Computers in Biology and Medicine 147 (2022) 105682

Contents lists available at ScienceDirect

Computers in Biology and Medicine

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

PCovNet: A presymptomatic COVID-19 detection framework using deep learning model using wearables data

Farhan Fuad Abir [a], Khalid Alyafei [b], Muhammad E.H. Chowdhury [c,*], Amith Khandakar [c], Rashid Ahmed [b,d], Muhammad Maqsud Hossain [e], Sakib Mahmud [c], Ashiqur Rahman [f], Tareq O. Abbas [g], Susu M. Zughaier [h], Khalid Kamal Naji [i,**]

[a] Department of Electrical and Electronic Engineering, University of Dhaka, Dhaka, 1000, Bangladesh
[b] Department of Mechanical and Industrial Engineering, College of Engineering, Qatar University, Doha, 2713, Qatar
[c] Department of Electrical Engineering, Qatar University, Doha, 2713, Qatar
[d] Biomedical Research Centre, Qatar University, Doha, 2713, Qatar
[e] NSU Genome Research Institute (NGRI), North South University, Dhaka, 1229, Bangladesh
[f] Institute of Multidisciplinary Research for Advanced Materials, Tohoku University, Japan
[g] Urology Division, Surgery Department, Sidra Medicine, Doha, Qatar, 26999
[h] Department of Basic Medical Sciences, College of Medicine, QU Health, Qatar University, Doha, 2713, Qatar
[i] College of Engineering, Qatar University, Doha, 2713, Qatar

ARTICLE INFO

Keywords:
Presymptomatic
COVID-19
Smartwatch
Resting heart rate
Anomaly detection
Long short-term memory
Variational autoencoder

ABSTRACT

While the advanced diagnostic tools and healthcare management protocols have been struggling to contain the COVID-19 pandemic, the spread of the contagious viral pathogen before the symptom onset acted as the Achilles' heel. Although reverse transcription-polymerase chain reaction (RT-PCR) has been widely used for COVID-19 diagnosis, they are hardly administered before any visible symptom, which provokes rapid transmission. This study proposes PCovNet, a Long Short-term Memory Variational Autoencoder (LSTM-VAE)-based anomaly detection framework, to detect COVID-19 infection in the presymptomatic stage from the Resting Heart Rate (RHR) derived from the wearable devices, i.e., smartwatch or fitness tracker. The framework was trained and evaluated in two configurations on a publicly available wearable device dataset consisting of 25 COVID-positive individuals in the span of four months including their COVID-19 infection phase. The first configuration of the framework detected RHR abnormality with average Precision, Recall, and F-beta scores of 0.946, 0.234, and 0.918, respectively. However, the second configuration detected aberrant RHR in 100% of the subjects (25 out of 25) during the infectious period. Moreover, 80% of the subjects (20 out of 25) were detected during the pre-symptomatic stage. These findings prove the feasibility of using wearable devices with such a deep learning framework as a secondary diagnosis tool to circumvent the presymptomatic COVID-19 detection problem.

1. Introduction

The COVID-19 pandemic has been one of the most significant global events in this decade that have affected the whole world at the same time and marked the most crucial struggle of humanity against a highly contagious viral pathogen in modern times. The pathogen, namely Severe Acute Respiratory Syndrome Coronavirus-2 (SARS-CoV-2), commonly known as the coronavirus, has caused approximately 491 million infection cases with about 6.1 million death tolls worldwide to date [1]. Despite the advancement of technology and healthcare protocols, the lack of preparedness for utilizing these technologies was a major lesson in this pandemic [2–4]. However, researchers from diverse sectors have come forward with their innovations and findings to fight back this pandemic by making effective use of our existing technologies, including the discovery of effective vaccines within the shortest time-span to date, improvement in the pandemic management employing digital technologies, widespread contact-tracing, fast and effective diagnosis methods, and new methods for detecting asymptomatic carriers.

