information

MDPI

Article

Predicting COVID-19 Cases in South Korea with All K-Edited Nearest Neighbors Noise Filter and Machine Learning Techniques

David Opeoluwa Oyewola [1], Emmanuel Gbenga Dada [2], Sanjay Misra [3] and Robertas Damaševičius [4,*]

[1] Department of Mathematics & Computer Science, Federal University Kashere, Gombe PMB 0182, Nigeria; davidoyewole@fukashere.edu.ng

[2] Department of Mathematical Sciences, University of Maiduguri, Maiduguri PMB 1069, Nigeria; gbengadada@unimaid.edu.ng

[3] Department of Computer Science and Communication, Østfold University College, 3001 Halden, Norway; ssopam@gmail.com

[4] Department of Applied Informatics, Vytautas Magnus University, 44404 Kaunas, Lithuania

* Correspondence: robertas.damasevicius@vdu.lt

Abstract: The application of machine learning techniques to the epidemiology of COVID-19 is a necessary measure that can be exploited to curtail the further spread of this endemic. Conventional techniques used to determine the epidemiology of COVID-19 are slow and costly, and data are scarce. We investigate the effects of noise filters on the performance of machine learning algorithms on the COVID-19 epidemiology dataset. Noise filter algorithms are used to remove noise from the datasets utilized in this study. We applied nine machine learning techniques to classify the epidemiology of COVID-19, which are bagging, boosting, support vector machine, bidirectional long short-term memory, decision tree, naïve Bayes, k-nearest neighbor, random forest, and multinomial logistic regression. Data from patients who contracted coronavirus disease were collected from the Kaggle database between 23 January 2020 and 24 June 2020. Noisy and filtered data were used in our experiments. As a result of denoising, machine learning models have produced high results for the prediction of COVID-19 cases in South Korea. For isolated cases after performing noise filtering operations, machine learning techniques achieved an accuracy between 98–100%. The results indicate that filtering noise from the dataset can improve the accuracy of COVID-19 case prediction algorithms.

Keywords: healthcare data mining; COVID-19 case prediction; noise filtering; machine learning; data mining; predictive analytics; artificial intelligence; neural networks

check for updates

Citation: Oyewola, D.O.; Dada, E.G.; Misra, S.; Damaševičius, R. Predicting COVID-19 Cases in South Korea with All K-Edited Nearest Neighbors Noise Filter and Machine Learning Techniques. *Information* **2021**, *12*, 528. https://doi.org/10.3390/info12120528

Academic Editors: Agnes Vathy-Fogarassy and János Abonyi

Received: 11 November 2021
Accepted: 17 December 2021
Published: 19 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

On 30 December 2019, the first diagnosis of COVID-19 was first reported at Wuhan Jinyintan Hospital in a patient with pneumonia of unknown etiology. The result showed that the virus had a family of coronaviruses called Betacoronavirus 2B [1]. Coronavirus bat-like SARS exhibited a close link to the virus of COVID-19. The World Health Organization (WHO) identified the novel coronavirus as extreme acute coronavirus syndrome 2 (SARS-COV-2) and referred to it as coronavirus disorder 2019 (COVID-19) on 30 January 2020 [2]. Symptoms of breathlessness, fever, headache, chills, myalgia or arthralgia, congested nose, diarrhea, hemoptysis, and conjunctival obstruction are typical symptoms of the disease [3]. This can result in kidney failure, death, and severe acute respiratory syndrome in severe cases of the coronavirus disease [4]. The present spread of coronavirus (COVID-19) threatens national health systems in all nations [5]. The United States has become one of the most affected countries to be hit by the increase in COVID-19 in public health, emergency health care, and hospitals [6]. Unfortunately, the rate of infections is expected to increase exponentially in many countries regardless of their health systems. Emergency steps are needed to provide good medical equipment and high-quality information in the health care system and hospital. Translational science can provide stakeholders and clinicians with appropriate evidence-based medicine concepts [7]. The world has recorded 252,469,591 confirmed cases of COVID-19 and 5,093,058 deaths in 222 countries in 30 January 2021 [8].

Management measures to mitigate transmission include the wearing of masks, hygienic hand screening, avoiding public contact, case identification, contact control, and quarantines [9]. COVID-19 infected subjects depend on symptomatic treatment, since no successful antiviral therapy has been discovered [10]. A scoping analysis suggested in [11] was adopted to investigate COVID-19 in [12] to better understand the cause, prevention, diagnosis, and control of this coronavirus. However, research papers focused primarily on causes, but prevention and regulation have improved over time. Diagnostic tests can discover viral infections that can be contagious and cause infection in other people. Antibody tests, conversely, examine whether the person has previously been infected with the virus. There are three reasons for testing for COVID-19:

1. Surveillance allows the government and health officials to monitor the rate of infections in a particular community. It seeks to observe the effectiveness of COVID-19 prevention measures, such as wearing a mask and maintaining social distancing. This could involve random testing of people in a particular location to know whether there is community transmission of the disease or not [13];
2. Screening involves testing anyone regardless of whether they show symptoms or are unaware of their exposure to someone who has been infected. It provides an effective means of recognizing those who are likely to have been infected with the virus to stop further transmission [14];
3. Diagnostic testing involves testing a person who is assumed to have been infected with COVID-19. The person may show symptoms of COVID-19, know that they have contacted people with confirmed cases of COVID-19 or have been infected, and is trying to perform more tests to verify that they are now negative [15].

Machine learning (ML) algorithms are used to solve problems by analyzing and interpreting large volumes of data to solve problems in the medical sector [16–20]. Several researchers have used machine learning algorithms to solve medical difficulties in this area. Since the beginning of the pandemic, a wide range of studies have been conducted to provide a better understanding of the case, prevention, diagnosis, and control of COVID-19. In this paper, ML algorithms are applied to determine the epidemiology of the COVID-19 pandemic. The ML algorithms have been shown to be very effective and robust algorithms that can handle large data successfully. Therefore, it can be used to analyze the epidemiology of COVID-19 [21–29].

The major contributions of this work include:

i. The exploration of dataset noise filtering techniques (all k-edited nearest neighbors, blame based noise reduction, and condensed nearest neighbors) on the dataset of COVID-19 infection cases in South Korea, which has not been conducted before;
ii. The combination of noise filtering combined with machine learning techniques in epidemiological data for the prediction of COVID-19 cases;
iii. The performance evaluation of all k-edited nearest-neighbors noise filters combined with machine learning algorithms using different performance metrics.

The rest of this paper is organized as follows: Section 2 is a review of the literature; Section 3 discusses the materials and methods used in this work, as well as our performance measurements used; the results and the discussion are presented in Section 4; and Section 5 presents the conclusions.

