Article

Prediction of COVID-19 Risk in Public Areas Using IoT and Machine Learning

Ersin Elbasi , Ahmet E. Topcu and Shinu Mathew

College of Engineering and Technology, American University of the Middle East, Egaila 54200, Kuwait; ahmet.topcu@aum.edu.kw (A.E.T.); shinu.mathew@aum.edu.kw (S.M.)

* Correspondence: ersin.elbasi@aum.edu.kw; Tel.: +965-65-75-2754

Abstract: COVID-19 is a community-acquired infection with symptoms that resemble those of influenza and bacterial pneumonia. Creating an infection control policy involving isolation, disinfection of surfaces, and identification of contagions is crucial in eradicating such pandemics. Incorporating social distancing could also help stop the spread of community-acquired infections like COVID-19. Social distancing entails maintaining certain distances between people and reducing the frequency of contact between people. Meanwhile, a significant increase in the development of different Internet of Things (IoT) devices has been seen together with cyber-physical systems that connect with physical environments. Machine learning is strengthening current technologies by adding new approaches to quickly and correctly solve problems utilizing this surge of available IoT devices. We propose a new approach using machine learning algorithms for monitoring the risk of COVID-19 in public areas. Extracted features from IoT sensors are used as input for several machine learning algorithms such as decision tree, neural network, naïve Bayes classifier, support vector machine, and random forest to predict the risks of the COVID-19 pandemic and calculate the risk probability of public places. This research aims to find vulnerable populations and reduce the impact of the disease on certain groups using machine learning models. We build a model to calculate and predict the risk factors of populated areas. This model generates automated alerts for security authorities in the case of any abnormal detection. Experimental results show that we have high accuracy with random forest of 97.32%, with decision tree of 94.50%, and with the naïve Bayes classifier of 99.37%. These algorithms indicate great potential for crowd risk prediction in public areas.

Keywords: COVID-19; Internet of Things; machine learning; crowd analysis

Citation: Elbasi, E.; Topcu, A.E.; Mathew, S. Prediction of COVID-19 Risk in Public Areas Using IoT and Machine Learning. *Electronics* **2021**, *10*, 1677. <https://doi.org/10.3390/electronics10141677>

Academic Editor: Rui L. Aguiar

Received: 17 May 2021

Accepted: 1 July 2021

Published: 14 July 2021

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

Copyright: © 2021 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 COVID-19 pandemic is impacting many areas, including technology, education, the economy, and social life. COVID-19 has caused increased morbidity around the world. Front-liners are fighting hard to eradicate this pandemic. People in crowds and public areas easily spread this pandemic. A globally accepted strategy to decrease its spread is social distancing. The World Health Organization (WHO) recommends preventive measures such as social distancing and masks and gloves to limit the transmission of the virus from person to person. However, checking or controlling these strategies is inefficient to ensure that a public area is risk-free. Authorities must take extra measures to know about a crowded place or obtain statistics on people who wear masks or practice social distancing [1]. Machine learning (ML) is a part of artificial intelligence that enables a system to make individual decisions based on trained data models. Currently, computer techniques are used to detect pathogens in the human body. There is an urgent need to find a way to implement the preventive measures needed to limit the spread of the coronavirus. This study presents a system that recognizes interpersonal distances in crowds within a community or a queue in any organization to detect whether a person is wearing personal protective equipment such as a surgical mask and to detect any high body temperatures in

a crowded place. This makes it possible to perform risk prediction in public areas. This study also proposes the application of the Internet of Things (IoT) to collect all these input data to be processed by ML algorithms for classifying and predicting the risk based on the given data objects [2].

Artificial intelligence-based solutions are currently applied for many biomedical health problems and complications such as detection of brain tumors or breast cancer. The existing ML technology can reveal image features that are not visible in the original images [3]. Moreover, the availability of existing data is enough to allow training of the ML models to predict a correct result. This study uses multiple datasets to compare ML algorithms' performances to predict risk factors. The analysis focuses on prevention measures such as wearing masks and social distancing, which are more likely to influence overall risk. The success of risk prediction depends on how data are collected from various places. It is recommended to use high-performance IoT sensors and cameras to collect the data required for ML algorithms to learn and understand so that the prediction will be more correct [2]. In addition, IoT-based ML systems can also minimize the risk of healthcare staff being infected by direct contact with COVID-19 patients. IoT-based health monitoring systems can maintain real-time communication with patients, minimizing contact with healthcare staff. IoT-based deep learning models can also be used for early assessment of COVID-19 by diagnosing infections from the chest X-rays of patients [4]. For instance, Choyon et al. [5] used Arduino or Raspberry Pi-based IoT hardware connected to a camera, touchscreen display, wearable physiological and biological sensors, and temperature sensors to collect patient data and feed the data into a cloud network with a powerful server running ML algorithms that can be accessed by healthcare professionals or government authorities. Embedded sensors attached to the IoT hardware can detect temperature, blood pressure, heartbeat, etc., whereas Raspberry Pi can identify different types of data before they are fed into the cloud server. The data received by the server will then be used for training the ML model and prediction.

Currently, many kinds of research on COVID-19 are using the concept of ML and other computer technologies. Some investigations focus on applying ML in detecting COVID-19, whereas others explore prediction of COVID-19 and risk analysis [6]. Various approaches are used to collect data in these studies, such as predefined data from WHO websites or research portals or real-time data collected from hospitals or people. The core of such research is constituted by data from various resources and ML models are designed and trained using these datasets. Different algorithms play roles in the data analysis, modeling, and prediction processes of these studies. Some algorithms cannot provide accurate results, but the present study finds the best algorithms that can provide more accurate results by applying different ML algorithms to be used along with IoT applications to process and provide valuable predictions. This comparison between popular ML algorithms will be a stepping stone for developers to implement an application based on the algorithms' performances in this research. This study also recommends that application developers collect data through IoT devices and process those data in cloud-based servers to make predictions in real-time environments.

This paper is organized as follows. A brief introduction to COVID-19 data applications using IoT and ML algorithms is presented in Section 2. Section 3 describes the data collection, architecture design, and classification methods. Experimental results are presented in Section 4 and conclusions in Section 5.

2. Literature Review

The IoT is a system of devices connected to each other through a network, enabling data transfer automatically among people or objects. The IoT connects different types of devices that can share data efficiently, consisting of sensors and cameras to read data (e.g., temperature, humans, and faces) and networks to send the categorized data to a cloud server, which runs the ML engine. Data analysis and data monitoring are the other aspects of IoT for making real-time decisions based on the input data. The IoT contributes

to data mining processes and ML, but the main challenge is the difficulty in processed data reaching the server through the internet. In [7], a study to assess various ML methods to process IoT data in a smart city environment was presented, using different ML algorithms and further data processing tasks for analyzing data on traffic, weather, citizens, the environment, and other topics. These data were then clustered and optimized to predict traffic, weather, travel patterns, emergencies and anomalies, etc. The study then combined classification and regression ML algorithms to classify the data.

