Biomedical Signal Processing and Control 62 (2020) 102149

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

journal homepage: www.elsevier.com/locate/bspc

An IoT-based framework for early identification and monitoring of COVID-19 cases

Mwaffaq Otoom [a,*], Nesreen Otoum [b], Mohammad A. Alzubaidi [a], Yousef Etoom [c,d,e], Rudaina Banihani [c,f]

[a] *Computer Engineering Department, Yarmouk University, Irbid, Jordan*
[b] *Software Engineering Department, University of Petra, Amman, Jordan*
[c] *Department of Pediatrics, Faculty of Medicine, University of Toronto, Toronto, Ontario, Canada*
[d] *Department of Pediatrics and Division of Pediatric Emergency Medicine, The Hospital for Sick Children, Sick Kids Research Institute, Toronto, Ontario, Canada*
[e] *Department of Pediatrics, St Joseph's Health Centre, Toronto, Ontario, Canada*
[f] *Department of Newborn and Developmental Pediatrics, Sunnybrook Health Science Centre, Toronto, Ontario, Canada*

ARTICLE INFO

Keywords:
COVID-19
Coronaviruses
Early identification or prediction
Internet of Things
Real-time monitoring
Treatment response

ABSTRACT

The world has been facing the challenge of COVID-19 since the end of 2019. It is expected that the world will need to battle the COVID-19 pandemic with precautious measures, until an effective vaccine is developed. This paper proposes a real-time COVID-19 detection and monitoring system. The proposed system would employ an Internet of Things (IoTs) framework to collect real-time symptom data from users to early identify suspected coronaviruses cases, to monitor the treatment response of those who have already recovered from the virus, and to understand the nature of the virus by collecting and analyzing relevant data. The framework consists of five main components: Symptom Data Collection and Uploading (using wearable sensors), Quarantine/Isolation Center, Data Analysis Center (that uses machine learning algorithms), Health Physicians, and Cloud Infrastructure. To quickly identify potential coronaviruses cases from this real-time symptom data, this work proposes eight machine learning algorithms, namely Support Vector Machine (SVM), Neural Network, Naïve Bayes, K-Nearest Neighbor (K-NN), Decision Table, Decision Stump, OneR, and ZeroR. An experiment was conducted to test these eight algorithms on a real COVID-19 symptom dataset, after selecting the relevant symptoms. The results show that five of these eight algorithms achieved an accuracy of more than 90 %. Based on these results we believe that real-time symptom data would allow these five algorithms to provide effective and accurate identification of potential cases of COVID-19, and the framework would then document the treatment response for each patient who has contracted the virus.

1. Introduction

Since its discovery in late December of 2019, there have been more than 14.5 million confirmed cases of COVID-19 reported in 185 countries, as of July 21, 2020 [1], with approximately a 2 % daily increase. Among these cases there have been more than 95 thousand deaths, which represents an approximate 4.2 % mortality rate. This novel coronavirus was characterized on March 11, 2020 as a pandemic by the World Health Organization [2]. Unfortunately, there is no successful treatment procedure or vaccine yet. It is expected that the development of an effective vaccine will take more than a year, especially since the nature of the virus has not yet been completely characterized [3].

Currently, the only way that the world can deal with this coronavirus is to slow down its spread, (i.e. "flatten the curve") by using measures such as social distancing, hand washing and face masks. However, technology could also help slow its spread, through early identification (or prediction) and monitoring of new cases [4,5]. Such technologies include big data, as well as cloud and fog capabilities [6], the use of data gathered through remote monitoring, such as mHealth, teleHealth, and real-time patient status follow-up [7].

This paper proposes a COVID-19 detection and monitoring system that would collect real-time symptom data from wearable sensor technologies. To quickly identify potential coronaviruses cases from this real-time data, this paper proposes the use of eight machine learning

[*] Corresponding author.
E-mail address: mof.otoom@yu.edu.jo (M. Otoom).

https://doi.org/10.1016/j.bspc.2020.102149
Received 18 April 2020; Received in revised form 22 July 2020; Accepted 7 August 2020
Available online 15 August 2020

algorithms, namely Support Vector Machine (SVM), Neural Network, Naïve Bayes, K-Nearest Neighbor (K-NN), Decision Table, Decision Stump, OneR, and ZeroR. This detection and monitoring system could be implemented with an IoT infrastructure that would monitor both potential and confirmed cases, as well as the treatment responses of patients who recover from the virus. In addition to real-time monitoring, this system could contribute to the understanding of the nature of the virus by collecting, analyzing and archiving relevant data.

The proposed framework consists of five main components: (1) real-time symptom data collection (using wearable devices), (2) treatment and outcome records from quarantine/isolation centers, (3) a data analysis center that uses machine learning algorithms, (4) healthcare physicians, and (5) a cloud infrastructure. The aim of this framework, is to reduce mortality rates through early detection, following up on recovered cases, and a better understanding of the disease.

This work conducts an experiment to test these eight machine learning algorithms on a real dataset. The results show that five of these eight algorithms achieved accuracies of more than 90 %. Using these five algorithms will provide effective and accurate prediction and identification of potential cases of COVID-19, based on real-time symptom data.

This paper is organized as follows. Section 2 reviews the relevant literature. Section 3 details the proposed framework, including the five components. Section 4 focuses on the identification (or prediction) of new cases, using machine learning algorithms. Lastly, Section 5 concludes the work.

2. Literature review

There is considerable work in the literature regarding the use of the Internet of Things (IoT) to deliver health services. Usak et al. conducted a systematic literature review of the use of IoT in health care systems. That work also included a discussion of the main challenges of using IoT to deliver health services, and a classification of the reviewed work in the literature [8].

Wu et al. proposed a hybrid IoT safety and health monitoring system. The goal was to improve outdoor safety. The system consists of two layers: one is used to collect user data, and the other to aggregate the collected data over the Internet. Wearable devices were used to collect safety indicators from the surrounding environment, and health signs from the user [9].

Hamidi studied authentication of IoT smart health data to ensure privacy and security of health information. The work proposed a biometric-based authentication technology [10].

Rath and Pattanayak proposed a smart healthcare hospital in urban areas using IoT devices, inspired by the literature. Issues such as safety, security and timely treatment of patients in VANET zone were discussed. Evaluation of the proposed system was conducted using simulators such as NS2 and NetSim [11].

Darwish et al. proposed a CloudIoT-Health paradigm, which integrates cloud computing with IoT in the health area, based on the relevant literature. The paper presented the challenges of integration, as well as new trends in CloudIoT-Health. These challenges are classified at three levels: technology, communication and networking, and intelligence [12].