2. Review of Related Works

In this section, we succinctly discuss recent research conducted in the field of application of machine learning to the COVID-19 pandemic. This Section follows from our explanations in Section 1 of this paper, in which it was pointed out that machine learning algorithms have gained wide acceptance by data scientists and researchers as a viable tool for solving the COVID-19 crisis. This is due to the effectiveness of these algorithms in the detection and diagnosis of health-related problems. For example, Nemati et al. [21] proposed the combination of statistical methods, support vector machine (SVM), and ensemble techniques that use COVID-19 data of patients to predict the date they are likely to be discharged from the isolation center. It also evaluates clinical information to determine the duration of the patient in the hospital. The downside of this work is that it is just a framework and there is no practical implementation of any machine learning or statistical algorithm. The effectiveness of the proposed method was also not evaluated.

Lalmuanawma et al. [22] presented a review on the role of artificial intelligence (AI) and machine learning (ML) in investigating and predicting the transmission rate of COVID-19. They also examined how these techniques can be used to recognize, evaluate, and handle people who have been exposed to COVID-19 to prevent further transmission. Furthermore, the authors examined how AI and ML can help in the process of bringing a new pharmaceutical drugs into clinical practice for SARS-CoV-2 and its associated endemic. The findings of this study indicated that AI and ML have significantly improved the treatment, testing, prediction, and cure/immunization steps needed to take COVID-19 drugs from concept to market availability. Malik et al. [23] used multiple machine learning models to obtain the correlation between different characteristics and the rate of transmission of COVID-19. ML models were used to evaluate the effect of climatic factors on the spread of COVID-19 by mining the connection between the number of confirmed cases and the variables of atmospheric condition variables in some counties. The authors opined that atmospheric characteristics are of great significance in forecasting the number of deaths due to COVID-19 compared to other factors mentioned in the paper. Kavadi et al. [24] developed a partial derivative regression and nonlinear machine learning (PDR-NML) method for predicting COVID-19. The PDR was used to explore the dataset for optimal parameters with little computer resource usage. Subsequently, the machine learning model was used to normalize the attributes that are used to make predictions with high accuracy. In a more specific study, Amar et al. [25] used various machine learning and statistical techniques to predict the transmission of the COVID-19 pandemic in Egypt. The authors aimed to assist the Egyptian government in managing the pandemic in the subsequent months. The experimental results showed that the exponential model outperforms other models compared in the paper. The authors deduced from their results that the COVID-19 pandemic in Egypt is not likely to end soon.

Goodman-Meza et al. [26] applied ensemble machine learning to diagnose COVID-19 in patients admitted to hospital and receiving treatment. The patients are in an environment where the PCR test is insufficient or inaccessible. The performance is good, though there is still room for improvement. The authors did not propose any new machine learning algorithm; rather, they only used an ensemble of different machine learning models. Ozturk et al. [27] used deep neural network models to automatically detect COVID-19 from rib cage radiographs of patients. Their model can classify images into two classes or more than two classes. The model can serve as a secondary or assisting diagnosis tool, especially in places where there is an unavailability of medical experts. The classification accuracy of the model is high for binary classes; however, the accuracy for multiclass is poor.

Khan et al. [28] suggested employing parallel fusion and deep learning model optimization with a contrast enhancement using a top-hat and Wiener filter combination. Two deep learning models (AlexNet and VGG16) that have been pre-trained are used and fine-tuned based on the target classes (COVID-19 and healthy). A parallel fusion approach is used, parallel positive correlation, to extract and fuse features. The entropy-controlled firefly optimization approach is used to identify optimal features. Machine learning classifiers, such as the multiclass SVM, are used for classification.

Rehman et al. [29] presented a framework for the diagnosis of 15 different forms of chest disease, including COVID-19, using a chest radiograph modality. They used a convolutional neural network (CNN) with a softmax classifier and a fully connected layer to extract deep features, which are input into traditional machine learning (ML) classification algorithms. The suggested architecture, conversely, improves the accuracy of COVID-19 detection and increases the prediction rates for other chest disorders.

Rustam et al. [30] investigated the ability of four machine learning models to predict the number of people who will be infected with COVID-19. Each of the models was used to predict the number of new confirmed cases, death toll, and the number of recovered cases in a period of 10 days. The results show that the predictive capability of the models under investigation was not very good. Therefore, there is a need to try other machine learning models.

Wieczorek et al. in [31,32] used artificial neural networks (ANN) to estimate future COVID-19 cases using geolocation and past case data. The results of the proposed model show high accuracy, which in some cases reaches above 99%. Ahouz and Golabpour [33] developed a least-squares-boosting classification model to predict the incidence rate two weeks in advance. The proposed model predicted the number of globally confirmed cases of COVID-19 with an accuracy of 98.45%. Zivkovic et al. [34] proposed a hybridized method combining machine learning, adaptive neurofuzzy inference system (ANFIS), and enhanced beetle antennae search metaheuristics. The proposed model achieved a correlation of 0.9763 correlation on China's COVID-19 outbreak data. For more related works, we would like to refer the readers to the review papers [35,36].

In summary, current machine learning methods have not been very successful in the prediction of confirmed cases due to challenges, such as the lack of historical data and the different approaches of governments toward testing, which makes the results hardly comparable [37]. The prediction of COVID-19 cases using the deep learning method has gained more attention currently due to the unavailability of more data. Deep learning methods can specifically handle nonlinear problems more effectively. However, they still face the same problems of governmental actions that influence the data [38].

3. Methodology

3.1. Dataset

The dataset used in this research comprises epidemiological data of COVID-19 infection cases in South Korea, which were obtained from the Kaggle database. The dataset is composed of data from 23 January 2020 to 24 June 2020 recorded daily, patient ID, sex, age, country, province, city, infected by, contact number, symptom onset date, confirmed date, released date, and state (which consists of released, deceased, and isolated). In this study, due to the nature of the dataset, we have extracted sex, age, country, symptom onset date, confirmed date, released date, and the state features as shown in Table 1. The no CS feature is the number of days from symptom onset to disease confirmation. It is obtained by subtracting the confirmed date from the symptom onset date, while the no RC feature, which is the number of days between the confirmation of disease to release from hospital, is obtained from subtracting the released date from the confirmed date.

Table 1. Description of COVID-19 infection cases.

S/No.	Attribute	Data Type
1.	Sex	Categorical
2.	Age	Nominal
3.	Country	Nominal
4.	Symptom onset date	Interval
5.	Confirmed date	Interval
6.	Released date	Interval
7.	State	Categorical

3.2. Machine Learning Algorithms

In this subsection, we briefly discuss the machine learning algorithms that are used for this work. We discuss bagging, stochastic gradient boosting, bi-directional long short-term memory, support vector machine, naïve Bayes, random forest, k-nearest neighbor, decision tree, and logistic regression classifiers, as well as noise filtering methods.