IoT devices are considered as things that can be used to collect various data. IoT sensors collect human data (positions, usage of masks, temperature, etc.) in indoor or outdoor spaces such as schools, streets, playgrounds, offices, shopping malls, and other public areas. Data collection is one of the basic tasks, and the IoT can contribute to ML. The devices or things consist of sensors, embedded systems, and cameras to collect data and transfer these data to a ML engine running in a cloud environment to perform prediction tasks. The communication between these IoT devices and the ML engine can be achieved using mobile networks or wireless networks. The K-means clustering algorithm can group the received data from IoT sensors [8].

Adam et al. [9] investigated the role of ML in IoT networks and discussed resource management solutions for wireless IoT. IoT networks support both long-range and short-range communications. Their study recommended that the IoT network in ML environments be scalable, adaptive, and efficient and have a fair network management method. Network resource management is another vital point that needs to be considered in IoT networks. This study emphasized that applying a ML algorithm efficiently can control network resources in an IoT network.

In ML approaches, the learning process is divided into three main parts: supervised learning, unsupervised learning, and reinforced learning. Supervised learning systems use trained data, where the output should meet the requirement of an input. Unsupervised learning systems do not require the desired result, whereas reinforced learning systems learn from external resources and make decisions accordingly [9]. Data need to be characterized into multiple categories to analyze and recognize safe or unsafe zones. ML applications mainly focus on these data classifications, which are discussed and analyzed in the present paper. Different ML algorithms are used to achieve this goal.

Figure 1 shows the general view of the system and data flow, with data collection through IoT sensors in two different crowded areas, area 1 and area 2. Data are then sent to a cloud server. The collected data will be analyzed using clustering, classification, and prediction algorithms by the ML server. The obtained results will be stored in a web server for public access through an HTTP connection. Finally, the client end will be able to see the area-wise risk factors.

Figure 1. General structure of data analysis and risk prediction.

The study conducted by Wang et al. [10] investigated the potential of a deep learning model for automatic COVID-19 diagnosis by chest computed tomography (CT) volumes.

They designed a high-performance COVID-19 diagnosis system with the help of ML classification algorithms and reported that the system requires minimal interpretation by doctors and that it is easy to train the model with artificial software. Built CT volumes were used to train the ML model, which helps predict COVID-19 probabilities using a deep classification network known as DeCoVNet. Research was also conducted in [7] using the Bayes and random forest algorithms to classify 3000 COVID-19 related posts. These data were collected from a Chinese social media application called Sina Weibo. The collected data were then classified into seven types of situational information to help authorities predict the scale of the spread, sense the public's mood, and understand the existing situation. The authors claimed that ML algorithms are powerful enough to handle massive data and produce accurate results.

There are different methods used in various studies to mitigate the spread of COVID-19. Most studies have tried to focus on social distancing strategies with the help of distinct types of monitoring devices such as crowd detection sensors and image recognition methods with ML algorithms. However, the prediction of high-risk and low-risk areas is crucial in such scenarios as the system should efficiently read and classify the input data and predict the outcome. In some places, though the crowd might be huge, 95% of the people may wear masks and gloves and practice proper social distancing. Still, if numbers of people are considered as input data to the system, the prediction will not be successful as it will predict such areas as being high-risk. The work presented in [11] exploits cellular network functionalities to detect if a specific region is at-risk or not. This approach assumes that everyone is carrying cellular mobile devices or other user equipment that can serve as a human tracker. The prediction of at-risk areas is based on the mobility of people. Higher mobility is considered to reflect higher risk. However, the system prediction depends on cellular mobile devices, the presence of which cannot be guaranteed. Another fact that needs to be considered for these systems is that the prediction is not based on social distancing or how many precautions are applied. It is strongly recommended to take safety precautions to reduce the risk of being in a public place more than banning events or implementing lockdowns, which will not be practical for more extended periods.

The efficiency of the ML approach in various studies depends on the number of trained models and the algorithms' accuracies. In [12], Sethi et al. used neural networks for medical image classification to identify different lung diseases and predict pneumonia. They used 100,000 X-ray images to train the model to increase the accuracy of COVID-19 prediction. This research proved that a ML algorithm with an efficiently trained model could successfully predict the results. Their accuracy in their results was 99% for all models tested in the study.

IoT networks consist of cameras and sensors to detect human faces and their locations. The prototype built in [13] utilized various IoT sensors to detect human faces and identify if people were wearing masks or not. The authors claimed that the sensors could clearly sense the face region regardless of size, shape, and color. Furthermore, a large database of models was prepared to achieve this task. The distance between people can also be calculated as the system detects human faces or human bodies. Similarly, IoT temperature sensors can sense an individual's temperature and send the data to ML software for a perfect prediction of high-risk or minimal-risk regions.

Researchers implemented a deep learning model for biometric face detection frameworks for cities under lockdown during the COVID-19 pandemic. This research aimed to detect human faces using IoT-based sensors to control public movement. It was found that the IoT devices used in the implementation were highly vulnerable to security issues. The researchers discussed the issue of cybercrime and recommended some data security solutions such as device authentication, secured cloud storage, secured internet connections, regular software updates, and encrypted security layers for the IoT sensors [14]. It is essential to ensure cloud platform security for the exchange of data as these systems handle large amounts of personal data, including individual images.

Performance of data analysis and prediction in ML are based on different algorithms. These algorithms are mainly classified into three categories. First, supervised algorithms are algorithms that will adjust the input variables to control the output. Second, unsupervised algorithms need no specific goal for achieving a target outcome. They manage and cluster all input data for different subsets or groups. Third, reinforced algorithms are trained in making decisions or predictions. Algorithms can learn by themselves by experience with the success or failure of output [15]. The K-means algorithm is an unsupervised algorithm used for forming data clusters. K in this algorithm is an input for the algorithm. Based on the input value, the algorithm selects K number of real or imaginary locations. The data points neighboring this centroid location are then combined with its location and create a cluster. This process continues until the imaginary or real centroid location does not change [15].

The random forest algorithm can be described as a collection of decision trees, and recent studies have shown that the random forest algorithm is an efficient classification algorithm as it can classify and handle large amounts of data correctly. To organize large volumes of medical data, in [16], researchers executed a combination of K-means clustering and the random forest classification method. They also conducted a comparative analysis and found that the K-means algorithm showed better performance. While other algorithms require more time for processing, K-means can partition high volumes of data in comparatively less time. The same study also proved random forest's efficiency as the algorithm helps ML models learn more quickly compared to other algorithms by interpreting data value sets. First, a randomized training dataset is created to design a random data subset. These data subsets are known as decision trees. The study's experimental results showed that random forest can classify large amounts of data efficiently compared to the existing LC-KNN and RC-KNN algorithms.

Naïve Bayes is another popular data classification algorithm in ML and it uses probability theory in classification. This algorithm uses a set of probabilities by calculating the frequency and combination of values in a dataset. Several classification methods prove that the naïve Bayes algorithm provides the best area under the curve value compared to other algorithms. Studies also show that this algorithm has the best accuracy compared to the zero-R and decision tree algorithms. The main advantages of using the naïve Bayes algorithm are that it requires less training data, has simple computing, is easy to implement, offers time efficiency and efficiency in handling big data, can handle missing or incomplete data, is not sensitive to irrelevant features, and does not suffer from data noise [17].