Zhong and Li studied the monitoring of college students during their physical activities. The paper focused on a Physical Activity Recognition and Monitoring (PARM) model, which involves data pre-processing. Several classifiers, such as decision tree, neural networks, and SVM, were tested and discussed [13].

Din and Paul proposed an IoT-based smart health monitoring and management architecture. The architecture is composed of three layers: (1) data generation from battery-operated medical sensors and processing, (2) Hadoop processing, and (3) application layers. Because of the limited capacity of batteries to power the sensors, the work employed an energy-harvesting approach using piezoelectric devices attached to the human body [14].

Otoom et al. developed an IoT-based prototype for real-time blood sugar control. ARIMA and Markov-based statistical models were used to determine the appropriate insulin dose [21]. Alshraideh et al. proposed an IoT-based system for Cardiovascular Disease detection. Several machine learning algorithms were used for CVD detection [22].

Nguyen presented a survey of Artificial intelligence (AI) methods being used in the research of COVID-19. This work classified these methods into several categories, including the use of IoT [15]. Maghdid proposed the use of sensors available on smartphones to collect health data, such as temperature [16].

Rao and Vazquez proposed the use of machine learning algorithms to identify possible COVID-19 cases. The learning is done on collected data from the user through web survey accessed from smartphones [17]. Allam and Jones discussed the need to develop standard protocols to share information between smart cities in pandemics, motivated by the outbreak of COVID-19. For instance, AI methods can be applied to data collected from thermal cameras installed in smart cities, to identify possible COVID-19 cases [18]. Fatima et al. proposed an IoT-based approach to identify coronavirus cases. The approach is based on a fuzzy inference system [19]. Peeri et al. conducted a comparison between MERS, SARS, and COVID-19, using the available literature. They suggested the use of IoT in mapping the spread of the infection [20].

To our knowledge, no one has developed a complete framework for using IoT technology for the identification and monitoring of COVID-19.

3. Methods

3.1. IoT background

The Internet of things (IoT) uses communication and sensor technologies, as well as ubiquitous and pervasive computing, to upgrade physical objects into smart objects [31]. This enables the delivery of smart services to users, to improve the quality of their lives [32]. IoT architectures consist mainly of three layers: physical, network, and application [33]. Physical objects are equipped with sensors to collect heterogeneous data. These sensors have a limited computational capacity, and a limited lifetime. The more data that they collect, the more helpful decisions can be made. However, data processing complexity becomes a bottleneck [34]. Connectivity can be used to cope with the limited computational power of these sensors. Several different communication technologies have been employed, including 6LoWPAN, Bluetooth, IEEE 802.15.4, RFID and near-field communication (NFC) [31].

The network layer is not just used to upload collected data, for analysis. It is also used to facilitate communication between heterogeneous IoT objects, at the physical layer. In doing this, the network layer should support scalability, as the number of the objects increases, as well as device discovery, and context awareness. Significantly, it should also provide security and privacy for IoT devices [35]. The data uploaded from the IoT devices can be deeply analyzed, to generate insights and help make decisions. Currently, Machine Learning and Deep Learning (ML/DL) algorithms are used for this purpose, and are replacing more traditional methods because of their ability to deal with big data [36]. Al-Garadi et al. provided a thematic taxonomy of ML/DL used for IoT Security [37].

There are a wide range of applications where IoT can be effectively used, including healthcare, smart cities, smart buildings, agriculture, and power grids. In healthcare, IoT is sometimes called Internet of Medical Things (IoMT) [38]. It has largely displaced traditional ICT-based methods, such as telemedicine or telehealth. IoMT can provide more advanced features than these traditional methods. For example, while traditional methods can connect patients with medical doctors remotely, IoMT also supports machine-human and machine-machine interactions, such as AI-based diagnosis.

One important issue in designing IoMT is the balance between data

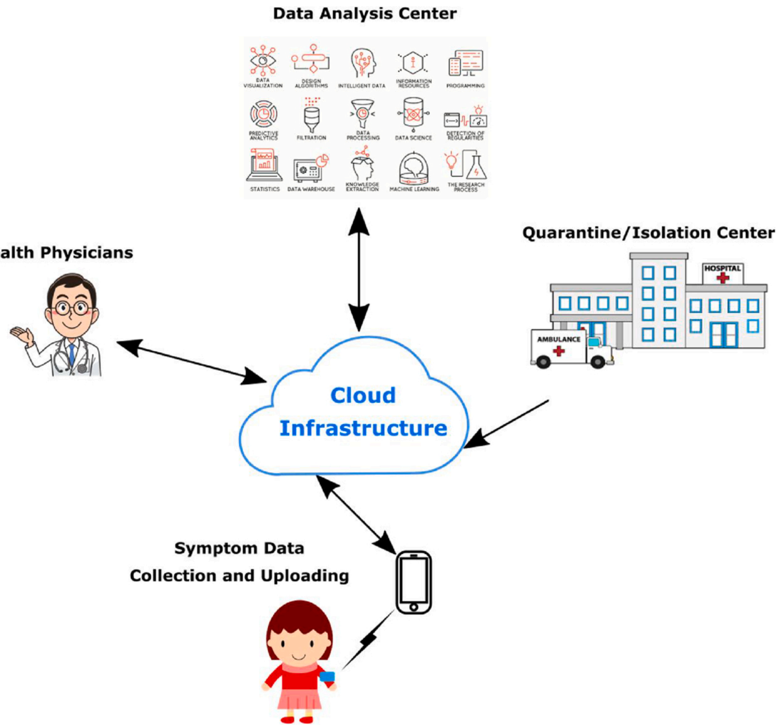


Fig. 1. Overall IoT-based Framework for Early Identification and Monitoring of Novel Human Coronaviruses.

privacy/security and patient safety [39]. Examples of cyber threats that challenge such designs are eavesdropping on communication channels (to sell the collected data), intervention, disruption, or even modification of the service. However, in cases where the patient's life is at risk, breaking some security measures to access the IoMT might be needed to save the patient's life [40]. ML/DL methods can be used to support this balance.

3.2. *The proposed IoT framework*

This section depicts and discusses our envisioned IoT-based framework, which could be used to monitor and identify (or predict) potential coronaviruses cases, in real time. Equally important, this framework could be used to predict the treatment response of confirmed cases, as well as to better understand the nature of the COVID-19 disease.

Fig. 1 shows the framework of our proposed IoT architecture. It consists of five main components: Symptom Data Collection and Uploading, a Quarantine/Isolation Center, a Data Analysis Center, an interface to Health Physicians, all of which are interconnected through a Cloud Infrastructure.