3.2.1. Bagging (BAG)

Bagging is a method that combines the predictions of many simple estimators with a given algorithm, so that generalizability and robustness can be improved over a single estimator [39]. Decisions made by multiple learners can be integrated into a single prediction. In the case of classification, it is a vote to combine these decisions. Models of bagging bear the same weight as good models of bagging because an executive can use a collection of expert advice based on their previous right predictions to achieve other outcomes. The model in which one obtains more votes than others is considered correct.

$$H\left(d_i,\ c_j\right) = \sum_{m=1}^{M} \alpha_m H_m\left(d_i, c_j\right), \tag{1}$$

where H_m are weak classifiers that decide over a subset of a dataset d_i with class c_j; d_i is classified into the classes c_j; and α_m is the weight of weak classifier H_m.

3.2.2. Stochastic Gradient Boosting (BST)

The stochastic gradient boosting (BST) method is a hybrid of boosting and bagging proposed by Friedman [40]. The BST is a set of learning algorithms with a combination of boosting and decision trees, which classifies the value of all trees by weighting all trees. The new model is constructed along the path of gradient descent of the loss function of the previous three. Th eloss function between classification and actual function is reduced by the training function of the classification function. The loss function is given as:

$$\rho(y_k, F_k(x)) = \sum_{k=0}^{K} y_k \log\left[\frac{e^{F_k(x)}}{\sum_{k=1}^{K} e^{F_k(x)}}\right], \tag{2}$$

$$\hat{y_k} = -\left[\frac{\partial\rho(Y_k, F_k(x))}{\partial F_k(x)}\right] = y_k - P_k(x), \tag{3}$$

where ρ is the loss function; y_k is the k-th output variable; x is the vector of input variables; $F_k(x)$ is the function that maps from input vector x to y_k; K is the number of classes; and $P_k(x)$ is the probability of k-th class given input vector x.

3.2.3. Bi-Directional Long Short-Term Memory (BLSTM)

Bi-directional long short-term memory (BLSTM) combines long short-term memory (LSTM) and bi-directional recurrent neural network (BiRNN) [41] for the analysis of classification and time-series data. The benefit of a recurrent neural network (RNN) is to encode dependencies between inputs. For long data classification, the RNN causes its gradient to erupt and vanish. LSTM is subsequently developed to address RNN long-term problems. There are three gates to LSTM. Input gate is required for the layer of input and also output and forget gate inclusive. Moreover, both LSTM and RNN can only obtain information from the past so that additional changes are made through the bi-directional network. Two pieces of information from front and back can be managed by BiRNN. The combination of BiRNN and LSTM generates BLSTM. Thus, a combination of LSTM advantages as a cell memory and BiRNN with context access information make BLSTM perform better. This allows the BLSTM to benefit from the input of LSTM for the next layer. However, BLSTM is also capable of handling long-range data. The forward function of BLSTM with inputs of L units and H as the number of hidden units is expressed by Equations (4) and (5), while Equations (6) and (7) are the backward calculation of BLTSM:

$$a_h^t = \sum_{i=1}^{L} x_i^t W_{ih} + \sum_{h=1}^{H} b_{h'}^{t-1} W_{h'h}, \tag{4}$$

$$b_h^t = f\left(a_h^t\right), \tag{5}$$

$$\frac{\delta O}{\delta W_{hk}} = \sum\nolimits_{i=1}^{T} \frac{\delta O}{\delta a_h^t} b_h^t, \tag{6}$$

$$\frac{\delta O}{\delta a_h^t} = f\left(a_h^t\left(\sum\nolimits_{k=1}^{K} \frac{\delta O}{\delta a_h^t} W_{hk} + \sum\nolimits_{h'=1}^{H} \frac{\delta O}{\delta a_h^{t+1}} W_{hh'}\right)\right), \tag{7}$$

where x^t is the input vector at time t; a_h^t is the network input to LSTM of unit h at time t; and the activation function of h at time t is denoted by b_h^t. W_{ih} is the weight of the input i towards h and $W_{h'h}$ is the weight of the hidden unit h towards the hidden unit h'. f is an activation function of the hidden unit of h and O is an objective function with unit K output.

3.2.4. Support Vector Machine (SVM)

The support vector machine (SVM) procedure categorizes both linear and non-linear data [42]. SVM uses a non-linear mapping to transform the training set to a high level. In this new dimension, SVM explores the ideal linear hyperplane separation as a decision limit by which the tuples of a class of one class are split from another. Two class data can be separated by a hyperplane with the proper, non-linear upper dimensional mapping. In contrast to the other approaches, hyperplanes are robust for overfitting.

3.2.5. Naïve Bayes (NB)

Naïve Bayes (NB) is one of the probabilistic methods that is used to describe, use, and acquire information. A maximum posterior rule is an approach for classifying a test sample x, to construct a probabilistic model for estimating the corresponding likelihood $P(y)$, and to measure it with the largest context likelihood. The Bayes theorem is given by:

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}, \tag{8}$$

where x is the input variable; P is the probability; and y is the target variable.

3.2.6. Random Forest (RF)

Random forest (RF) [43] is a decision-making ensemble classifier with various types of trees. An arbitrary sequence of features at each node is used to evaluate the division to create a decision tree. Each tree is based on the individual values of a random variable. We can shape an RF using bagging along with the selection of the random attribute, using the CART method, to increase the trees. RF uses a random linear combination of the input attributes. The sub-cluster of features is not chosen randomly, but new attributes are created, which reflect a linear combination of existing features.

3.2.7. K-Nearest Neighbor (KNN)

K-nearest neighbor (KNN) [44] is a lazy learning technique that learns by comparison of a tested sample with similar training samples. A distance metric, such as Euclidean distance, describes closeness. To classify using KNN, the sample that is not known is classified as the most common class among its neighbors.

3.2.8. Decision Tree (DT)

Decision trees (DT) classify by dividing training data into pieces and mainly holding the result of each part. It is a natural non-parametric supervised learning model, called classification and regression tree (CART), which produces accurate classifications with easily understood regulations. Model transparency makes them highly relevant.

3.2.9. Multinomial Logistic Regression (MLR)

The multinomial logistic regression (MLR) model that contains more than two target variables, discrete and unordered categories, with nominal features and a multinomial distribution, represents an extension of the binomial logistic regression. LR with a single category dependent variable must have logistic regression. The likelihood that a target variable is labeled as k-th is defined in the LR as in Equation (9):

$$\pi(x) = \frac{e^{\alpha+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_k x_k}}{1+e^{\alpha+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_k x_k}} \tag{9}$$

where $\pi(x)$ defines the natural logarithm of the odds ratio given an independent variable vector x; α and β signify the coefficients of parameters; and x_i represents the i-th independent variable.

3.3. Noise Filtering Methods

The effectiveness of the classifiers that we typically want to optimize under those circumstances will not only depend on the quality of the data, but also on the robustness of the noise-reduction method. Therefore, analyzing noise data is challenging, and it is often difficult to find accurate solutions [45–47]. Data noise can affect the inherent essence of a classification problem, as this can lead to the introduction of new properties into the problem area. The data from the real world usually contain noise and sometimes they are corrupted. These can hamper the efficiency of the method. Data from the real-world are therefore never flawless and frequently suffer from manipulation that can impair system efficiency. To have clean data from the classes of released, deceased, and isolated, we employed three noise filter algorithms, which are discussed below.