Regression analysis in ML consists of the prediction of a continuous outcome variable. A neural network is considered an intelligent tool in ML, giving some desired output for the given input achieved through various training processes. A training set consists of pairs of input datasets and output sets. This method is the first successful approach introduced in deep learning, which requires minimal data preprocessing [12]. In previous studies [18,19], authors successfully analyzed the current trends of the transmission of COVID-19 in the world using regression algorithms and found an accuracy of approximately 93% in predicting the rise in cases for a period. Neural networks are also among the deep learning techniques used in ML imaging classification. In [12], Sethi et al. recommended deep learning systems to train a model in a study using a large volume of chest X-rays to pretrain the model, and they found that neural networks helped to learn about the images. Detailed knowledge was transferred to the data model. It was then possible to predict or diagnose cases based on the X-rays in the input data model [12].

ML models are mainly categorized as supervised and unsupervised models. Supervised models learn a function that directs input data toward output based on the trained input-output pairs. Regression and classification are the two subcategories of supervised models. Unsupervised ML models deal with unlabeled data, such as employee salaries without employee names and IDs. These data will then need to be grouped and segmented based on the behavior of the datasets using a clustering method [20]. Classification is a process to prepare input datasets by correcting anomalies, removing null values, and

finding patterns or relations for training a model to successfully predict the output. The most challenging part of ML classification is the possibility of missing or incorrect data, which may affect model training and classification. There are many classification algorithms used in ML to mine data and each algorithm has its own methods; some of them filter the data based on logic and perception while others use statistical and instance-based learning methods [21].

Jiang et al. proposed an automated mask detection method with better performance compared to other methods. The “Properly Wearing Masked Face Detection Dataset” was used with the proposed algorithm, which has 9205 images. The Squeeze and Excitation-YOLOv3 object detector algorithm gave very promising results, being more effective, robust, and efficient [22]. Castillo et al. proposed a model to forecast and predict COVID-19 using a combination of several different artificial intelligence algorithms. This model helps doctors and decision-makers decrease the spread of the virus with early diagnosis and treatment of patients. A neuro-hybrid epidemic prediction model was applied for data extraction and analysis. A recurrent network algorithm was also applied in high-level processing for rule extraction, which had high accuracy, robustness, and speed [23]. Hnatiuc et al. developed a system to recognize subjects based on gait posture, which gives information for subject identification. Data were collected from different age, gender, height, and weight groups via inertial measurement units. ML algorithms are used for data processing and analysis. Decision tree and KNN algorithms give more promising results at rates above 90% [24].

The decision tree algorithm can lead ML systems to many outputs based on a given set of inputs for making a prediction or decision. The algorithm can facilitate both prediction and classification in ML. Trees are formed with different hierarchical procedures that will help the system reach certain decisions, such as whether a patient is positive or negative. The algorithm mainly follows an induction process, by which the system will build a tree based on given inputs and a pruning process will remove the complexities of the existing tree [25].

3. Methodology

Today, data are derived from sensors, smart factories, and social media very quickly in different fields. For this reason, it is necessary to analyze data with different methods to gain full value from them. ML methods are a wise choice for interpreting data in the best, most accurate way.

The main property of ML is the ability to learn from experiences or examples. Significant research has shown that ML technology can predict expected outputs by analyzing all available sample inputs. Computer vision is another essential technique used to train a system to recognize human presence, masks, gloves, and interpersonal distance by incorporating these models' patterns. Risk prediction could be implemented based on computer vision data inputs and data analytic tools that use specific algorithms [26].

In this article, data from IoT sensors are analyzed with ML algorithms, results obtained in different ways are compared, and risk areas are determined using the best ML algorithms. This model will help reduce the number of COVID-19 cases by the calculation of risk parameters such as temperature, interpersonal distance, and use of personal protective equipment.

3.1. Data Models and Training

Modeling different data beforehand and then training with different ML methods is the first stage of the system. Classifying different values obtained from the data model is important. The system classifies its data into safe and unsafe categories based on interpersonal distance, body temperature, and presence of mask and gloves. These parameters can be measured using IoT devices and detected using different ML algorithms. The decision tree algorithm and naïve Bayes classifier are used to classify these data by automatically dividing them into multiple classifications. These algorithms can provide rules of classifications in a human-legible format [27]. The input data are analyzed by applying statistical

methods and relational expressions to make multiple predictions [9]. The random forest algorithm uses mathematical expressions between these predictions to achieve risk prediction in this work. The algorithm also provides a regression technique to randomly select data from among all the input data.

A data clustering technique is used to compare the input data and the trained data. First, it will identify a set of matched data objects as described in [10]. Secondly, data are partitioned into safe and unsafe datasets based on parameters such as temperature, interpersonal distance, and personal protective equipment (mask and gloves). The K-means classical clustering algorithm is used to achieve this by making multiple clusters, in which aggregated data instances in a cluster are the same but data objects in different clusters are different [28].

This paper proposes a system in which IoT sensors collect nonpersonal data such as the distance between human objects, temperature, and use of a mask; personal data such as person's image, age, or name are not necessary for the system to analyze and predict the risk. Hence, this paper ensures that the privacy of each individual is protected. In this work, we do not collect personal private data such as age, name, address, or image; therefore, there is no issue of privacy in data collection.

3.2. Architecture Design

The architecture, as depicted in Figure 2, consists of the following parts:

- **Data Collection:** IoT sensors are used to collect data, such as interpersonal distance and images. Initially, data are collected from IoT devices in order to incorporate them into the data procession operation.
- **Data Processing:** The collected data will be sent to the data processing module. There, they are initially classified and clustered. Some of the data are used for the training model. This module deals with organizing, modeling, and clustering the data using ML algorithms to predict the results. Different ML algorithms are used to evaluate the performance of different algorithms.
- **Machine Learning Engine Platform:** Different ML algorithms yield different results for risk parameters. The results are compared to obtain the best decision by the calculation of safety parameters.
- **Display:** Processed data from the data processing module are sent to the systems that need to take action about the results. Based on the received and processed data, details about places and their risk assessments denoted as safe or unsafe will be the output of the model.

Figure 2. Architecture of the model.

ML algorithms have been used in many areas, from biomedical data classification to self-driving cars [29,30]. IoT applications are prevalent nowadays because video cameras and hardware are tiny and cheaply available everywhere [31]. Devices connected to the internet collect information about moving objects to process and monitor them for security purposes in automated decision-making. IoT sensors collect several types of moving object data from crowds in both outdoor and indoor spaces [32,33]. In this work, we collected data for three different people's contacts and created a data vector for each crowd as follows:

$$V = [\{\text{Distance from person A to person B}\}, \{\text{Distance from person B to person C}\}, \{\text{Distance from person A to person C}\}, \{\text{Mask status of person A}\}, \{\text{Mask status of person B}\}, \{\text{Mask status of person C}\}, \{\text{Temperature of person A}\}, \{\text{Temperature of person B}\}, \{\text{Temperature of person C}\}, \{\text{CLASS: Safe or Unsafe}\}]$$

Here, A, B, and C represent three different people in the same crowd. In our research, we collected data using IoT sensors. Several ML algorithms are applied to predict, classify, and cluster crowd data vectors. We use 800 crowd data vectors $\{V_1, V_2, \dots, V_{800}\}$ for a total of 7200 data. Some of the data are used for training and others are used for testing depending on the algorithm that we apply. Some of the data are available from the Multidisciplinary University Research Initiatives (MURI) dataset [34]. In addition to MURI, other data are simulated and labeled using medical experts' support in the classification of data as safe or unsafe for physical distances in crowds. Each vector belongs to one of the two classes of safe and unsafe. Automatically understanding whether a group is safe or unsafe in terms of COVID-19 physical distancing rules is vital for reducing the spreading of the virus in public. Low-level processing is the first phase of the automated decision. First, objects and crowds should be detected, and then features should be extracted for the second phase. High-level processing is the second phase, where the feature vector is an input [35–37]. Data samples for each class is given in the below.