3.2.1. *Symptom data collection and uploading*

The aim of this component is to collect real-time symptom data through a set of wearable sensors on the user's body. In our earlier study [41], the most relevant COVID-19 symptoms were identified, based on a real COVID-19 patient dataset. These identified symptoms were: Fever, Cough, Fatigue, Sore Throat, and Shortness of Breath.

There are several biosensors available to detect these symptoms. For instance, temperature-based sensors can be used for the detection of Fever [23]. Cough and its classifications for different ages can be detected using audio-based sensors with acoustic and aerodynamic models [24]. Motion-based and heart-rate sensors can be used to detect Fatigue [25]. Sore Throat can be detected using image-based classification [26]. Finally, oxygen-based sensors can be used to detect Shortness of Breath [27].

Other relevant data – such as travel and contact history during the past 3–4 weeks, can be collected in an ad-hoc manner through mobile applications.

3.2.2. *Quarantine/isolation center*

This component collects data records from users who have been quarantined or isolated in a health care center. These records include both health (or technical) and non-technical data. For health (or technical) data, each record includes time-series data of the above-mentioned symptoms, while for non-technical data, each record includes travel and contact history during the past 3–4 weeks, chronic diseases, age, gender, and any other relevant information, such as family history of illness. Each record would eventually also include the treatment response for each case.

3.2.3. *Data analysis center*

The Data Center hosts data analysis and machine learning algorithms. These algorithms are used to build a model for COVID-19, and to provide a real-time dashboard of the processed data. The model could then be used to quickly identify or predict potential COVID-19 cases, based on real-time data collected and uploaded from users. The model can also predict the patient's treatment response. Over time, the disease models developed from this data will provide useful information about the nature of the disease.

3.2.4. *Health physicians*

Physicians will monitor suspected cases whose real-time uploaded symptom data indicates a possible infection by our proposed machine learning based identification/prediction model. The physicians will then be able to respond swiftly to these suspected cases by following up with any further clinical investigation needed to confirm the case. This allows the confirmed cases to be isolated and given appropriate health care.

3.2.5. *Cloud infrastructure*

The cloud infrastructure is interconnected through the Internet, and (1) allows upload of real-time symptom data from each user, (2) maintains personal health records, (3) communicates prediction results, (4) communicates physician recommendations, and (5) provides for storage of information.

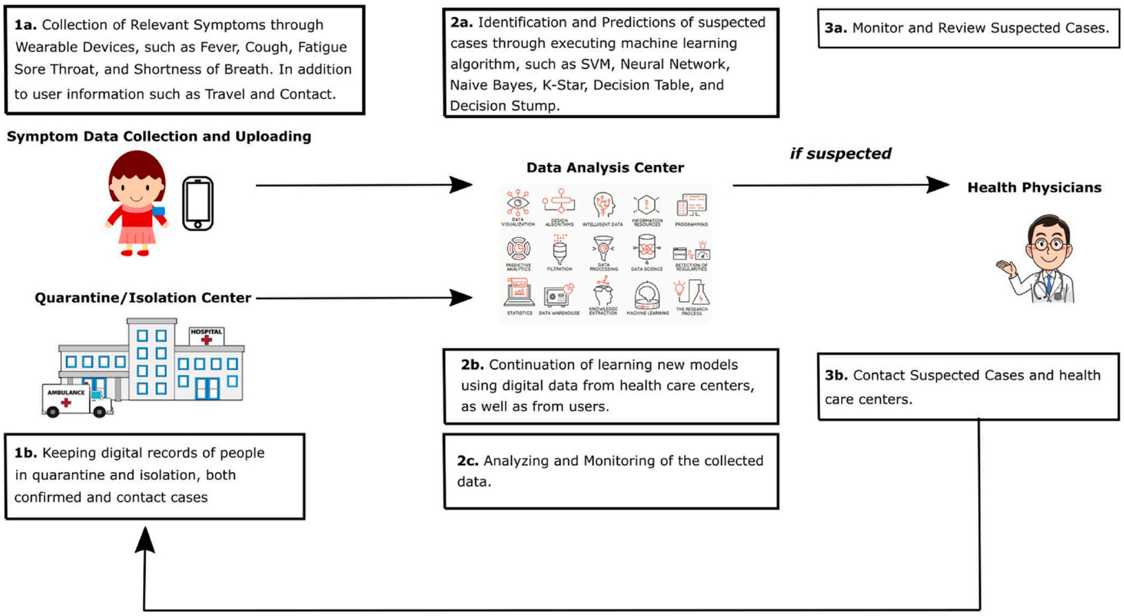

1a. Collection of Relevant Symptoms through Wearable Devices, such as Fever, Cough, Fatigue Sore Throat, and Shortness of Breath. In addition to user information such as Travel and Contact.
2a. Identification and Predictions of suspected cases through executing machine learning algorithm, such as SVM, Neural Network, Naive Bayes, K-Star, Decision Table, and Decision Stump.
3a. Monitor and Review Suspected Cases.
Symptom Data Collection and Uploading
Data Analysis Center
if suspected
Health Physicians
DATA VISUALIZATION
DESIGN ALGORITHMS
INTELLIGENT DATA
INFORMATION RESOURCES
PROGRAMMING
PREDICTIVE ANALYTICS
FILTRATION
DATA PROCESSING
DATA SCIENCE
DETECTION OF REGULARITIES
STATISTICS
DATA WAREHOUSE
KNOWLEDGE EXTRACTION
MACHINE LEARNING
THE RESEARCH PROCESS
Quarantine/Isolation Center
HOSPITAL
AMBULANCE
2b. Continuation of learning new models using digital data from health care centers, as well as from users.
3b. Contact Suspected Cases and health care centers.
1b. Keeping digital records of people in quarantine and isolation, both confirmed and contact cases
2c. Analyzing and Monitoring of the collected data.

Fig. 2. Flowchart of the proposed framework scenarios.

Fig. 2 presents the scenario (or workflow) employed by the framework, which can be described as follows.