3.3.1. All K-Edited Nearest Neighbors (AENN)

The all k-edited nearest neighbors [48] method classifies each training dataset using samples $x \in D$, where D is called a design set. A new design set D' contains exactly those samples from D, which have been classified correctly. For a given value of k and a given sample x, the procedure of AENN is as follows:

4. If $i = 1$, find i nearest neighbors of x;
5. If the majority of $k(x, i)$ classify x incorrect and end;
6. $i++$;
7. If $i < k$ go to step 2, otherwise end;
8. After processing all samples from D, eliminate incorrectly classified samples.

3.3.2. Blame Based Noise Reduction (BBNR)

Blamed based noise reduction (BBNR) [49] emphasizes the cases that cause misclassifications rather than the cases that are misclassified. It attempts to remove mislabeled cases and unhelpful cases that cause misclassification as follows:

1. For each case (c) in Training set T;
2. Split the training set into two which are coverage set C and liability set L;
3. Sort L in descending order;
4. While $|L > 0|$;
5. $T = T - c$, misclassifiedFlag = False;
6. For each x in C;
7. If x cannot be correctly classified by T, misclassifiedFlag = true;
8. End if,
9. If misclassifiedFlag = true, $T = T + c$.

3.3.3. Condensed Nearest Neighbors (CNN)

The condensed nearest neighbors (CNN) was developed in [50]. The training sample set is divided into STORE and GRABBAG as follows:

1. The first training sample is placed in STORE;
2. The second sample that is correctly classified using the KNN rule is placed in GRABBAG, but if it is incorrectly classified it is placed in STORE;

3. Loop through GRABBAG until termination is reached when any of the following conditions are satisfied:
    a. GRABBAG is empty, with all its members now transferred to STORE; or
    b. A complete pass is made through GRABBAG with no transfer to the STORE.
4. The content of STORE is used as reference points for the KNN.

3.3.4. Computational Complexity of the Methods

The computational complexity of the AENN method is $O\ (n \times d \times k)$, where 'n' is the number of training features, d is the number of dimensions, and k is the number of neighbors considered.

The computational complexity of the BBNR method is quadratic, because it needs to perform the classification with respect of each neighbor removed from the liability dataset, i.e., its computational complexity is $O\ (n \times n)$.

The computational complexity of the CNN method grows quadratically with the number of training samples, because the K-NN-based training set filtering technique is employed in the development stage of the proposed strategy, and identifying the NNs for each training sample requires computing the distances between all training samples, so it is $O\ (n \times n)$.

3.4. Performance Measures

In this study, accuracy (A_c), sensitivity (S_e), specificity (S_p), Kappa (K), and balanced accuracy (BA) are used.

$$A_C = 1 - \frac{F_N + F_P}{T_N + F_N + T_P + F_P}, \tag{10}$$

$$S_e = \frac{T_P}{T_P + F_N}, \tag{11}$$

$$S_p = \frac{T_N}{T_N + F_P}, \tag{12}$$

$$K = \frac{P_o - P_e}{1 - P_e}, \tag{13}$$

$$BA = \frac{S_e + S_p}{2}, \tag{14}$$

where T_P is the true positive; T_N is the true negative; F_P is the false positive; F_N is the false negative; P_o is the probability of the observed accuracy; and P_e is the probability of expected accuracy obtained from the confusion matrix.

4. Results and Discussion

This section presents the experimental results of machine learning techniques, such as bagging (BAG), stochastic gradient boosting (BST), bi-directional long short-term memory (BLSTM), support vector machine (SVM), naïve Bayes (NB), random forest (RF), k-nearest neighborhood (KNN), decision tree, and the multinomial logistic regression (LR) for the diagnosis of COVID-19 infection cases.

For our experiments, we used MATLAB 2021a (MathWorks Inc., Nattick, MA, USA) on a laptop computer with 64-bit Windows 10 OS with Intel Core i5-8265U CPU 1.80 GHz with 8 GB RAM.

We compared the performances of the algorithms under consideration using sensitivity, specificity, and balanced accuracy, kappa, accuracy, and p-value to discern which is more accurate in the diagnosis of COVID-19 cases, such as the number of released, deceased, and isolated cases. We used data from the Kaggle database for COVID-19 infection cases in South Korea. The data were segmented into both training (60%) and testing (40%) datasets. The training set was used to train the model, while the test set was used to test it.

Classification of data has three classes—released, deceased, and an isolated class, consisting of 5165 data samples.

Table 2 shows the comparison of the performance metrics used in this research: sensitivity, specificity, and balanced accuracy. Most of the machine learning algorithms, such as BAG, BST, BLSTM, SVM, NB, RF, KNN, DT, and LR, can classify isolated and released classes, but fails to classify the deceased in sensitivity metrics. The specificity of the three classes, released, deceased, and isolated, is within the range of 78–100%, except in the isolated class of BLSTM.

Table 2. Performance Metrics of BAG, BST, BLSTM, SVM, NB, RF, KNN, DT, and LR.

Algorithm	**Sensitivity (%)**			**Specificity (%)**			**Balanced Accuracy (%)**		
	Released	**Deceased**	**Isolate**	**Released**	**Deceased**	**Isolate**	**Released**	**Deceased**	**Isolate**
BAG	80.57	0.00	90.84	90.86	100.00	78.49	85.72	50.00	84.67
BST	80.64	0.00	88.69	89.07	100.00	78.49	84.85	50.00	83.59
BLSTM	29.08	0.00	98.36	98.39	100.00	29.30	64.18	50.00	63.83
SVM	79.24	0.00	92.79	92.83	100.00	77.14	86.04	50.00	84.96
NB	76.53	2.22	99.81	99.46	98.07	77.01	87.99	50.14	88.41
RF	80.44	0.00	90.06	90.32	100.00	78.30	85.38	50.00	84.18
KNN	80.37	0.00	90.06	90.50	100.00	78.17	85.44	50.00	84.11
DT	80.70	0.00	88.69	89.07	100.00	78.56	84.89	50.00	83.63
LR	79.64	0.00	99.22	98.75	100.00	77.53	89.19	50.00	88.37

Table 3 shows the comparison of accuracy, kappa, and p-value of BAG, BST, BLSTM, SVM, NB, RF, KNN, DT, and LR. The overall best accuracy was obtained from LR with an accuracy of 82.77%, while the lowest was obtained from BLSTM with an accuracy of 65.96%. The result is not encouraging when compared with other state-of-the-art techniques. We use the proposed method to filter the noise from the COVID-19 dataset.

Table 3. Performance Metrics of BAG, BST, BLSTM, SVM, NB, RF, KNN, DT, and LR.