SAFE Class: distance $\{A, B\} = 0.8$; distance $\{B, C\} = 1.2$; distance $\{A, C\} = 1.3$; mask $\{A\} = 1$; mask $\{B\} = 1$; mask $\{C\} = 0$; temperature $\{A\} = 37$; temperature $\{B\} = 36.5$; temperature $\{C\} = 35.9$.

UNSAFE Class: distance $\{A, B\} = 0.6$; distance $\{B, C\} = 0.8$; distance $\{A, C\} = 0.5$; mask $\{A\} = 0$; mask $\{B\} = 1$; mask $\{C\} = 0$; temperature $\{A\} = 38.5$; temperature $\{B\} = 36.5$; temperature $\{C\} = 36.2$.

In this study, we evaluate data using high-level processing for classification and rule extraction. The decision tree (DT), neural network (NN), naïve Bayes classifier (NBC), support vector machine (SVM), Bayesian network classifier (BNC), locally weighted learning (LWL), logistic regression (LR), AdaBoost.M1 (ABM1), Hoeffding tree (HT), and random forest (RF) algorithms are used to classify the dataset. Comparing the building model's performances, LWL yields the best performance along with the NBC and RF algorithms. Test model performance is worst for LWL. Other models give much better performance for test modeling. Weka is used as open-source, freely available, and platform-independent software for implementation environments. Weka is a tool for ML algorithms written in Java and developed at the University of Waikato. In the present study, Weka is used for preprocessing, regression, classification, and clustering of data retrieved from the IoT [38].

This paper proposes a system in which IoT sensors collect nonpersonal data such as the distance between human objects, temperature, and mask usage; personal data such as a person's image, age, or name are not necessary for the system to analyze and predict risk. Hence, this paper ensures that individuals' privacy is protected. The distance dataset is taken from MURI research, while the mask and temperature datasets are simulated using the Weka data generator tool. There are two classes in the dataset, labeled based on WHO standards. IoT sensors in this system should be low-cost and can be easily connected to other hardware and software. The VL531x time of flight (ToF) sensor can be used to measure distance between human objects. A Raspberry I camera can also be included in the system to detect distances between individuals. LM34/TMP34, LM35/TMP35, or TMP36 temperature sensors are suitable for the proposed system with wider temperature

ranges. For mask detection, the computer vision-based Raspberry Pi single-board computer equipped with a camera module can be used.

Random forest, naïve Bayes classifier and decision tree algorithms (Algorithms 1–3) are given in the below [16,17,20].

Algorithm 1: Random Forest Algorithm

INPUT: Data set V ; ensemble size S ; subspace dimension n .
 FOR $s = 1$ to S DO
 Build a bootstrap sample $vs.$ from V by sampling $|V|$ data points with replacement.
 Select n features at random and reduce the dimensionality of $vs.$ accordingly
 Train a tree model M_s on $vs.$ without pruning
 END
 OUTPUT: Ensemble of tree models

Algorithm 2: Decision Tree Algorithm

INPUT: Data partition V , attribute list L , attribute selection list A
 Create a node N
 IF dataset in V are belongs to the same class C
 THEN return N
 Function attribute selection method to FIND best split attribute
 FOR each outcome i of splitting criterion
 V_i set for satisfying outcome
 IF V_i is empty
 THEN attach leaf with majority class in V
 ELSE
 Attach the node returned by function generate tree
 END
 OUTPUT: A decision tree

Algorithm 3: Naïve Bayes Classifier Algorithm

INPUT: Training data set V , attribute list $A = \{a_1, a_2, \dots, a_n\}$
 Read the training set V
 Calculate the mean and standard deviation of the predictor attributes for each class
 FOR each attribute
 Calculate the probability of all predictor attributes $A = \{a_1, a_2, \dots, a_n\}$
 END
 Calculate the probability for each class
 OUTPUT: A class of testing dataset

4. Experimental Results

There are several ML algorithms used for classification and rule extraction in this work. The decision tree (DT), neural network (NN), naïve Bayes classifier (NBC), support vector machine (SVM), Bayesian network classifier (BNC), locally weighted learning (LWL), logistic regression (LR), AdaBoost.M1 (ABM1), Hoeffding tree (HT), and random forest (RF) algorithms are used to classify the dataset. The neural network algorithm, decision tree, and AdaBoost.M1 give more than 93% accuracy based on the experiments [39]. The random forest and naïve Bayes classifiers result in more than 97% accuracy, and the support vector machine algorithm gives 85.75% accuracy for the same dataset. Table 1 shows the accuracy rates of these classification algorithms for crowd safety data taken from IoT sensors. For all classification algorithms, 67% of the data are used for training and the rest of the data are used for testing [40–42]. The study includes 800 crowd data, 542 of which belong to the safe class while 258 belong to the unsafe class. Sixty percent of the data are used for training and the rest are used for testing for all ML algorithms. Many of the works related

to prediction of COVID-19 from patient records have used a few ML algorithms regardless of performance accuracy, whereas this paper addresses the collection of real-time data with the help of IoT sensors to predict the risk of COVID-19 in public areas and analyze the performance of the leading ML algorithms to suggest the best algorithm based on performance accuracy in terms of prediction.

Table 1. Crowd data accuracy for machine learning classification algorithms.

Classification Algorithm	Correctly Classified	Incorrectly Classified
Decision Tree (DT)	94.50%	5.50%
Naïve Bayes Classifier (NBC)	99.37%	0.63%
Random Forest (RF)	97.32%	2.68%
Neural Network (NN)	93.33%	6.67%
Bayes Network Classifier (BNC)	91.50%	8.50%
Logistic Regression (LR)	85.62%	14.38%
Support Vector Machine (SVM)	85.75%	14.25%
Locally Weighted Learning (LWL)	86.25%	13.75%
AdaBoost.M1 (ABM1)	93.12%	6.88%
Hoeffding Tree (HT)	84.38%	15.62%

Table 2 and Figure 3 show the performance indicators for the applied ML algorithms for crowd data classification. “True positive” (*TP*) indicates the rate of correctly classified data and “false positive” indicates the rate of incorrectly classified data. The naïve Bayes classifier gives a 90.9% *TP* rate for the safe class and 78.2% *TP* rate for the unsafe class. The random forest and locally weighted learning algorithms have the lowest *TP* rate for the unsafe class, while neural network has the highest *TP* rate for the unsafe class. The *F-measure* is also considered as the combined accuracy of precision and recall for the given data:

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (1)$$

The Matthews correlation coefficient (MCC) measures the quality of ML algorithms for two-class problems. The MCC uses true positives and negatives and false positives and negatives. It measures the balance of accuracy between the classes and is calculated as follows:

$$\frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FN) \times (TP + FP) \times (TN + FN) \times (TN + FP)}} \quad (2)$$

If there is a correlation between the predicted and observed classifications of the dataset, the MCC value will be between -1 and $+1$. If the MCC value is $+1$, it means that there is perfect prediction between the classes. If the MCC value is -1 , it means that there is disagreement between the classes. Experimental results show that all algorithms tested in this study give excellent MCC values. This shows the quality of the data, which have been observed from IoT sensors. The receiver operating characteristics (ROC) curve is another important quality measurement for ML classification algorithms. It shows the general performance of the data for each class. To create the ROC curve area, *TP* and *FP* rates are needed. If the ROC area is 1, it shows the perfect quality of the ML classification algorithm. If the ROC area is 0.5, it shows the random (chance) accuracy. Table 2 shows that all algorithms have ROC area values of more than 0.9, which is very promising. Based on the ROC area, the neural network algorithm gives more accurate results than the other algorithms. The precision-recall curve (PRC) is used less frequently than ROC curves. If data are unbalanced and complex, however, it is better to use the PRC. Perfect datasets and classifications give PRC results of 1. In this case, the neural network and random forest algorithms give better PRC values than the other algorithms [43,44].

A decision tree is a flowchart-like structure in which each inner node shows an attribute such as distance from A to B, distance from A to C, distance from B to C, Mask

A, Mask B, Mask C, etc. Each leaf node represents a category label, which is SAFE or UNSAFE. The Hoeffding tree ML classification algorithm retrieves rules from complex data streams, being an incremental version of the decision tree algorithm. With the Hoeffding tree algorithm, nodes are expanded as soon as there is enough statistical evidence for that path. Crowd data classification, with data received from IoT sensors, is very suitable for both the decision tree and Hoeffding tree rule extraction algorithms. The random forest (RF) algorithm, meanwhile, is very successful at classification and regression. It uses several decision trees to predict rules from large datasets. Decision trees use the entire dataset for classification; on the other hand, the random forest algorithm selects specific features and records to classify data. Table 3 below shows all rules extracted from the decision tree, Hoeffding tree, and random forest algorithms. The random forest and Hoeffding tree algorithms produced more rules than the decision tree algorithm. The decision tree algorithm has longer paths than the other two algorithms. Another advantage of the random forest algorithm is running time complexity to build a model. The random forest running time is 0.02 s, the Hoeffding tree running time is 0.05 s, and the decision tree running time is 0.09 s with Weka [45].

Table 2. TP rate, FP rate, precision, recall, F-measure, MCC, ROC area, PRC area, and class for each machine learning algorithm.

Method	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
NBC	0.909	0.218	0.856	0.909	0.881	0.702	0.941	0.958	S
	0.782	0.091	0.857	0.782	0.818	0.702	0.941	0.931	U
BNC	0.962	0.152	0.900	0.962	0.930	0.825	0.947	0.977	S
	0.848	0.038	0.940	0.848	0.892	0.825	0.967	0.962	U
NN	0.989	0.000	1.000	0.989	0.995	0.987	0.998	0.999	S
	1.000	0.011	0.985	1.000	0.992	0.987	0.998	0.996	U
SVM	0.915	0.227	0.851	0.915	0.882	0.702	0.950	0.968	S
	0.773	0.085	0.864	0.773	0.816	0.702	0.950	0.934	U
LWL	0.983	0.309	0.819	0.983	0.894	0.727	0.982	0.988	S
	0.691	0.017	0.966	0.691	0.806	0.727	0.982	0.973	U
DT	0.968	0.088	0.940	0.968	0.954	0.886	0.982	0.979	S
	0.912	0.032	0.953	0.912	0.932	0.886	0.982	0.970	U
HT	0.887	0.218	0.853	0.887	0.870	0.676	0.946	0.962	S
	0.782	0.113	0.830	0.782	0.805	0.676	0.946	0.962	U
RF	0.962	0.319	0.819	0.983	0.894	0.727	0.982	0.988	S
	0.681	0.038	0.966	0.691	0.806	0.727	0.982	0.973	U
LR	0.915	0.227	0.851	0.915	0.882	0.702	0.948	0.966	S
	0.773	0.085	0.864	0.773	0.816	0.702	0.948	0.932	U
ABM1	0.949	0.097	0.933	0.949	0.941	0.855	0.976	0.985	S
	0.903	0.051	0.925	0.903	0.914	0.855	0.976	0.960	U

Figure 3. TP rate, FP rate, precision, recall, F-measure, MCC, ROC area, and PRC area for each algorithm.

Table 3. Extracted rules via DT, HT and RF algorithms.

Sample retrieved rules with decision tree algorithm.	
•	CONDITION: (Distance {A, B} ≤ 0.7) && (Distance {A, C} > 0.8) && (Mask {A, B} = 0) SET: Class = {Unsafe}
•	CONDITION: (Distance {A, B} ≤ 0.7) && (Distance {A, C} > 0.8) && (Mask {A, B} = 1) SET: Class = {Safe}
•	CONDITION: (Distance {A, B} > 0.7) && (Temp {A} ≤ 38) && (Mask {C} = 0) && (Distance {A, C} ≤ 0.9) SET: Class = {Unsafe}
•	CONDITION: (Distance {A, B} > 0.7) && (Temp {A} ≤ 38) && (Mask {C} = 1) && (Distance {A, C} > 0.9) SET: Class = {Safe}
•	CONDITION: (Distance {A, B} > 0.7) && (Temp {A} ≤ 37) && (Temp {B} ≤ 38) && (Mask {C} = 0) && (Distance {A, C} ≤ 0.9) SET: Class = {Unsafe}
•	CONDITION: (Distance {A, B} > 0.7) && (Temp {A} ≤ 37) && (Temp {B} ≤ 38) && (Mask {C} = 0) && (Distance {A, C} > 0.9) && (Mask {B} = 1) SET: Class = {Safe}
Sample retrieved rules with Hoeffding tree algorithm.	
•	CONDITION: (Distance {A, C} ≤ 0.8) && (Temp {B} ≤ 37) SET: Class = {Safe}
•	CONDITION: (Distance {A, C} ≤ 0.8) && (Distance {A, C} ≤ 1.2) && (Temp {B} ≥ 37) SET: Class = {Unsafe}
•	CONDITION: (Distance {A, B} > 0.7) && (Temp {B} ≤ 37) && (Distance {B, C} ≤ 0.7) SET: Class = {Safe}
•	CONDITION: (Distance {A, B} > 0.7) && (Temp {B} ≤ 37) && (Distance {B, C} > 0.8) SET: Class = {Unsafe}
•	CONDITION: (Temp {B} > 37) && (Distance {A, C} ≤ 1.5) && (Distance {B, C} ≥ 1.1) && (Temp {A} ≤ 37) SET: Class = {Safe}
•	CONDITION: (Distance {A, C} ≤ 1.2) && (Distance {B, C} > 0.8) && (Distance {A, B} ≤ 1.1) SET: Class = {Unsafe}
Sample retrieved rules with random forest algorithm.	
•	CONDITION: (Distance {A, B} < 0.75) && (Distance {A, C} ≥ 0.75) && (Temp {C} ≤ 37) && (Mask {B} = 0) SET: Class = {Unsafe}
•	CONDITION: (Distance {A, B} ≥ 0.75) && (Temp {A} < 37.5) && (Temp {B} < 38.5) && (Distance {B, C} ≤ 0.65) && (Mask {A} = 0) SET: Class = {Unsafe}
•	CONDITION: (Distance {A, B} ≥ 0.75) && (Temp {A} < 37.5) && (Temp {B} < 38.5) && (Distance {B, C} ≤ 0.65) && (Mask {A} = 1) SET: Class = {Safe}
•	CONDITION: (Distance {A, B} ≥ 0.75) && (Temp {A} < 37.5) && (Temp {B} < 38.5) && (Distance {B, C} ≥ 0.65) SET: Class = {Safe}
•	CONDITION: (Distance {A, B} ≥ 0.75) && (Temp {A} < 37.5) && (Temp {B} ≥ 38.5) && (Distance {A, C} < 1.25) && (Mask {A} = 0) SET: Class = {Unsafe}
•	CONDITION: (Distance {A, B} ≥ 0.75) && (Temp {A} < 37.5) && (Temp {B} ≥ 38.5) && (Distance {A, C} < 1.25) && (Mask {A} = 1) SET: Class = {Safe}