1 The system non-invasively collects real-time user symptom data through wearable devices and sensors. Again, these symptoms are: Fever, Cough, Fatigue, Sore Throat, and Shortness of Breath. Further, the user submits information via a mobile application about living in (or travel to) infected areas, as well as possible contact with COVID-19 infected persons. The Quarantine/Isolation Center also periodically submits data from their isolated and quarantined patients who are housed in the center. The content of that data is similar to the real-time data collected from users.
2 The sensed symptom data are uploaded to the Data Analysis Center using a smartphone, through the Cloud Infrastructure. Digital records from the health care center are also regularly sent to the Data Analysis Center through the Cloud Infrastructure. The Data Analysis Center hosts machine learning algorithms, which use the data received from the health care center to continuously update its models. The models are then used to identify potential cases, based on the real-time symptom data from each user. The data center also analyzes all its data, and presents the results on a real-time dashboard. That dashboard can be informative to physicians about the nature of the virus.
3 If a potential case is identified, it will be sent to the relevant *physician* to follow up with the patient. The patient will then be called and encouraged to visit the health care center for clinical tests, such as the Polymerase Chain Reaction (PCR) test, which is used to identify positive cases. If it turns out that the case is confirmed, the patient can be isolated, and all contacts will be contacted and quarantined.

A complementary and integral component to this framework is the use of the same mobile application to educate users, by including useful information on how they can avoid illness, and how to avoid being exposed to the virus.

3.3. *Prediction of potential cases*

This section further discusses the predictive models, and the machine learning algorithms that will be employed in the Data Center component of the proposed IoT-based framework.

In particular, an experiment was conducted to investigate the possibility of using machine learning algorithms for quick identification (or prediction) of potential COVID-19 infections. The rest of this section describes that experimental setup, and presents and discusses the results.

3.3.1. *Dataset*

A dataset of 14251 confirmed COVID-19 cases from the COVID-19 Open Research Dataset (CORD-19) repository [28] was used. The data contains different types of information about each case. Our work focused on symptoms, travel history to suspicious areas, and contact history with potentially infected people. However, some of this information was missing for many of the cases documented within the database. Moreover, the data was not well structured for use by machine learning algorithms.

3.3.2. *Data preprocessing*

In our previous work [41], the data was preprocessed and structured to be better suited for machine learning. The cases with documented symptoms were collected. This resulted in a list of 80 symptoms. However, many of these symptoms were judged to be synonyms. Thus, the number of symptoms was reduced to 20. This merging of synonymous symptoms was done in an ad-hoc manner by two medical doctors, who are co-authors of this work. For example, “anorexia” and “loss of appetite” were merged together.

Our previous work also determined the relative importance of these 20 symptoms. The following six different statistically-based feature selection algorithms were employed in that work, to rank the 20 symptoms, based on their importance: *Spectral Score*, *Information Score*, *Pearson Correlation*, *Intra-Class Distance*, *Interquartile Range*, and our *Variance Based Feature Weighting* [41]. The first five of these methods had been proposed earlier in the literature [42]. The sixth method was a new one. It not only ranks the symptoms, but also assigns importance weights to each of them. It was found that the most important five symptoms (ordered from most important to least important) are: *Fever*, *Cough*, *Fatigue*, *Sore Throat*, and *Shortness of Breath*.

Based on the findings of that earlier work, this work uses those five most important symptoms. In addition, two extra features were added: *Live* and *Contact*. The first feature (*Live*) represents whether or not the person lived, travelled to, or passed by a potentially infected area. The second feature (*Contact*) represents whether or not the person was known to be in contact with a potentially infected person. This resulted in a preprocessed dataset of 1476 × 7 data records. Among which 854 of those records were from confirmed COVID-19 cases, and 622 records were for non-confirmed cases.

3.3.3. *Predictive model*

This work used this preprocessed dataset to build a predictive model for our identification (or prediction) system. The function of this model is to estimate the likelihood that a given person is infected by COVID-19.

Several learning algorithms (i.e. classifiers) could have been used for this purpose. Those classifiers can be categorized into multiple categories. WEKA Software [30], (which we used in this work) categorizes the classifiers into six categories: (1) function-based classifiers, such as Support Vector Machines, (2) lazy classifiers, such K-Nearest Neighbors, (3) Bayes based classifiers, such as Naïve Bayes, (4) rule-based classifiers, such as Decision Tables, ZeroR, and OneR, (5) tree-based classifiers, such as Decision Stump, and (6) meta classifiers, such Neural Networks.

In this work, at least one classifier from each category was selected. Specifically, this work compares the performance of eight machine learning algorithms: (1) Support Vector Machine (SVM), using Radial Basis Function (RBF) kernel, (2) Neural Network, (3) Naïve Bayes, (4) K-Nearest Neighbor (K-NN), (5) Decision Table, (6) Decision Stump, (7) OneR, and (8) ZeroR [29].

This work used WEKA Software to run all of these algorithms on our dataset [30]. The default parameter values were used for each of the eight algorithms. Below is a brief description of the eight algorithms:

3.3.3.1. Support Vector Machine (SVM). SVM is a supervised learning method. Given a set of training examples that are labeled (i.e. each instance in the training set either belongs to the positive or negative class), SVM learns the hyperplane that best separates the instances from each class, and maximizes the margin between the data instances and the hyperplane itself. This learnt hyperplane is then used to assign (or predict) a class label for any new test instance.

3.3.3.2. Artificial Neural Network (ANN). ANN is a supervised learning method. The learning process tries to mimic the learning that takes place inside the human brain. To do so, multiple layers of nodes are connected through edges. The edges connecting between the nodes are represented as numerical weights. The output of each node is computed as weighted sum of its inputs.

Given a set of training examples that are labeled (i.e. each instance either belongs to the positive or negative class), the ANN learns the numerical weights that best classify the instances from each class.

This learnt model is then used to assign (or predict) a class label for any given test instance. The test instance drives the inputs to the nodes of the first layer. Then a threshold is applied to the outputs of the final layer, to determine the label for that test instance.

3.3.3.3. Naïve Bayes. Naïve Bayes is a supervised learning method. The learning process follows a probabilistic approach. It uses Bayes theorem to compute the model parameters.

Given a set of training examples that are labeled (i.e. each instance either belongs to the positive or negative class), Naïve Bayes computes multiple model parameters, such as the probability of each class label to occur. These parameters are then used to assign (or predict) a class label for any given test instance. This is done by computing the probabilities of the test instance to be assigned to each of the possible class labels. The maximum value among these probabilities decides the label of that test instance.

3.3.3.4. K-Nearest Neighbors (K-NN). K-NN is a supervised instance-based learning method. The learning process follows a lazy approach. It does not compute a model.

Given a set of training examples that are labeled (i.e. each instance either belongs to the positive or negative class), K-NN computes distances between a given test instance and all the training instances. These distances are then used to assign (or predict) a class label for the test instance. This is done by aggregating the class labels of the K closest training instances to the test instance.

Table 1
Confusion Matrix.