Algorithm	**Accuracy (%)**	**Kappa**	**p-Value**
BAG	81.36	0.5919	2.2×10^{-6}
BST	80.88	0.5789	2.2×10^{-6}
BLSTM	65.96	0.2904	2.2×10^{-6}
SVM	80.88	0.5878	2.2×10^{-6}
NB	80.69	0.6029	2.2×10^{-6}
RF	81.07	0.5853	2.2×10^{-6}
KNN	81.03	0.5846	2.2×10^{-6}
DT	80.93	0.5797	2.2×10^{-6}
LR	82.77	0.6335	2.2×10^{-6}

Table 4 presents the performance comparison of all the ML models for the AENN filtered dataset using sensitivity, specificity, and balanced accuracy. LR attained 82.77% accuracy, while BLSTM produced the worst accuracy, at 65.96%.

Table 4. Performance Metrics of AENN on BAG, BST, BLSTM, SVM, NB, RF, KNN, DT, and LR. The best results are shown in bold.

Algorithm	Sensitivity (%)	Specificity (%)	Balanced Accuracy (%)
BAG	**100.00**	**100.00**	**100.00**
BST	97.56	99.96	98.76
BLSTM	63.33	99.22	81.27
SVM	29.26	100.00	64.63
NB	87.81	98.76	93.28
RF	**100.00**	**100.00**	**100.00**
KNN	73.17	99.75	86.46
DT	39.02	99.96	69.49
LR	36.58	99.71	68.15

Table 5 presents the performance comparison of all the ML models for the AENN filtered dataset using accuracy, kappa, and *p*-value. Both BAG and RF attained 100% accuracy, while LR produced the worst accuracy, at 98.81%.

Table 5. Performance of AENN on BAG, BST, BLSTM, SVM, NB, RF, KNN, DT, and LR.

Algorithm	Accuracy (%)	Kappa	*p*-Value
BAG	100.00	1.00	2.2×10^{-6}
BST	99.93	0.9753	1.05×10^{-15}
BLSTM	98.84	0.5295	0.7454
SVM	98.98	0.4492	0.03
NB	98.60	0.6362	0.47
RF	100.00	1.00	2.2×10^{-6}
KNN	99.37	0.766	4.11×10^{-5}
DT	99.09	0.5479	0.007
LR	98.81	0.4632	0.15

Table 6 depicts the performance comparison of all the ML algorithms on the BBNR filtered dataset using sensitivity, specificity, and balanced accuracy. Both SVM and NB achieved 100% sensitivity and specificity, while LR produced the worst accuracy of the results.

Table 6. Performance of BBNR on BAG, BST, BLSTM, SVM, NB, RF, KNN, DT, and LR.

Algorithm	Sensitivity (%)			Specificity (%)			Balanced Accuracy (%)		
	Released	Deceased	Isolated	Released	Deceased	Isolate	Released	Deceased	Isolated
BAG	66.36	2.85	87.13	87.54	100.00	64.87	76.95	51.43	76.00
BST	64.40	0.00	88.50	88.86	100.00	62.89	76.63	50.00	75.69
BLSTM	65.22	0.00	84.82	85.30	100.00	63.69	75.26	50.00	74.25
SVM	54.05	0.00	100.00	100.00	100.00	52.79	77.03	50.00	76.39
NB	24.74	0.00	100.00	100.00	100.00	24.16	62.37	50.00	62.08
RF	58.42	2.85	97.74	97.81	100.00	57.11	78.11	51.42	77.43
KNN	99.21	0.00	3.34	3.42	100.00	99.09	51.31	50.00	51.22
DT	56.53	0.00	98.63	98.68	100.00	55.20	77.60	50.00	76.92
LR	54.57	0.00	99.53	99.50	100.00	53.32	77.03	50.00	76.43

Table 7 represents the performance comparison of accuracy, kappa, and *p*-value of all the ML models for the BBNR filtered dataset. BAG produced the best performance, closely followed by RF with 74.12% and 74.01% accuracy, respectively, while NB produced the worst accuracy, at 55.69%.

Table 7. Performance of BBNR on BAG, BST, BLSTM, SVM, NB, RF, KNN, DT, and LR. The best results are shown in bold.

Algorithm	Accuracy (%)	Kappa	*p*-Value
BAG	**74.12**	**0.50**	2.2×10^{-16}
BST	73.53	0.49	2.2×10^{-16}
BLSTM	72.48	0.47	2.2×10^{-16}
SVM	72.42	0.49	2.2×10^{-16}
NB	55.69	0.21	0.98
RF	**74.01**	**0.51**	2.2×10^{-16}
KNN	57.99	0.02	0.08
DT	73.26	0.49	2.2×10^{-16}
LR	72.52	0.49	2.2×10^{-16}

Table 8 depicts the performance comparison of all the ML algorithms on dataset that was filtered by CNN using sensitivity, specificity, and balanced accuracy as performance metrics. NB achieved a sensitivity of 99.32% and specificity of 100%.

Table 8. Performance of CNN on BAG, BST, BLSTM, SVM, NB, RF, KNN, DT, and LR.

Algorithm	Sensitivity (%)			Specificity (%)			Balanced Accuracy (%)		
	Released	Deceased	Isolated	Released	Deceased	Isolated	Released	Deceased	Isolated
BAG	69.82	12.82	95.48	95.60	99.79	63.13	82.71	56.30	79.30
BST	60.56	1.28	96.11	96.11	99.95	53.22	78.34	50.62	74.66
BLSTM	67.44	0.00	94.06	94.14	100.00	58.78	80.77	50.00	76.42
SVM	12.67	1.28	99.74	99.64	99.91	11.83	56.16	50.59	55.78
NB	99.32	0.00	14.88	13.22	100.00	99.24	56.27	50.00	57.06
RF	68.61	15.38	95.74	95.96	99.70	62.43	82.28	57.54	79.09
KNN	68.21	8.97	95.06	95.15	99.79	61.39	81.68	54.38	78.22
DT	60.56	0.00	96.11	96.11	100.00	52.87	78.34	50.00	74.49
LR	15.69	1.28	98.74	98.63	99.87	14.78	57.16	50.57	56.76

Table 9 shows the performance comparison of all the ML models for the CNN filtered the dataset using accuracy, kappa, and *p*-value. RF produced the best performance and was closely followed by BAG with 87.76% and 87.72% accuracy, respectively, while SVM produced the worst accuracy, at 79.16%. The main result of Table 9 is that the BAG and RF methods achieve the best performance in terms of accuracy and kappa.

Table 9. Performance of CNN on BAG, BST, BLSTM, SVM, NB, RF, KNN, DT, and LR. The best results are shown in bold.

