft, H.S., A.A., R.M.A.S., N.E.-R., S.E.-S. and S.M.; Writing—review & editing, A.A., N.E.-R., R.M.A.S., S.E.-S. and S.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All datasets used to support the findings of this study are available from the direct link in the dataset citations.

Acknowledgments: This research project was supported by a grant from Midocean University.

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

Abbreviations

The following abbreviations are used in this manuscript:

Artificial intelligence	AI
Machine learning	ML
Deep learning	DL
Convolutional neural network	CNN
Gated recurrent unit	GRU
Long short-term memory	LSTM
Recurrent neural network	RNN
Support vector machine	SVM
k-nearest neighbors	KNN
Random forest	RF
Multilayer perceptron	MLP
XGBoost	XGB
Naive Bayes	NB
Decision tree	DT
LSTM-Layer1	One-layer LSTM
LSTM-Layer2	Two-layer LSTM
GRU-Layer1	One-layer GRU
GRU-Layer2	Two-layer GRU
RNN-Layer1	One-layer RNN
RNN-Layer2	Two-layer RNN
MLP-Layer1	One hidden layer
MLP-Layer2	Two hidden layers
Visual geometry group	VGG
Deep residual networks	ResNet
Accuracy	A
Recall	R
Precision	P
F1-score	F1

References

1. Coronavirus Cases. Available online: <https://www.worldometers.info/coronavirus/> (accessed on 1 May 2022).
2. Dessie, Z.G.; Zewotir, T. Mortality-related risk factors of COVID-19: A systematic review and meta-analysis of 42 studies and 423,117 patients. *BMC Infect. Dis.* **2021**, *21*, 855. [[CrossRef](#)]
3. Kumar, R.; Arora, R.; Bansal, V.; Sahayashela, V.J.; Buckchash, H.; Imran, J.; Narayanan, N.; Pandian, G.N.; Raman, B. Classification of COVID-19 from chest x-ray images using deep features and correlation coefficient. *Multimed. Tools Appl.* **2022**, *81*, 27631–27655. [[CrossRef](#)] [[PubMed](#)]
4. Vaishya, R.; Javaid, M.; Khan, I.H.; Haleem, A. Artificial Intelligence (AI) applications for COVID-19 pandemic. *Diabetes Metab. Syndr. Clin. Res. Rev.* **2020**, *14*, 337–339. [[CrossRef](#)]
5. Huang, S.; Yang, J.; Fong, S.; Zhao, Q. Artificial intelligence in the diagnosis of COVID-19: Challenges and perspectives. *Int. J. Biol. Sci.* **2021**, *17*, 1581. [[CrossRef](#)]
6. Zoabi, Y.; Deri-Rozov, S.; Shomron, N. Machine learning-based prediction of COVID-19 diagnosis based on symptoms. *NPJ Digit. Med.* **2021**, *4*, 3. [[CrossRef](#)] [[PubMed](#)]
7. Podder, P.; Bharati, S.; Mondal, M.R.H.; Kose, U. 9—Application of machine learning for the diagnosis of COVID-19. In *Data Science for COVID-19*; Kose, U., Gupta, D., de Albuquerque, V.H.C., Khanna, A., Eds.; Academic Press: Cambridge, MA, USA, 2021; pp. 175–194. [[CrossRef](#)]
8. Ahmed, H.; Younis, E.M.; Hendawi, A.; Ali, A.A. Heart disease identification from patients' social posts, machine learning solution on Spark. *Future Gener. Comput. Syst.* **2020**, *111*, 714–722. [[CrossRef](#)]
9. Elmannai, H.; El-Rashidy, N.; Mashal, I.; Alohal, M.A.; Farag, S.; El-Sappagh, S.; Saleh, H. Polycystic Ovary Syndrome Detection Machine Learning Model Based on Optimized Feature Selection and Explainable Artificial Intelligence. *Diagnostics* **2023**, *13*, 1506. [[CrossRef](#)]
10. Alsekait, D.M.; Saleh, H.; Gabralla, L.A.; Alnowaiser, K.; El-Sappagh, S.; Sahal, R.; El-Rashidy, N. Toward Comprehensive Chronic Kidney Disease Prediction Based on Ensemble Deep Learning Models. *Appl. Sci.* **2023**, *13*, 3937. [[CrossRef](#)]
11. Sahal, R.; Alsamhi, S.H.; Brown, K.N.; O'Shea, D.; Alouffi, B. Blockchain-based digital twins collaboration for smart pandemic alerting: Decentralized COVID-19 pandemic alerting use case. *Comput. Intell. Neurosci.* **2022**, *2022*, 7786441. [[CrossRef](#)] [[PubMed](#)]