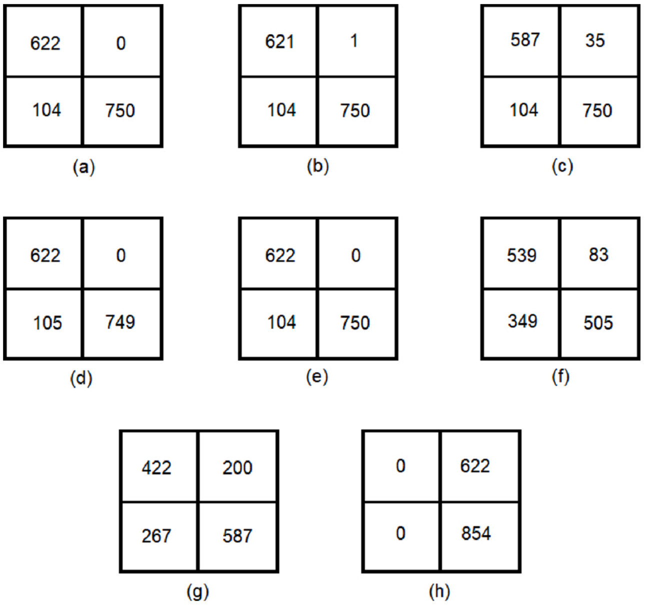
True Positive (TP)	False Negative (FN)
False Positive (FP)	True Negative (TN)

3.3.3.5. Decision Table. Decision Table is a supervised learning method. Given a set of training examples that are labeled (i.e. each instance either belongs to the positive or negative class), this method computes a model by building a decision table. That table consists of a set of conditions and corresponding actions. The table is complete if it considers every possible combination of input instances for the conditions, and prescribes the corresponding actions for each of them.

3.3.3.6. Decision Stump. Decision Stump is a supervised learning method. Given a set of training examples that are labeled (i.e. each instance either belongs to the positive or negative class), this method computes a model by building a decision tree, with only one internal node. In other words, it makes the prediction for any given test instance using only one feature of that instance. This feature is determined by computing the information gain for all features across all training instances, selecting the one with the maximum information gain value.

3.3.3.7. One Rule (OneR). OneR is a supervised learning method. Given a set of training examples that are labeled (i.e. each instance either belongs to the positive or negative class), this method computes a model by generating one rule for each feature in the data set. It then selects the one with the minimum total error.

3.3.3.8. Zero Rule (ZeroR). ZeroR is a supervised learning method. Given a set of training examples that are labeled (i.e. each instance either belongs to the positive or negative class), this method computes a model by using only the target feature (i.e. class) while ignoring all other features. It is considered the simplest classification method. It assigns



622	0
104	750

(a)

621	1
104	750

(b)

587	35
104	750

(c)

622	0
105	749

(d)

622	0
104	750

(e)

539	83
349	505

(f)

422	200
267	587

(g)

0	622
0	854

(h)

Fig. 3. Confusion matrices. (a) SVM. (b) Neural Network. (c) Naïve Bayes. (d) K-NN. (e) Decision Table. (f) Decision Stump. (g) OneR. (h) ZeroR.

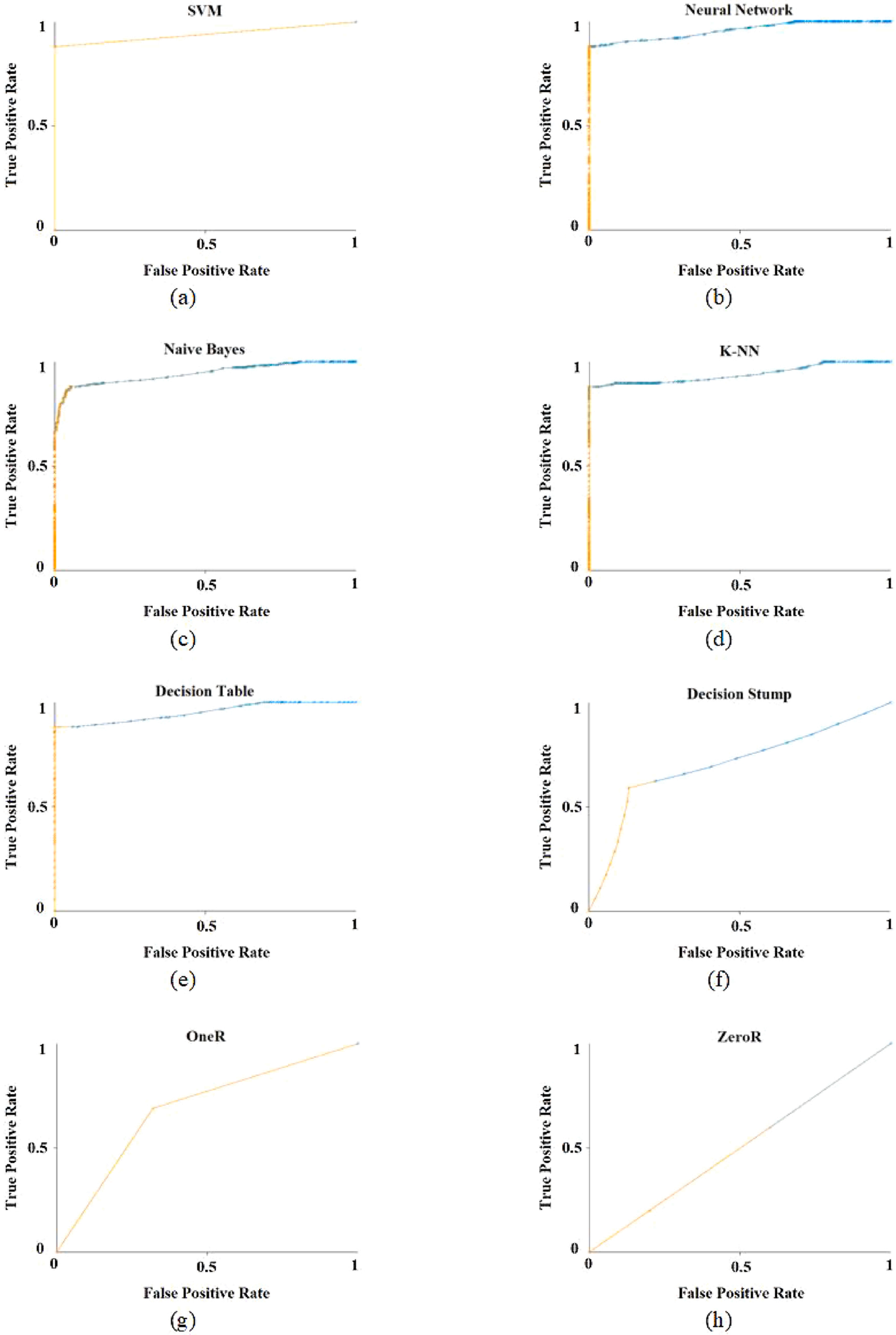


Fig. 4. ROC curves. (a) SVM. (b) Neural Network. (c) Naïve Bayes. (d) K-NN. (e) Decision Table. (f) Decision Stump. (g) OneR. (h) ZeroR.