Table 4 shows error measurement values for each classification algorithm, including the kappa statistic (KS), relative absolute error (RAE), root mean squared error (RMSE), and root relative squared error (RRSE). The KS is a corrected measurement between observed and actual classes. Kappa is calculated as follows

$$\kappa = \frac{p_0 - p_e}{1 - p_e} \quad (3)$$

Here, p_e is the probability of change agreement and p_0 is observed agreement between rates. If there is an agreement between observed and expected values, then κ will be 1; otherwise, it will be ≤ 0 . Any value of κ greater than 0 shows that the results are better than chance. Table 4 shows that the kappa values for the neural network and random forest algorithms are very close to 1; on the other hand, the kappa values of the naïve Bayes classifier and Hoeffding tree are close to 0.5. The RAE measurement is used frequently in

ML and data mining applications. It is a measurement between the actual value and naïve model value:

$$RAE = \frac{\sqrt{\sum_{i=1}^n (A - B)^2}}{\sqrt{\sum_{i=1}^n B_i^2}} \quad (4)$$

Here, A reflects actual values and B expected values. Ideal RAE values are less than 1. Random forest and neural network give the best RAE values; on the other hand, locally weighted learning and the naïve Bayes classifier give less promising RAE values. Another metric, the $RRSE$, uses the average of actual values as a simple predictor. Ideal $RRSE$ values are close to 0. In this work, the best $RRSE$ value is that for the random forest algorithm at 0.0091, reflecting the minimum error value based on the $RRSE$ metric.

Table 4. Kappa, MAE, RMSE, RAE, and RRSE values for the machine learning algorithms.

	DT	NBC	RF	NN	BNC	LR	SVM	LWL	ABM1	HT
KS	0.9721	0.6995	0.9972	0.9871	0.8221	0.6987	0.6998	0.7037	0.8551	0.6748
RMSE	0.1170	0.3118	0.0045	0.0796	0.2695	0.2994	0.3775	0.3126	0.2511	0.3070
RAE	12.89%	37.20%	00.22%	02.78%	23.12%	36.29%	29.39%	46.24%	31.43%	36.85%
RRSE	23.76%	63.35%	00.91%	16.17%	54.75%	60.81%	76.68%	63.49%	51.01%	62.36%

There are several ways to compare ML algorithms, including accuracy, error rate, visualization, and time performance. Table 5 shows the time performance for building a model (training) and time performance for testing a model. Locally weighted learning takes only 0.01 s to build the model; on the other hand, the neural network algorithm needs 0.89 s to build the model. Testing the model usually takes less time than training. In this work, locally weighted learning takes 1.72 s, the naïve Bayes classifier takes 0.07 s, support vector machine takes 0.04 s, and other algorithms take only 0.01 s for 800 crowd data [46].

Table 5. Time performance of each algorithm.

Classification Algorithm	Time to Build a Model (Seconds)	Time to Test a Model (Seconds)
Decision Tree (DT)	0.09	0.01
Naïve Bayes Classifier (NBC)	0.02	0.07
Random Forest (RF)	0.02	0.01
Neural Network (NN)	0.89	0.01
Bayes Network Classifier (BNC)	0.06	0.01
Logistic Regression (LR)	0.07	0.01
Support Vector Machine (SVM)	0.14	0.04
Locally Weighted Learning (LWL)	0.01	1.72
AdaBoost.M1 (ABM1)	0.05	0.01
Hoeffding Tree (HT)	0.05	0.01

In ML, there are many different clustering algorithms. Several algorithms use similarity or distance measures between examples in the characteristic space, which is an excellent way to discover dense observations. In this research, several clustering algorithms are applied, including EM, K-means, density, filtered, and farthest first. The accuracy for correctly clustering data for each cluster (safe and unsafe) is given for each algorithm in Table 6. We applied a linear regression algorithm for prediction [36]. It works by estimating coefficients for a line or hyperplane that satisfactorily fits the training statistics. It is a very simple regression algorithm and fast to train, and it may have superb performance if the output variable for the data is a linear mixture of the inputs. The regression algorithm gives very promising results (90% accuracy) for testing data [37,39].

Table 6. Accuracy and build time for each cluster.

Cluster Algorithm	Accuracy in “Safe” Cluster (%)	Accuracy in “Unsafe” Cluster (%)	Accuracy (%)	Build Time (s)
EM	82.4	69.1	79.07	0.59
K-Means	94.2	89.6	93.05	0.04
Density	91.4	87.4	90.40	0.03
Filtered	85.9	89.2	86.72	0.02
Farthest First	74.9	78.6	75.83	0.02

Figure 4 shows the margin curve for the classification algorithms that we applied for the crowd dataset. A margin curve generates points illustrating the prediction margin. The margin is defined as the difference between the probability predicted for the actual class and the highest probability predicted for the other classes. The x -axis shows the curve value and the y -axis indicates the number of data. In this work, we used 800 crowd data. The margin is between -1 and $+1$. Incorrect classification yields values close to -1 and correct classification yields values close to $+1$ in Figure 4. In Figure 4a, based on the naïve Bayes classifier, some of the data have values close to -1 , which is incorrect classification, but most of the data have values close to $+1$, reflecting correct classification. BNC, LR, SVD, LWL, ABM1, and HT have behavior similar to that of the NBC algorithm. However, the DT, NN, and RF algorithms give more promising results according to the figure below because none of them generate values close to -1 .

Figure 4. Cont.

Figure 4. Margin curves for NBC, BNC, DT, LR, NN, SVM, LWL, ABM1, HT, and RF algorithms.