12. Saleh, H.; Mostafa, S.; Alharbi, A.; El-Sappagh, S.; Alkhalifah, T. Heterogeneous ensemble deep learning model for enhanced Arabic sentiment analysis. *Sensors* **2022**, *22*, 3707. [CrossRef]
13. Saleh, H.; Mostafa, S.; Gabralla, L.A.; Aseeri, A.O.; El-Sappagh, S. Enhanced Arabic Sentiment Analysis Using a Novel Stacking Ensemble of Hybrid and Deep Learning Models. *Appl. Sci.* **2022**, *12*, 8967. [CrossRef]
14. Lee, S.J.; Xu, Z.; Li, T.; Yang, Y. A novel bagging C4. 5 algorithm based on wrapper feature selection for supporting wise clinical decision making. *J. Biomed. Inform.* **2018**, *78*, 144–155. [CrossRef]
15. Fusco, R.; Grassi, R.; Granata, V.; Setola, S.V.; Grassi, F.; Cozzi, D.; Pecori, B.; Izzo, F.; Petrillo, A. Artificial intelligence and COVID-19 using chest CT scan and chest X-ray images: Machine learning and deep learning approaches for diagnosis and treatment. *J. Pers. Med.* **2021**, *11*, 993. [CrossRef]
16. Ardabili, S.F.; Mosavi, A.; Ghamisi, P.; Ferdinand, F.; Varkonyi-Koczy, A.R.; Reuter, U.; Rabczuk, T.; Atkinson, P.M. COVID-19 outbreak prediction with machine learning. *Algorithms* **2020**, *13*, 249. [CrossRef]
17. Sumari, P.; Syed, S.J.; Abualigah, L. A novel deep learning pipeline architecture based on CNN to detect Covid-19 in chest X-ray images. *Turk. J. Comput. Math. Educ. (TURCOMAT)* **2021**, *12*, 2001–2011.
18. Kassania, S.H.; Kassanib, P.H.; Wesolowski, M.J.; Schneider, K.A.; Deters, R. Automatic detection of coronavirus disease (COVID-19) in X-ray and CT images: A machine learning based approach. *Biocybern. Biomed. Eng.* **2021**, *41*, 867–879. [CrossRef] [PubMed]
19. Singh, V.; Kamaleswaran, R.; Chalfin, D.; Buño-Soto, A.; San Roman, J.; Rojas-Kenney, E.; Molinaro, R.; von Sengbusch, S.; Hodjat, P.; Comaniciu, D.; et al. A deep learning approach for predicting severity of COVID-19 patients using a parsimonious set of laboratory markers. *Iscience* **2021**, *24*, 103523. [CrossRef] [PubMed]
20. Laatifi, M.; Douzi, S.; Bouklouz, A.; Ezzine, H.; Jaafari, J.; Zaid, Y.; El Ouahidi, B.; Naciri, M. Machine learning approaches in Covid-19 severity risk prediction in Morocco. *J. Big Data* **2022**, *9*, 5. [CrossRef] [PubMed]
21. Jamshidi, M.; Roshani, S.; Daneshfar, F.; Lalbakhsh, A.; Roshani, S.; Parandin, F.; Malek, Z.; Talla, J.; Peroutka, Z.; Jamshidi, A.; et al. Hybrid deep learning techniques for predicting complex phenomena: A review on COVID-19. *AI* **2022**, *3*, 416–433. [CrossRef]
22. Dabbah, M.A.; Reed, A.B.; Booth, A.T.; Yassae, A.; Despotovic, A.; Klasmer, B.; Binning, E.; Aral, M.; Plans, D.; Morelli, D.; et al. Machine learning approach to dynamic risk modeling of mortality in COVID-19: A UK Biobank study. *Sci. Rep.* **2021**, *11*, 16936. [CrossRef]