Algorithm	Accuracy (%)	Kappa	*p*-Value
BAG	**87.72**	**0.63**	2.2×10^{-16}
BST	85.99	0.56	2.2×10^{-16}
BLSTM	85.74	0.57	2.2×10^{-16}
SVM	79.16	0.16	0.00
NB	79.24	0.17	0.00
RF	**87.76**	**0.63**	2.2×10^{-16}
KNN	86.95	0.61	2.2×10^{-16}
DT	85.95	0.55	2.2×10^{-16}
LR	79.00	0.18	0.00

The accuracy results from Tables 5, 7 and 9 are visualized in Figure 1. We summarize the results of experiments in Figure 2, which shows that the AENN method allows to achieve a statistically significant improvement ($p < 0.001$, using the *t*-test) of classification performance in terms of accuracy metric. AENN, on average, improved the accuracy by 19.7833 ± 4.9896% and BBNR was ineffective and led to the decrease in performance by 9.9500 ± 9.3480%, while the CNN filtering method increased the accuracy by 4.6600 ± 6.9520%.

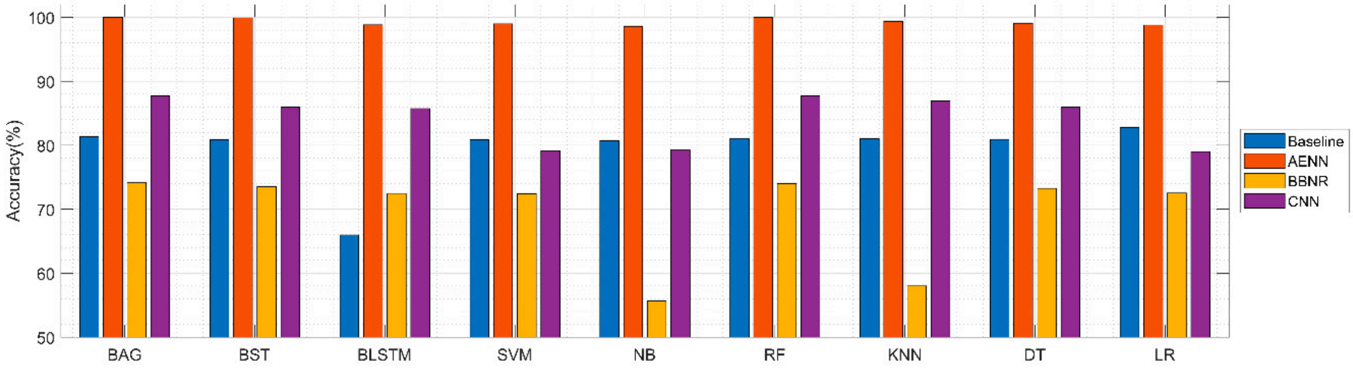

Figure 1. Performance of baseline (no filtering used) and filtering methods (AENN, BBNR, and CNN).

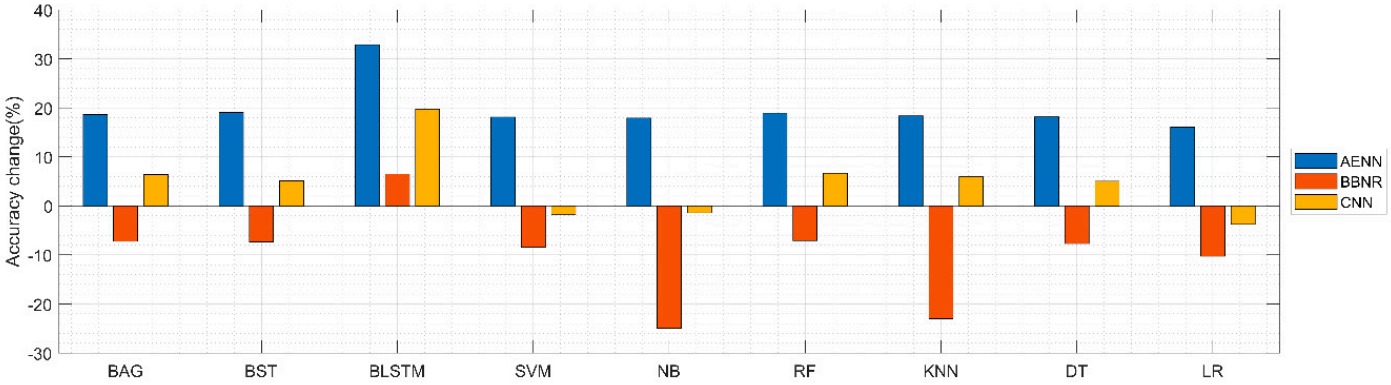

Figure 2. Performance change of filtering methods (AENN, BBNR, and CNN) as compared to baseline (no filtering used) results.

The results of our study underscore the need for data filtering to improve the performance of machine learning classifiers. This study demonstrated the superiority of the AENN filtering method, which outperformed the BBNR and CNN filtering methods. This finding is in line with other recent studies [51–53]. However, more research is needed to confirm our results.

The limitation of the current study is that only a limited dataset from a single country was used. More research with larger datasets is still needed to validate the proposed methods.

5. Conclusions

Machine learning techniques have been successful in the classification and prediction of sequential data in recent years. Several algorithms, for example, gradient boosting and neural networks, were explored for strength in the classification of COVID-19. However, the removal of noise from the data has remained unexploited in this field. In this paper, we have used noise filter algorithms to remove noise from all data sets utilized in this study. As a result of denoising, machine learning models have produced high results for the prediction of COVID-19 cases in South Korea. The technique has proven to be effective in the classification of released, deceased, and isolated classes. The presented methodology can contribute to the analysis of epidemiological data and the monitoring of the spread of infections. The results of this study can catalyze the governments of nations to take well-timed actions and make quality decisions to effectively address the COVID-19 emergency.

In the future, this work will be continuously enhanced by exploring more efficient machine learning and deep learning models to determine the epidemiology of COVID-19 in real-time using the up-to-date datasets. Further validation of our framework on other, possible larger datasets, if they become available, will also be a subject of our future work.

Author Contributions: Conceptualization, S.M.; methodology, S.M.; software, D.O.O. and E.G.D.; validation, D.O.O., E.G.D., S.M. and R.D.; formal analysis, S.M. and R.D.; investigation, D.O.O. and E.G.D.; writing—original draft preparation, D.O.O. and E.G.D.; writing—review and editing, S.M. and R.D.; visualization, D.O.O. and E.G.D.; supervision, S.M.; funding acquisition, R.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset is available from: https://www.kaggle.com/kimjihoo/coronavirusdataset (accessed on 12 December 2021).