Table 2
Summary of performance results.

	Accuracy	Root Mean Square Error	F-measure	ROC Area
Support Vector Machine (SVM)	92.95 %	26.54 %	93.0 %	93.9 %
Neural Network	92.89 %	24.23 %	92.9 %	95.5 %
Naïve Bayes	90.58 %	30.99 %	90.6 %	94.2 %
K-Nearest Neighbor (K-NN)	92.89 %	28.06 %	92.9 %	93.9 %
Decision Table	92.95 %	23.97 %	93.0 %	95.0 %
Decision Stump	70.73 %	43.86 %	70.6 %	70.1 %
OneR	68.36 %	56.25 %	68.5 %	68.3 %
ZeroR	57.86 %	49.38 %	57.9 %	49.7 %

any new test instance to the majority class. Usually, it is used as a benchmark to determine baseline performance.

3.3.4. Performance evaluation

To evaluate the performance of the eight learning algorithms, four performance measures were used: Accuracy, Root Mean Square Error, F-measure, and ROC area. These measures can be computed using a confusion matrix and cross validation methods.

3.3.4.1. Confusion matrix. The confusion matrix is used to visualize the performance of a binary (2-class) supervised learning problem by creating a 2-by-2 matrix. Each row in the matrix shows the instances in the predicted (or computed) class, while each column shows the instances in the actual class. The resulting matrix consists of four values (see Table 1).

- o True Positive (TP): are the number of instances that were classified (using the predictive model) as positive, and are actually positive.
- o False Positive (FP): are the number of instances that were classified (using the predictive model) as positive, but they are actually negative.
- o False Negative (FN): are the number of instances that were classified (using the predictive model) as negative, but they are actually positive.
- o True Negative (TN): are the number of instances that were classified (using the predictive model) as negative, and they are actually negative.

3.3.4.2. Cross validation. Cross Validation is a statistical method used to measure the performance of learning and classification methods. This is done by splitting the available labeled data instances into k folds. One of these folds is used for testing, and the rest are used for training. This work used 10-fold cross validation. The data instances are divided into 10 folds. For 10 iterations, one-fold was used for testing and 9 folds for training, such that in every iteration a different fold is used for testing.

3.3.4.3. Accuracy. The accuracy of a classifier is computed as the number of correctly classified instances to the total number of instances. It is given by:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

3.3.4.4. Root mean square error. The Root Mean Square Error (RMSE) is computed as the square root of the average of squared differences between the predicted classes (or labels) and the actual ones. It is given by:

$$RMSE = \sqrt{\frac{FP + FN}{TP + TN + FP + FN}}$$

3.3.4.5. F-measure. The F-measure is computed by combining the two measures of precision and recall. It is given by:

$$Fmeasure = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

where,

$$Precision = \frac{TP}{TP + FP}$$

and

$$Recall = \frac{TP}{TP + FN}$$

3.3.4.6. ROC area. The Receiver Operating Characteristic (ROC) is another way to measure the performance of a classifier. This is done by plotting the True Positive Rate against the False Positive Rate. The area under the resulting ROC curve is then used to measure the performance of the classifier. The closer the area to 1 is, the better the classifier is. The true/false positive rates are given by:

$$True\ Positive\ Rate = \frac{TP}{TP + FN}$$

and

$$False\ Positive\ Rate = \frac{FP}{FP + TN}$$

4. Results and discussion

4.1. Confusion matrices

Fig. 3 shows the confusion matrices that resulted from applying 10-fold cross validation to the eight selected classifiers. (Large numbers in the upper-left and lower right boxes of these matrices represent good scores. Large numbers in the lower-left and upper right boxes of these matrices represent bad scores.)

4.2. ROC curves

Fig. 4 shows the ROC curves that resulted from applying 10-fold cross validation to the eight selected classifiers.

4.3. Performance measures

Fig. 4 shows the ROC curves that resulted from applying 10-fold cross validation to the eight selected classifiers.

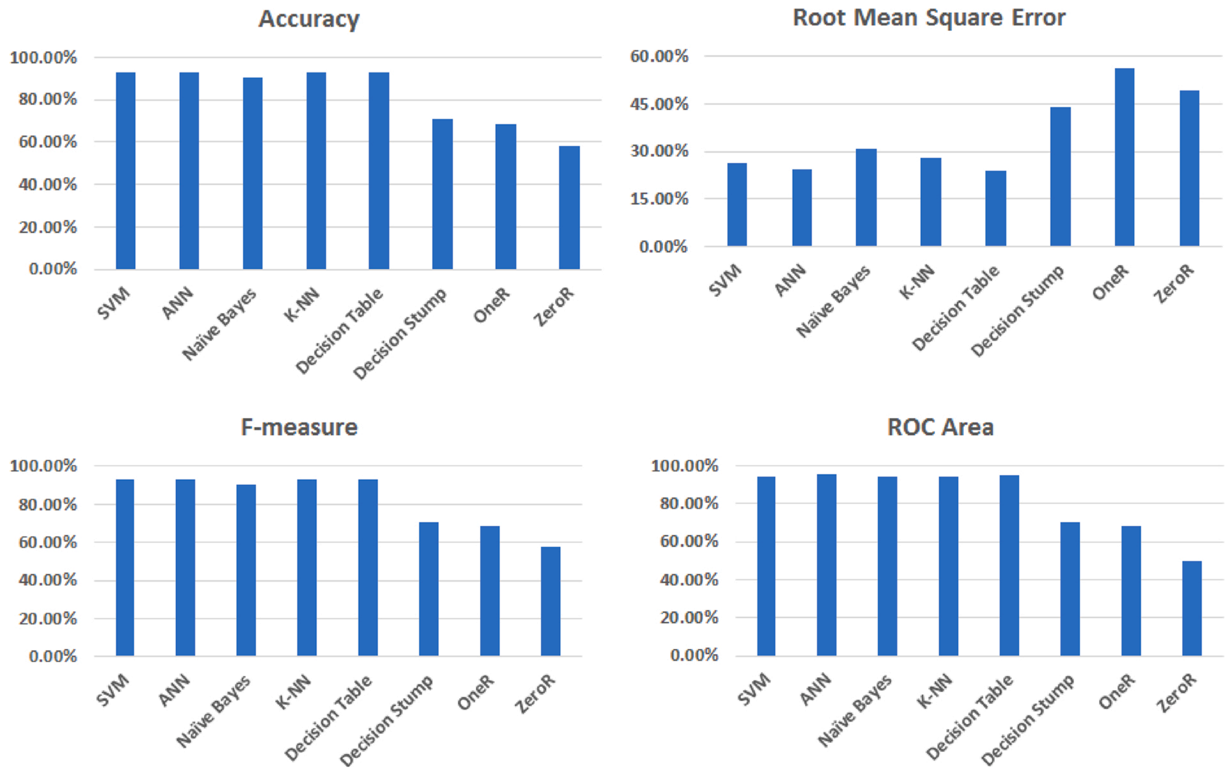
Table 2 and Fig. 5 compare the performance of the eight algorithms. It shows the Accuracy, Root Mean Square Error, F-measure and ROC Area of each algorithm, which were calculated using the well-known 10-fold cross validation method [29].