23. YALÇIN, N.; ÜNALDI, S. Symptom Based COVID-19 Prediction Using Machine Learning and Deep Learning Algorithms. *J. Emerg. Comput. Technol.* **2022**, *2*, 22–29.
24. Symptoms and COVID Presence (May 2020 Data). Available online: <https://www.kaggle.com/datasets/hemanthhari/symptoms-and-covid-presence> (accessed on 1 May 2023).
25. Pal, M.; Parija, S.; Mohapatra, R.K.; Mishra, S.; Rabaan, A.A.; Al Mutair, A.; Alhumaid, S.; Al-Tawfiq, J.A.; Dhama, K. Symptom-Based COVID-19 Prognosis through AI-Based IoT: A Bioinformatics Approach. *BioMed Res. Int.* **2022**, *2022*, 3113119. [CrossRef] [PubMed]
26. Antoñanzas, J.M.; Perramon, A.; López, C.; Boneta, M.; Aguilera, C.; Capdevila, R.; Gatell, A.; Serrano, P.; Poblet, M.; Canadell, D.; et al. Symptom-based predictive model of COVID-19 disease in children. *Viruses* **2021**, *14*, 63. [CrossRef]
27. Guefrechi, S.; Jabra, M.B.; Ammar, A.; Koubaa, A.; Hamam, H. Deep learning based detection of COVID-19 from chest X-ray images. *Multimed. Tools Appl.* **2021**, *80*, 31803–31820. [CrossRef] [PubMed]
28. Apostolopoulos, I.D.; Mpesiana, T.A. COVID-19: Automatic detection from x-ray images utilizing transfer learning with convolutional neural networks. *Phys. Eng. Sci. Med.* **2020**, *43*, 635–640. [CrossRef]
29. Wang, L.; Lin, Z.Q.; Wong, A. Covid-net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. *Sci. Rep.* **2020**, *10*, 19549. [CrossRef]
30. Saha, P.; Neogy, S. Concat_CNN: A Model to Detect COVID-19 from Chest X-ray Images with Deep Learning. *SN Comput. Sci.* **2022**, *3*, 305. [CrossRef]
31. Alqudah, A.M.; Qazan, S.; Alquran, H.; Qasmieh, I.A.; Alqudah, A. COVID-19 detection from X-ray images using different artificial intelligence hybrid models. *Jordan J. Electr. Eng.* **2020**, *6*, 168–178. [CrossRef]
32. Rafiq, A.; Imran, M.; Alhajlah, M.; Mahmood, A.; Karamat, T.; Haneef, M.; Alhajlah, A. Deep Feature Extraction for Detection of COVID-19 Using Deep Learning. *Electronics* **2022**, *11*, 4053. [CrossRef]
33. Elaziz, M.A.; Hosny, K.M.; Salah, A.; Darwish, M.M.; Lu, S.; Sahlol, A.T. New machine learning method for image-based diagnosis of COVID-19. *PLoS ONE* **2020**, *15*, e0235187. [CrossRef]
34. Saad, A.; Kamil, I.S.; Alsayat, A.; Elaraby, A. Classification COVID-19 Based on Enhancement X-ray Images and Low Complexity Model. *Comput. Mater. Contin.* **2022**, *72*, 561–576. [CrossRef]
35. Kc, K.; Yin, Z.; Wu, M.; Wu, Z. Evaluation of deep learning-based approaches for COVID-19 classification based on chest X-ray images. *Signal Image Video Process.* **2021**, *15*, 959–966. [CrossRef] [PubMed]
36. El Asnaoui, K.; Chawki, Y. Using X-ray images and deep learning for automated detection of coronavirus disease. *J. Biomol. Struct. Dyn.* **2021**, *39*, 3615–3626. [CrossRef]
37. Symptoms COVID-19. Available online: <https://github.com/Mahmoudkhorshed-Queens/Covid19-Prediction> (accessed on 1 May 2023).