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

Abbreviations

Notation	Description
AENN	All k-edited nearest neighbors
BAG	Bagging
BBNR	Blame based noise reduction
BiRNN	Bi-directional recurrent neural network
BLSTM	Bi-directional long short-term memory
BST	Stochastic gradient boosting
CNN	Condensed nearest neighbors
DT	Decision tree
KNN	K-nearest neighbor
NB	Naïve Bayes
RF	Random forest
SVM	Support vector machine

References

1. World Health Organization. *Coronavirus disease 2019 (COVID-19): Situation Report*; World Health Organization: Geneva, Switzerland, 2020.
2. Hui, D.S.; Azhar, E.I.; Madani, T.A.; Ntoumi, F.; Kock, R.; Dar, O.; Ippolito, G.; Mchugh, T.D.; Memish, Z.A.; Drosten, C.; et al. The continuing 2019-nCoV epidemic threat of novel coronaviruses to global health—The latest 2019 novel coronavirus outbreak in Wuhan, China. *Int. J. Infect. Dis.* **2020**, *91*, 264–266. [CrossRef]
3. Li, Q.; Guan, X.; Wu, P.; Wang, X.; Zhou, L.; Tong, Y.; Ren, R.; Leung, K.S.M.; Lau, E.H.Y.; Wong, J.Y.; et al. Early transmission dynamics in Wuhan, China, of novel coronavirus–infected pneumonia. *N. Engl. J. Med.* **2020**, *382*, 1199–1207. [CrossRef]
4. Roush, S.; Fast, H.; Miner, C.E.; Vins, H.; Baldy, L.; McNall, R.; Kang, S.; Vundi, V. National Center for Immunization and Respiratory Diseases (NCIRD) support for modernization of the Nationally Notifiable Diseases Surveillance System (NNDSS) to strengthen public health surveillance infrastructure in the US. In Proceedings of the 2019 CSTE Annual Conference, Raleigh, NC, USA, 2–6 June 2019.
5. Abdi, M. Coronavirus disease 2019 (COVID-19) outbreak in Iran: Actions and problems. *Infect. Control Hosp. Epidemiol.* **2020**, *41*, 754–755. [CrossRef]
6. Boulos, M.N.K.; Geraghty, E.M. Geographical tracking and mapping of coronavirus disease COVID-19/severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) epidemic and associated events around the world: How 21st century GIS technologies are supporting the global fight against outbreaks and epidemics. *Int. J. Health Geogr.* **2020**, *19*, 1–12. [CrossRef]
7. Dong, E.; Du, H.; Gardner, L. An interactive web-based dashboard to track COVID-19 in real time. *Lancet Infect. Dis.* **2020**, *20*, 533–534. [CrossRef]
8. Worldometer. COVID-19 Coronavirus Pandemic. Available online: www.worldometers.info/coronavirus/ (accessed on 11 November 2021).
9. Hageman, J.R. The coronavirus disease 2019 (COVID-19). *Pediatric Ann.* **2020**, *49*, e99–e100. [CrossRef] [PubMed]

10. Huang, C.; Wang, Y.; Li, X.; Ren, L.; Zhao, J.; Hu, Y.; Zhang, L.; Fan, G.; Xu, J.; Gu, X.; et al. Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. *Lancet* **2020**, *395*, 497–506. [CrossRef]
11. Arksey, H.; O'Malley, L. Scoping studies: Towards a methodological framework. *Int. J. Soc. Res. Methodol.* **2005**, *8*, 19–32. [CrossRef]
12. Adhikari, S.P.; Meng, S.; Wu, Y.-J.; Mao, Y.-P.; Ye, R.-X.; Wang, Q.-Z.; Sun, C.; Sylvia, S.; Rozelle, S.; Raat, H.; et al. Epidemiology, causes, clinical manifestation and diagnosis, prevention and control of coronavirus disease (COVID-19) during the early outbreak period: A scoping review. *Infect. Dis. Poverty* **2020**, *9*, 1–12. [CrossRef] [PubMed]
13. Ritchie, H.; Ortiz-Ospina, E.; Beltekian, D. *Coronavirus (COVID-19) Testing*; University of Oxford: Oxford, UK, 2020.
14. Salathé, M.; Althaus, C.L.; Neher, R.; Stringhini, S.; Hodcroft, E.; Fellay, J.; Zwahlen, M.; Senti, G.; Battegay, M.; Wilder-Smith, A.; et al. COVID-19 epidemic in Switzerland: On the importance of testing, contact tracing and isolation. *Swiss Med. Wkly.* **2020**, *150*, w20225. [CrossRef] [PubMed]
15. Padula, W.V. Why only test symptomatic patients? Consider random screening for COVID-19. *Appl. Health Econ. Health Policy* **2020**, *18*, 333–334. [CrossRef]
16. Damaševičius, R.; Abayomi-Alli, O.; Maskeliūnas, R.; Abayomi-Alli, A. BiLSTM with data augmentation using interpolation methods to improve early detection of parkinson disease. In Proceedings of the 2020 Federated Conference on Computer Science and Information Systems, Sofia, Bulgaria, 6–9 September 2020; Polish Information Processing Society PTI: Warsaw, Poland, 2020; Volume 21, pp. 371–380.
17. Guimaraes, M.T.; Medeiros, A.G.; Almeida, J.S.; Martin, M.F.Y.; Damasevicius, R.; Maskeliunas, R.; Mattos, C.L.C.; Filho, P.P.R. An optimized approach to Huntington's Disease detecting via audio signals processing with dimensionality reduction. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–8.
18. Kadry, S.; Damasevicius, R.; Taniar, D.; Rajinikanth, V.; Lawal, I.A. U-Net Supported Segmentation of Ischemic-Stroke-Lesion from Brain MRI Slices. In Proceedings of the 2021 Seventh International Conference on Bio Signals, Images, and Instrumentation (ICBSII), Chennai, India, 25–27 March 2021; pp. 1–5.
19. Maqsood, S.; Damaševičius, R.; Maskeliūnas, R. Hemorrhage detection based on 3D CNN deep learning framework and feature fusion for evaluating retinal abnormality in diabetic patients. *Sensors* **2021**, *21*, 3865. [CrossRef] [PubMed]
20. Rajinikanth, V.; Kadry, S.; Damasevicius, R.; Taniar, D.; Rauf, H.T. Machine-learning-scheme to detect choroidal-neovascularization in retinal OCT image. In Proceedings of the 2021 Seventh International conference on Bio Signals, Images, and Instrumentation (ICBSII), Chennai, India, 25–27 March 2021; pp. 1–5. [CrossRef]
21. Nemati, M.; Ansary, J.; Nemati, N. Machine-Learning approaches in COVID-19 Survival analysis and discharge-time likelihood prediction using clinical data. *Gene Expr. Patterns* **2020**, *1*, 100074. [CrossRef] [PubMed]
22. Lalmuanawma, S.; Hussain, J.; Chhakchhuak, L. Applications of machine learning and artificial intelligence for Covid-19 (SARS-CoV-2) pandemic: A review. *Chaos Solitons Fractals* **2020**, *139*, 110059. [CrossRef] [PubMed]
23. Malki, Z.; Atlam, E.S.; Hassanien, A.E.; Dagnew, G.; Elhosseini, M.A.; Gad, I. Association between weather data and COVID-19 pandemic predicting mortality rate: Machine learning approaches. *Chaos Solitons Fractals* **2020**, *138*, 110137. [CrossRef]
24. Kavadi, D.P.; Patan, R.; Ramachandran, M.; Gandomi, A.H. Partial derivative Nonlinear Global Pandemic Machine Learning prediction of COVID 19. *Chaos Solitons Fractals* **2020**, *139*, 110056. [CrossRef] [PubMed]
25. Amar, L.A.; Taha, A.A.; Mohamed, M.Y. Prediction of the final size for COVID-19 epidemic using machine learning: A case study of Egypt. *Infect. Dis. Model.* **2020**, *5*, 622–634. [CrossRef]
26. Goodman-Meza, D.; Rudas, A.; Chiang, J.N.; Adamson, P.C.; Ebinger, J.; Sun, N.; Botting, P.; Fulcher, J.A.; Saab, F.G.; Brook, R.; et al. A machine learning algorithm to increase COVID-19 inpatient diagnostic capacity. *PLoS ONE* **2020**, *15*, e0239474. [CrossRef] [PubMed]
27. Ozturk, T.; Talo, M.; Yildirim, E.A.; Baloglu, U.B.; Yildirim, O.; Acharya, U.R. Automated detection of COVID-19 cases using deep neural networks with X-ray images. *Comput. Biol. Med.* **2020**, *121*, 103792. [CrossRef] [PubMed]
28. Khan, M.A.; Alhaisoni, M.; Tariq, U.; Hussain, N.; Majid, A.; Damaševičius, R.; Maskeliūnas, R. COVID-19 Case recognition from chest CT images by deep learning, entropy-controlled firefly optimization, and parallel feature fusion. *Sensors* **2021**, *21*, 7286. [CrossRef]
29. Rehman, N.-U.; Zia, M.S.; Meraj, T.; Rauf, H.T.; Damaševičius, R.; El-Sherbeeny, A.M.; El-Meligy, M.A. A Self-Activated CNN Approach for Multi-Class Chest-Related COVID-19 Detection. *Appl. Sci.* **2021**, *11*, 9023. [CrossRef]
30. Rustam, F.; Reshi, A.A.; Mehmood, A.; Ullah, S.; On, B.-W.; Aslam, W.; Choi, G.S. COVID-19 future forecasting using supervised machine learning models. *IEEE Access* **2020**, *8*, 101489–101499. [CrossRef]
31. Wieczorek, M.; Siłka, J.; Połap, D.; Woźniak, M.; Damaševičius, R. Real-time neural network based predictor for cov19 virus spread. *PLoS ONE* **2020**, *15*, e0243189. [CrossRef] [PubMed]
32. Wieczorek, M.; Siłka, J.; Woźniak, M. Neural network powered COVID-19 spread forecasting model. *Chaos Solitons Fractals* **2020**, *140*, 110203. [CrossRef] [PubMed]
33. Ahouz, F.; Golabpour, A. Predicting the incidence of COVID-19 using data mining. *BMC Public Health* **2021**, *21*, 1087. [CrossRef]
34. Zivkovic, M.; Bacanin, N.; Venkatachalam, K.; Nayyar, A.; Djordjevic, A.; Strumberger, I.; Al-Turjman, F. COVID-19 cases prediction by using hybrid machine learning and beetle antennae search approach. *Sustain. Cities Soc.* **2021**, *66*, 102669. [CrossRef]