The results presented in Table 2 and Fig. 5 suggest that the models built using SVM, Neural Network, Naïve Bayes, K-NN and Decision Table algorithms are effective in predicting confirmed and potential cases of COVID-19.

Taken together, this suggests that our proposed IoT-based framework could use a combination of these five effective models. This could be done by aggregating the results of these five learnt models, based on majority votes.

5. Conclusions

This paper has proposed an IoT-based framework to reduce the impact of communicable diseases. The proposed framework was used to employ potential COVID-19 case information and health records of confirmed COVID-19 cases to develop a machine-learning-based predictive model for disease, as well as for analyzing the treatment response. The framework also communicates these results to healthcare physicians, who can then respond swiftly to suspected cases identified


Accuracy

Root Mean Square Error

F-measure

ROC Area

Fig. 5. Performance measures of the eight algorithms. (a) Accuracy. (b) Root Mean Square Error, (c) F-measure. (d) ROC Area.

by the predictive model by following up with any further clinical investigation needed to confirm the case. This allows the confirmed cases to be isolated and given appropriate health care.

An experiment was conducted to test eight machine learning algorithms on a real COVID-19 dataset. They are: (1) Support Vector Machine, (2) Neural Network, (3) Naïve Bayes, (4) K-Nearest Neighbor (K-NN), (5) Decision Table, (6) Decision Stump, (7) OneR, and (8) ZeroR. The results showed that all these algorithms, except the Decision Stump, OneR, and ZeroR achieved accuracies of more than 90 %. Using the five best algorithms would provide effective and accurate identification of potential cases of COVID-19.

Employing the proposed real-time framework could potentially reduce the impact of communicable diseases, as well as mortality rates through early detection of cases. This framework would also provide the ability to follow up on recovered cases, and a better understanding the disease.

CRediT authorship contribution statement

Mwaffaq Otoom: Conceptualization, Data curation, Formal analysis, Methodology, Software, Validation, Writing - original draft, Writing - review & editing. **Nesreen Otoum:** Conceptualization, Data curation, Formal analysis, Methodology, Software, Validation, Writing - original draft, Writing - review & editing. **Mohammad A. Alzubaidi:** Conceptualization, Data curation, Formal analysis, Methodology, Software, Validation, Writing - original draft, Writing - review & editing. **Yousef Etoom:** Conceptualization, Data curation, Formal analysis, Methodology, Software, Validation, Writing - original draft, Writing - review & editing. **Rudaina Banihani:** Conceptualization, Data curation, Formal analysis, Methodology, Software, Validation, Writing - original draft, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Coronavirus COVID-19 Global Cases by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University (JHU), World Health Organization, 2020. Accessed 21 July, https://coronavirus.jhu.edu/map.html.

[2] WHO Director-General's Opening Remarks at the media Briefing on COVID-19 -. 11 March, 2020. Accessed 11 March 2020, https://www.who.int/dg/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19-.11-march-2020.

[3] New York Post, The Most Promising Coronavirus Breakthroughs so Far, from Vaccines to Treatments, 2020. April 8, https://nypost.com/2020/04/08/coronavirus-breakthroughs-how-close-are-we-to-a-vaccine/.

[4] M.P. Kelly, Digital technologies and disease prevention, Am. J. Prev. Med. 51 (5) (2016) 861–866, https://doi.org/10.1016/j.amepre.2016.06.012.

[5] P.M. Hlaing, T.R. Nopparatjamjomras, S. Nopparatjamjomras, Digital technology for preventative health care in Myanmar, Digital Medicine 4 (3) (2018) 117–121, https://doi.org/10.4103/digm.digm_25_18.

[6] L. Cerina, S. Notargiacomo, M.G. Paccanit, M.D. Santambrogio, A fog-computing architecture for preventive healthcare and assisted living in smart ambients. IEEE 3rd International Forum on Research and Technologies for Society and Industry (RTSI), 2017, pp. 1–6, https://doi.org/10.1109/RTSI.2017.8065939. Modena.

[7] B. Dinesen, B. Nonnecke, D. Lindeman, et al., Personalized telehealth in the future: a global research agenda, J. Med. Internet Res. 18 (3) (2016) e53, https://doi.org/10.2196/jmir.5257.

[8] M. Usak, M. Kubiatko, M.S. Shabbir, O.V. Dudnik, K. Jermsittiparsert, L. Rajabion, Health care service delivery based on the Internet of things: a systematic and comprehensive study, Int. J. Commun. Syst. 33 (2) (2020), e4179.

[9] F. Wu, T. Wu, M.R. Yuce, An internet-of-things (IoT) network system for connected safety and health monitoring applications, Sensors 19 (1) (2019) 21.

[10] H. Hamidi, An approach to develop the smart health using Internet of Things and authentication based on biometric technology, Future Gen. Comput. Syst. 91 (2019) 434–449.

[11] M. Rath, B. Pattanayak, Technological improvement in modern health care applications using Internet of Things (IoT) and proposal of novel health care approach, Int. J. Hum. Rights Healthcare 12 (2) (2019) 148–162, https://doi.org/10.1108/IJHRH-01-2018-0007.

[12] A. Darwish, A.E. Hassanien, M. Elhoseny, A.K. Sangaiah, K. Muhammad, The impact of the hybrid platform of internet of things and cloud computing on healthcare systems: opportunities, challenges, and open problems, J. Ambient Intell. Hum. Comput. 10 (10) (2019) 4151–4166.