5. Conclusions

COVID-19 is a community-acquired infection with symptoms resembling influenza and bacterial pneumonia. It has negatively affected the entire world in terms of economy, social life, education, and technology. Many health organizations confirm that it is very important to maintain physical distance and wear a mask to reduce the risk of COVID-19. Fever is also among the signs and symptoms of COVID-19. In our study, we have proposed a new approach that uses ML algorithms to monitor the risk of COVID-19 in public spaces using physical distancing, the presence of masks, and temperature. Our model can also make decisions to calculate risk with ML algorithms by adding other risk parameters to the system. In this work, we have proposed a new approach for monitoring crowds using IoT technologies and machine learning algorithms to detect safe and unsafe areas. IoT sensors collect data from indoor and outdoor areas in the feature vector. There are several ML algorithms used for classification, clustering, and prediction. The decision tree algorithm gives 94.50%, the naïve Bayes classifier gives 99.37%, and the random forest algorithm gives 97.32% accuracy using the same training and testing dataset. Expectation maximization (EM) and K-means algorithms are applied for clustering, and the K-means algorithm gives better results for two clusters compared to other algorithms. Time performances for building the model (training) and testing the model also provide very promising results. Locally weighted learning takes only 0.01 s to build the model; on the other hand, the neural network algorithm needs 0.89 s to build the model. Testing models usually takes less time than training them, and in this work, locally weighed learning takes 1.72 s, the naïve Bayes classifier takes 0.07 s, support vector machine takes 0.04 s, and the other algorithms take only 0.01 s for 800 crowd data.

We achieved high validation rates for our model by calculating risks using datasets. We achieved classification accuracy of 99.37% using the naïve Bayes classifier. We used a fixed dataset in our model. However, people often move at certain speeds indoors. The limitation of this study is related to the accurate capture and use of large numbers of data from moving objects. We plan to add other ML algorithms such as deep learning techniques to improve our model in our future work.

Author Contributions: E.E., A.E.T. and S.M. were involved in the full process of producing this paper, including conceptualization, methodology, modeling, validation, visualization, and preparing the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

References

1. Community Mitigation Framework (CMF, 2020). Implementation of Mitigation Strategies for Communities with Local COVID-19 Transmission. Available online: <https://www.cdc.gov/COVID19> (accessed on 17 May 2021).
2. Andreas, A.; Mavromoustakis, C.X.; Mastorakis, G.; Mumtaz, S.; Batalla, J.M.; Pallis, E. Modified Machine Learning Technique for Curve Fitting on Regression Models for COVID-19 projections. In Proceedings of the IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Pisa, Italy, 14–16 September 2020; pp. 1–6.
3. Elaziz, M.A.; Hosny, K.M.; Salah, A.; Derwish, M.M.; Lu, S.; Sahlol, A.T. New machine learning method for image-based diagnosis of COVID-19. *PLoS ONE* **2020**, *15*, e0235187. [CrossRef]
4. Ahmed, I.; Ahmad, A.; Jeon, G. An IoT-based deep learning framework for early assessment of COVID-19. *IEEE Internet Things J.* **2020**. [CrossRef]
5. Choyon, M.M.S.; Rahman, M.; Kabir, M.M.; Mridha, M.F. IoT-based Health Monitoring & Automated Predictive System to Confront COVID-19. In Proceedings of the IEEE 17th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT, and AI (HONET), Charlotte, NC, USA, 14–16 December 2020; pp. 189–193. [CrossRef]
6. Abdulkareem, K.H.; Mohammed, M.A.; Salim, A.; Arif, M.; Geman, O.; Gupta, D.; Khanna, A. Realizing an effective COVID-19 diagnosis system based on machine learning and IOT in smart hospital environment. *IEEE Internet Things J.* **2021**. [CrossRef]
7. Li, L.; Zhang, Q.; Wang, X.; Zhang, J.; Wang, T.; Gao, T.L. Characterizing the propagation of situational information in social media during COVID-19 epidemic: A case study on Weibo. *IEEE Trans. Comput. Soc. Syst.* **2020**, *7*, 556–562. [CrossRef]
8. Hussain, F.; Hassan, S.; Hussain, R.; Hossain, E. Machine learning for resource management in cellular and IoT networks: Potentials, current solutions, and open challenges. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1251–1275. [CrossRef]
9. Adam, B.; Smith, I.F.C.; Asce, F. Reinforcement learning for structural control. *J. Comput. Civ. Eng.* **2008**, *22*, 133–139. [CrossRef]
10. Wang, X.; Deng, X.; Fu, Q.; Zhou, Q.; Feng, J.; Ma, H.; Liu, W.; Zheng, C. A weakly-supervised framework for COVID-19 classification and lesion localization from chest CT. *IEEE Trans. Med. Imaging* **2020**, *39*, 2615–2625. [CrossRef]
11. Alsaeedy, A.A.R.; Chong, E.K.P. Detecting regions at risk for spreading COVID-19 using existing cellular wireless network functionalities. *IEEE Open J. Eng. Med. Biol.* **2020**, *1*, 187–189. [CrossRef]
12. Sethi, R.; Mehrotra, M.; Sethi, D. Deep Learning based Diagnosis Recommendation for COVID-19 using Chest X-rays Images. In Proceedings of the 2nd International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 15–17 July 2020; pp. 1–4. [CrossRef]
13. Kolhar, M.; Al-Turjman, F.; Alameen, A.; Abualhaj, M.M. A three-layered decentralized IoT biometric architecture for city lockdown during COVID-19 outbreak. *IEEE Access* **2020**, *8*, 163608–163617. [CrossRef]
14. Khan, N.M.; Abraham, N.; Hon, M.; Guan, L. Machine Learning on Biomedical Images: Interactive Learning, Transfer Learning, Class Imbalance, and Beyond. In Proceedings of the IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), San Jose, CA, USA, 28–30 March 2019; pp. 85–90. [CrossRef]
15. Sidath, A. 10 Machine Learning Algorithms You Need to Know. 2017. Available online: <https://towardsdatascience.com/10-machine-learning-algorithms-you-need-to-know-77fb0055fe0> (accessed on 17 May 2021).
16. Saravana Kumar, S.; Manikandan, P. Medical big data classification using a combination of random forest classifier and KMeans clustering. *Int. J. Intell. Syst. Appl. Eng.* **2018**, *10*, 11–19. [CrossRef]
17. Wibawa, A.P.; Kurniawan, A.C.; Prawidya Murti, D.M.; Adiperkasa, R.P.; Putra, S.M.; Kurniawan, S.A.; Nugraha, Y.R. Naïve Bayes classifier for journal quartile classification. *Int. J. Recent Contrib. Eng. Sci. IT* **2019**, *7*, 91–99. [CrossRef]
18. Mishra, C.; Gupta, D.L. Deep machine learning and neural networks: An overview. *IAES Int. J. Artif. Intell.* **2017**, *6*, 66–73. [CrossRef]
19. Gambhir, E.; Jain, R.; Gupta, A.; Tomer, U. Regression Analysis of COVID-19 using Machine Learning Algorithms. In Proceedings of the 2020 International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 10–12 September 2020; pp. 65–71. [CrossRef]
20. Ławrynowicz, A.; Tresp, V. Introducing Machine Learning. In *Perspectives on Ontology Learning*; Lehmann, J., Voelker, J., Eds.; AKA Verlag Heidelberg/IOS Press: Amsterdam, The Netherlands, 2014; pp. 35–50.
21. Soofi, A.A.; Awan, A. Classification techniques in machine learning: Applications and issues. *J. Basic Appl. Sci.* **2017**, *13*, 459–465. [CrossRef]
22. Jiang, X.; Gao, T.; Zhu, Z.; Zhao, Y. Real-Time Face Mask Detection Method Based on YOLOv3. *Electronics* **2021**, *10*, 837. [CrossRef]
23. Castillo Ossa, L.F.; Chamoso, P.; Arango-López, J.; Pinto-Santos, F.; Isaza, G.A.; Santa-Cruz-González, C.; Ceballos-Marquez, A.; Hernández, G.; Corchado, J.M. A hybrid model for COVID-19 monitoring and prediction. *Electronics* **2021**, *10*, 799. [CrossRef]