35. Alyasseri, Z.A.A.; Al-Betar, M.A.; Abu Doush, I.; Awadallah, M.A.; Abasi, A.K.; Makhadmeh, S.N.; Alomari, O.A.; Abdulkareem, K.H.; Adam, A.; Damasevicius, R.; et al. Review on COVID-19 diagnosis models based on machine learning and deep learning approaches. *Expert Syst.* **2021**, e12759. [CrossRef] [PubMed]
36. Kumar, V.; Singh, D.; Kaur, M.; Damaševičius, R. Overview of current state of research on the application of artificial intelligence techniques for COVID-19. *PeerJ Comput. Sci.* **2021**, *7*, e564. [CrossRef]
37. Ahmad, A.; Garhwal, S.; Ray, S.K.; Kumar, G.; Malebary, S.J.; Barukab, O.M. The number of confirmed cases of COVID-19 by using machine learning: Methods and challenges. *Arch. Comput. Methods Eng.* **2021**, *28*, 2645–2653. [CrossRef] [PubMed]
38. Devaraj, J.; Elavarasan, R.M.; Pugazhendhi, R.; Shafiullah, G.; Ganesan, S.; Jeysree, A.K.; Khan, I.A.; Hossain, E. Forecasting of COVID-19 cases using deep learning models: Is it reliable and practically significant? *Results Phys.* **2021**, *21*, 103817. [CrossRef] [PubMed]
39. Yaman, E.; Subasi, A. Comparison of bagging and boosting ensemble machine learning methods for automated EMG signal classification. *BioMed Res. Int.* **2019**, *2019*, 9152506. [CrossRef]
40. Friedman, J.H. Stochastic gradient boosting. *Comput. Stat. Data Anal.* **2002**, *38*, 367–378. [CrossRef]
41. Schuster, M.; Paliwal, K.K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [CrossRef]
42. Cortes, C.; Vapnik, V.N. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
43. Ho, T.K. Random decision forests. In Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, Canada, 14–16 August 1995; Volume 1, pp. 278–282. [CrossRef]
44. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [CrossRef]
45. Sáez, J.A.; Galar, M.; Luengo, J.; Herrera, F. INFFC: An iterative class noise filter based on the fusion of classifiers with noise sensitivity control. *Inf. Fusion* **2016**, *27*, 19–32. [CrossRef]
46. Prati, R.C.; Luengo, J.; Herrera, F. Emerging topics and challenges of learning from noisy data in nonstandard classification: A survey beyond binary class noise. *Knowl. Inf. Syst.* **2018**, *60*, 63–97. [CrossRef]
47. García-Gil, D.; Luengo, J.; García, S.; Herrera, F. Enabling smart data: Noise filtering in big data classification. *Inf. Sci.* **2019**, *479*, 135–152. [CrossRef]
48. Tomek, I. An experiment with the edited nearest-neighbor rule. *IEEE Trans. Syst. Man Cybern.* **1976**, *6*, 448–452. [CrossRef]
49. Delany, S.J.; Cunningham, P. An analysis of case-base editing in a spam filtering system. In *Advances in Case-Based Reasoning*; Springer: Berlin/Heidelberg, Germany, 2004; Volume 204, pp. 128–141.
50. Hart, P. The condensed nearest neighbor rule (Corresp.). *IEEE Trans. Inf. Theory* **1968**, *14*, 515–516. [CrossRef]
51. Goyal, S. Predicting the defects using stacked ensemble learner with filtered dataset. *Autom. Softw. Eng.* **2021**, *28*, 1–81. [CrossRef]
52. Jiménez, F.; Sánchez, G.; Palma, J.; Sciavicco, G. Three-objective constrained evolutionary instance selection for classification: Wrapper and filter approaches. *Eng. Appl. Artif. Intell.* **2021**, *107*, 104531. [CrossRef]
53. Sáez, J.A.; Corchado, E. ANCES: A novel method to repair attribute noise in classification problems. *Pattern Recognit.* **2022**, *121*, 108198. [CrossRef]