[13] C.-L. Zhong, Y.-L. Li, Internet of things sensors assisted physical activity recognition and health monitoring of college students, Measurement 159 (2020), 107774.
[14] S. Din, A. Paul, Erratum to "Smart health monitoring and management system: toward autonomous wearable sensing for Internet of Things using big data analytics, Future Gener. Comput. Syst 91 (2019) 611–619.
[15] T.T. Nguyen, Artificial Intelligence in the Battle against Coronavirus (COVID-19): A Survey and Future Research Directions, 2020, https://doi.org/10.13140/RG.2.2.36491.23846. Preprint.
[16] H.S. Maghdid, K.Z. Ghafoor, A.S. Sadiq, K. Curran, K. Rabie, A Novel AI-Enabled Framework to Diagnose Coronavirus COVID-19 Using Smartphone Embedded Sensors: Design Study, arXiv preprint arXiv:2003.07434, 2020.
[17] A.S.S. Rao, J.A. Vazquez, Identification of COVID-19 can be quicker through artificial intelligence framework using a mobile phone-based survey in the populations when cities/towns are under quarantine, Infect. Control Hosp. Epidemiol. (2020) 1–18, https://doi.org/10.1017/ice.2020.61.
[18] Z. Allam, D.S. Jones, On the coronavirus (COVID-19) outbreak and the smart city network: universal data sharing standards coupled with artificial intelligence (AI) to benefit urban health monitoring and management, Healthcare 8 (1) (2020) 46.
[19] S.A. Fatima, N. Hussain, A. Balouch, I. Rustam, M. Saleem, M. Asif, IoT enabled smart monitoring of coronavirus empowered with fuzzy inference system, Int. J. Adv. Res. Ideas Innov. Technol. 6 (1) (2020).
[20] N.C. Peeri, N. Shrestha, M.S. Rahman, R. Zaki, Z. Tan, S. Bibi, M. Baghbanzadeh, N. Aghamohammadi, W. Zhang, U. Haque, The SARS, MERS and novel coronavirus (COVID-19) epidemics, the newest and biggest global health threats: what lessons have we learned, Int. J. Epidemiol. 49 (3) (2020) 717–726, https://doi.org/10.1093/ije/dyaa033.
[21] M. Otoom, H. Alshraideh, H.A. Almasaeid, D. López-de-Ipiña, J. Bravo, Real-time statistical modeling of blood sugar, J. Med. Syst. 39 (10) (2015) 123.
[22] H. Alshraideh, M. Otoom, A. Al-Araida, H. Bawaneh, J. Bravo, A web based cardiovascular disease detection system, J. Med. Syst. 39 (10) (2015) 122.
[23] J. Medina, M. Espinilla, Á.L. García-Fernández, L. Martínez, Intelligent multi-dose medication controller for fever: from wearable devices to remote dispensers, Comput. Electrical Eng. 65 (2018) 400–412.
[24] Y. Umayahara, Z. Soh, K. Sekikawa, T. Kawae, A. Otsuka, T. Tsuji, A mobile cough strength evaluation device using cough sounds, Sensors 18 (2018) 3810.
[25] D. Ichwana, R.Z. Ikhlas, S. Ekariani, Heart rate monitoring system during physical exercise for fatigue warning using non-invasive wearable sensor, in: 2018 International Conference on Information Technology Systems and Innovation (ICITSI), Bandung - Padang, Indonesia, 2018, pp. 497–502.
[26] B. Askarian, S.-H. Yoo, J.W. Chong, Novel image processing method for detecting strep throat (streptococcal pharyngitis) using smartphone, Sensors 19 (15) (2019) 3307.
[27] A. Gaidhani, K.S. Moon, Y. Ozturk, S.Q. Lee, W. Youm, Extraction and analysis of respiratory motion using wearable inertial sensor system during trunk motion, Sensors (Basel) 17 (12) (2017) 2932, https://doi.org/10.3390/s17122932.
[28] COVID-19 Open Research Dataset (CORD-19), 2020, https://doi.org/10.5281/zenodo.3715506. Version 2020-03-13. Retrieved from: https://pages.semanticscholar.org/coronavirus-research. Accessed 2020-03-22.
[29] P.N. Tan, Introduction to Data Mining, Pearson Education, India, 2018.
[30] F. Eibe, M.A. Hall, I.H. Witten, The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", fourth edition, Morgan Kaufmann, 2016.
[31] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of things: a survey on enabling technologies, protocols, and applications, IEEE Commun. Surveys Tutorials 17 (4) (2015) 2347–2376.
[32] A.V. Dastjerdi, R. Buyya, Fog computing: helping the Internet of Things realize its potential, Computer 49 (8) (2016) 112–116.
[33] K. Zhao, L. Ge, A survey on the internet of things security, in: 2013 9th International Conference on Computational Intelligence and Security (CIS), IEEE, 2013, pp. 663–667.
[34] C. Perera, A. Zaslavsky, P. Christen, D. Georgakopoulos, Context aware computing for the internet of things: a survey, IEEE Commun. Surveys Tutorials 16 (1) (2013) 414–454.
[35] P. Sethi, S.R. Sarangi, Internet of things: architectures, protocols, and applications, J. Electr. Comput. Eng. 2017 (2017) 1–25, https://doi.org/10.1155/2017/9324035.
[36] E. Ahmed, I. Yaqoob, I.A.T. Hashem, I. Khan, A.I.A. Ahmed, M. Imran, A. V. Vasilakos, The role of big data analytics in Internet of Things, Comput. Netw. 129 (2017) 459–471.
[37] M.A. Al-Garadi, A. Mohamed, A. Al-Ali, X. Du, I. Ali, M. Guizani, A survey of machine and deep learning methods for Internet of Things (IoT) security, IEEE Commun. Surveys Tutorials (2018), https://doi.org/10.1109/COMST.2020.2988293.
[38] Internet of Medical Things, Forecast to 2021 [Online]: https://store.frost.com/internet-of-medical-things-forecastto-2021.html, Accessed on 6 June 2020.
[39] C. Camara, P. Peris-Lopez, J.E. Tapiador, Security and privacy issues in implantable medical devices: a comprehensive survey, J. Biomed. Inform. 55 (2015) 272–289.
[40] R. RlTawy, A.M. Youssef, Security tradeoffs in cyber physical systems: a case study survey on implantable medical devices, IEEE Access 4 (2016) 959–979.
[41] M.A. Alzubaidi, M. Otoom, N. Otoum, Y. Etoom, R. Banihani, A Novel Computational Method for Assigning Weights of Importance to Symptoms of COVID-19 Patients, 2020. Under Review.
[42] L.H. Lorena, A.C. Carvalho, A.C. Lorena, Filter feature selection for one-class classification, J. Intell. Rob. Syst. 80 (1) (2015) 227–243.