24. Hnatiuc, M.; Geman, O.; Avram, A.G.; Gupta, D.; Shankar, K. Human signature identification using IoT technology and gait recognition. *Electronics* **2021**, *10*, 852. [\[CrossRef\]](#)
25. Data Flair. Available online: <https://data-flair.training/> (accessed on 17 May 2021).
26. Broad, W.J.A.I. Versus the Coronavirus. The New York Times. 2020. Available online: <https://www.nytimes.com/2020/03/26/science/ai-versus-the-coronavirus.html> (accessed on 17 May 2021).
27. Mohammad, S.M.; Mohammadreza, R.; Mohammadamin, B.; Peyman, A.; Payam, B.; Amit, P.S. Machine learning for Internet of things data analysis: A survey. *Digit. Commun. Netw.* **2018**, *4*, 161–175. [\[CrossRef\]](#)
28. Berkhin, P. A Survey of Clustering Data Mining Techniques. In *Grouping Multidimensional Data*; Kogan, J., Nicholas, C., Teboulle, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 25–71. [\[CrossRef\]](#)
29. Alimadadi, A.; Aryal, S.; Manandhar, I.; Munroe, P.B.; Joe, B.; Cheng, X. Artificial intelligence and machine learning to fight COVID-19. *Physiol. Genom.* **2020**, *52*. [\[CrossRef\]](#) [\[PubMed\]](#)
30. Sharma, K.; Nandal, R. A Literature Study on Machine Learning Fusion with IOT. In Proceedings of the 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 23–25 April 2019; pp. 1440–1445. [\[CrossRef\]](#)
31. Al-Hajri, M.I.; Ali, N.T.; Shubair, R.M. Classification of indoor environments for IoT applications: A machine learning approach. *IEEE Antennas Wirel. Propag. Lett.* **2018**, *17*, 2164–2168. [\[CrossRef\]](#)
32. Ghosh, N.; Maity, K.; Paul, R.; Maity, S. Outlier Detection in Sensor Data Using Machine Learning Techniques for IoT Framework and Wireless Sensor Networks: A Brief Study. In Proceedings of the International Conference on Applied Machine Learning (ICAML), Bhubaneswar, India, 25–26 May 2019; pp. 187–190. [\[CrossRef\]](#)
33. Wu, C.M.; Patil, P.; Gunaseelan, S. Comparison of Different Machine Learning Algorithms for Multiple Regression on Black Friday Sales Data. In Proceedings of the IEEE 9th International Conference on Software Engineering and Service Science (ICSSESS), Beijing, China, 23–25 November 2018; pp. 16–20. [\[CrossRef\]](#)
34. Elbasi, E.; Zuo, L.; Mehrotra, K.; Mohan, C.; Varshney, P. Control charts approach for scenario recognition in video sequences. *Turk. J. Elec. Eng.* **2005**, *13*, 303–310.
35. Ferrag, M.A.; Shu, L.; Choo, K.K.R. Fighting COVID-19 and Future Pandemics with the Internet of Things: Security and Privacy Perspectives. *IEEE/CAA J. Autom. Sin.* **2021**, *8*, 1477–1499. [\[CrossRef\]](#)
36. Topcu, A.E.; Rmis, A.M. Analysis and evaluation of the riak cluster environment in distributed databases. *Comput. Stand. Interfaces* **2020**, *72*, 103452. [\[CrossRef\]](#)
37. Elbasi, E.; Zreikat, A.; Ayanoglu, D.K. Determination of Patient's Hearing Sensitivity Using Data Mining Techniques. In Proceedings of the IEEE 12th International Conference on Application of Information and Communication Technologies (AICT), Almaty, Kazakhstan, 17–19 October 2018; pp. 1–6. [\[CrossRef\]](#)
38. Witten, I.H.; Frank, E.; Hall, M.A. *The WEKA Workbench. Online Appendix for Data Mining: Practical Machine Learning Tools and Techniques*, 4th ed.; Morgan Kaufmann: Cambridge, MA, USA, 2016.
39. Acar, M.A.; Tolun, M.R.; Elbasi, E. An Application of Data Mining and Knowledge Discovery Process in the Field of Natural Gas Exploration. In Proceedings of the IEEE 8th International Conference on Application of Information and Communication Technologies (AICT), Astana, Kazakhstan, 15–17 October 2014; pp. 1–6. [\[CrossRef\]](#)
40. Kuwil, F.H.; Shaar, F.; Topcu, A.E.; Murtagh, F. A new data clustering algorithm based on critical distance methodology. *Expert Syst. Appl.* **2019**, *129*, 296–310. [\[CrossRef\]](#)
41. Aggarwal, G.; Jain, S. Analysis of Genes Responsible for the Development of Cancer Using Machine Learning. In Proceedings of the 2019 Third International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, 10–11 January 2019; pp. 162–167. [\[CrossRef\]](#)
42. Oztürk, U. Comparison of Machine Learning Algorithms on Different Datasets. In Proceedings of the 26th Signal Processing and Communications Applications Conference (SIU), Izmir, Turkey, 2–5 May 2018; pp. 1–4. [\[CrossRef\]](#)
43. Elbasi, E. B-DCT based watermarking algorithm for patient data protection in IoMT. In Proceedings of the 2020 International Conference on Information Security and Cryptology (ISCTURKEY), Ankara, Turkey, 3–4 December 2020; pp. 1–4. [\[CrossRef\]](#)
44. Sugu, S. Building a distributed K-Means model for Weka using remote method invocation (RMI) feature of Java. *Concurr. Comp. Pract. E* **2019**, *31*. [\[CrossRef\]](#)
45. Wang, Y.; Meng, W.; Liu, Z.; Liu, Y.; Xue, H. Adaptive machine learning-based alarm reduction via edge computing for distributed intrusion detection systems. *Concurr. Comp. Pract. E* **2019**, *31*. [\[CrossRef\]](#)
46. Nguyen, C.T.; Saputra, Y.M.; Huynh, N.V.; Nguyen, N.T.; Khoa, T.V.; Tuan, B.M.; Nguyen, D.N.; Hoang, D.T.; Vu, T.X.; Dutkiewicz, E.; et al. A comprehensive survey of enabling and emerging technologies for social distancing—Part I: Fundamentals and enabling technologies. *IEEE Access* **2020**, *8*, 153479–153507. [\[CrossRef\]](#)