38. Wanto, A.; Windarto, A.P.; Hartama, D.; Parlina, I. Use of binary sigmoid function and linear identity in artificial neural networks for forecasting population density. *Int. J. Inf. Syst. Technol.* **2017**, *1*, 43–54. [CrossRef]
39. Abirami, S.; Chitra, P. Energy-efficient edge based real-time healthcare support system. In *Advances in Computers*; Elsevier: Amsterdam, The Netherlands, 2020; Volume 117, pp. 339–368.
40. Medsker, L.; Jain, L.C. *Recurrent Neural Networks: Design and Applications*; CRC Press: Boca Raton, FL, USA, 1999.
41. Sherstinsky, A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Phys. D Nonlinear Phenom.* **2020**, *404*, 132306. [CrossRef]
42. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput.* **2019**, *31*, 1235–1270. [CrossRef]
43. Skansi, S. *Introduction to Deep Learning: From Logical Calculus to Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2018.
44. Basili, R.; Croce, D. Recurrent neural networks. *Des. Appl.* **2008**, *5*, 64–67.
45. Pezeshki, M. Sequence modeling using gated recurrent neural networks. *arXiv* **2015**, arXiv:1501.00299.
46. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Gated feedback recurrent neural networks. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 2067–2075.
47. O'Malley, T.; Bursztein, E.; Long, J.; Chollet, F.; Jin, H.; Invernizzi, L. Keras tuner. Retrieved May 2019, 21, 2020.
48. Rajagopal, S.; Kundapur, P.P.; Hareesha, K.S. A stacking ensemble for network intrusion detection using heterogeneous datasets. *Secur. Commun. Networks* **2020**, *2020*, 4586875. [CrossRef]
49. Goyal, S.; Bhatia, P.K. Heterogeneous stacked ensemble classifier for software defect prediction. *Multimed. Tools Appl.* **2021**, 1–23.
50. Sesmero, M.P.; Ledezma, A.I.; Sanchis, A. Generating ensembles of heterogeneous classifiers using stacked generalization. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2015**, *5*, 21–34. [CrossRef]
51. COVID-19 Chest X-ray Images. Available online: <https://www.kaggle.com/datasets/pranavraikokte/covid19-image-dataset> (accessed on 1 May 2023).
52. COVID-19 Chest X-ray Images. Available online: <https://www.kaggle.com/datasets/prashant268/chest-xray-covid19-pneumonia> (accessed on 1 May 2023).
53. Van Dyk, D.A.; Meng, X.L. The art of data augmentation. *J. Comput. Graph. Stat.* **2001**, *10*, 1–50. [CrossRef]
54. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
55. Shorten, C.; Khoshgoftaar, T.M.; Furht, B. Text data augmentation for deep learning. *J. Big Data* **2021**, *8*, 101. [CrossRef] [PubMed]
56. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
57. Carvalho, T.; de Rezende, E.R.; Alves, M.T.; Balieiro, F.K.; Sovat, R.B. Exposing computer generated images by eye's region classification via transfer learning of VGG19 CNN. In Proceedings of the 2017 16th IEEE International Conference on Machine Learning And Applications (ICMLA), Cancun, Mexico, 18–21 December 2017; pp. 866–870.
58. Mascarenhas, S.; Agarwal, M. A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification. In Proceedings of the 2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON), Bengaluru India, 19–21 November 2021; Volume 1, pp. 96–99.
59. Butt, U.M.; Letchmunan, S.; Hassan, F.H.; Zia, S.; Baqir, A. Detecting video surveillance using VGG19 convolutional neural networks. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 674–682. [CrossRef]
60. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
61. Sarker, L.; Islam, M.M.; Hannan, T.; Ahmed, Z. COVID-DenseNet: A deep learning architecture to detect COVID-19 from chest radiology images. *Preprint* **2020**, 2020050151.
62. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
63. Targ, S.; Almeida, D.; Lyman, K. Resnet in resnet: Generalizing residual architectures. *arXiv* **2016**, arXiv:1603.08029.
64. Allen-Zhu, Z.; Li, Y. What can resnet learn efficiently, going beyond kernels? *Adv. Neural Inf. Process. Syst.* **2019**, 32.
65. He, F.; Liu, T.; Tao, D. Why resnet works? residuals generalize. *IEEE Trans. Neural Networks Learn. Syst.* **2020**, *31*, 5349–5362. [CrossRef] [PubMed]
66. Alom, M.Z.; Hasan, M.; Yakopcic, C.; Taha, T.M. Inception recurrent convolutional neural network for object recognition. *arXiv* **2017**, arXiv:1704.07709.
67. Murphy, J. An overview of convolutional neural network architectures for deep learning. *Microway Inc.* **2016**, 1–22.
68. Pak, M.; Kim, S. A review of deep learning in image recognition. In Proceedings of the 2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT), Kuta, Bali, 8–10 August 2017; pp. 1–3.
69. Xiong, Z.; Yuan, Y.; Wang, Q. AI-NET: Attention inception neural networks for hyperspectral image classification. In Proceedings of the IGARSS 2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; pp. 2647–2650.
70. Carreira, J.; Madeira, H.; Silva, J.G. Xception: A technique for the experimental evaluation of dependability in modern computers. *IEEE Trans. Softw. Eng.* **1998**, *24*, 125–136. [CrossRef]
71. Lo, W.W.; Yang, X.; Wang, Y. An xception convolutional neural network for malware classification with transfer learning. In Proceedings of the 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Canary Islands, Spain, 24–26 June 2019; pp. 1–5.

72. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
73. Khasoggi, B.; Ermatita, E.; Sahmin, S. Efficient mobilenet architecture as image recognition on mobile and embedded devices. *Indones. J. Electr. Eng. Comput. Sci.* **2019**, *16*, 2019. [[CrossRef](#)]
74. Brownlee, J. *How to Calculate Precision, Recall, and F-Measure for Imbalanced Classification*; Machine Learning Mastery: San Juan, Puerto Rico, 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.