SARS-CoV-2 has some similar characteristics (e.g., transmission

* Corresponding author.
** Corresponding author. College of Engineering, Qatar University, Doha, 2713, Qatar.
E-mail addresses: mchowdhury@qu.edu.qa (M.E.H. Chowdhury), knaji@qu.edu.qa (K.K. Naji).

https://doi.org/10.1016/j.compbiomed.2022.105682
Received 19 February 2022; Received in revised form 19 May 2022; Accepted 30 May 2022
Available online 7 June 2022

routes, reproduction number (R0), and incubation period) compared to other respiratory viral pathogens, namely SARS-CoV, Middle East Respiratory Syndrome Coronavirus (MERS-CoV), and 2009 H1N1 Influenza (H1N1-pdm09) [5,6]. However, its presymptomatic infection rate, viral shedding duration, and variability of characteristics across different countries have made the COVID-19 pandemic more unpredictable and harder to contain [6,7]. Moreover, several studies found evidence behind its transmission by asymptomatic individuals, which has been the most crucial variable in the success of COVID-19 containment [8–10]. Contact tracing [11] and frequent testing [12] have been proposed to detect and quarantine the asymptomatic COVID-19 carriers. Although reverse transcription-polymerase chain reaction (RT-PCR) has a margin of error in the result [13,14], it is still considered the standard for testing COVID-19 infection on a mass scale. These measures could succeed in a controlled environment with sufficient resources; however, their feasibility comes into question for areas with limited resources and a higher population density during a surging pandemic. Hence, several promising avenues have been explored on the alternate COVID-19 detection systems for such resource constraint environments. An early detection system [15] and a mortality prediction system [16] for COVID-19 infection were proposed using blood biomarkers.

However, continuous monitoring is widely considered a viable alternative to direct COVID-19 testing in an attempt to resolve these constraints. Ponomarev et al. [17] observed a significant correlation between Heart Rate Variability (HRV) and underlying symptoms of COVID-19 infection. Heart Rate is the variation of time intervals between two successive heartbeats. Buchhorn et al. [18] tracked the Electrocardiogram (ECG) of a 58-year-old male patient during the span of COVID-19 infection with a Holter Monitor and found an abrupt decline in HRV and Heart Rate (HR). Nowadays, smartwatches and fitness trackers are capable of monitoring essential vitals, e.g., HR, steps, burned calories, sleep duration, and sleep stages. Despite having accuracy issues among different models and brands of wearable devices, such devices are gradually integrated into the healthcare monitoring systems [19–21]. Moreover, these wearable devices are non-invasive, well--connected with smartphones, and used worldwide, which signify the potential of such devices in solving the asymptomatic COVID-19 detection problem.

Compared to the volume of COVID-19 research, asymptomatic COVID-19 detection based on wearable devices has an inadequate number of works. Mishra et al. [22] presented an extensive statistical study of physiological (Resting Heart Rate) and activity (steps and sleep duration) data from 105 subjects, including 32 COVID-positive individuals, and found alterations in their vitals even before the symptom onset. They also claimed that their two-tiered warning system alerts the user in the event of an abnormal HR with a 63% detection rate for known COVID -infected individuals. Although they explored the correlations of these three parameters with the infection, their presymptomatic detection system employed the difference in Resting Heart Rate (RHR-diff) and the ratio of HR and steps. Additionally, Radin et al. [23] also found a higher correlation between elevated RHR and increased sleep duration with the COVID-19 infection among Fitbit users in 5 states of the United States. On the other hand, Quer et al. [24] proposed a complementary method to the traditional COVID-19 testing by statistically evaluating the self-reported symptom survey along with sensor data from the fitness tracker. Moreover, Smarr et al. [25] explored the possibility of using wearable devices to monitor elevated body temperature due to the COVID-19 infection.

In addition to the statistical approaches, deep learning methods have also been used in COVID-19 infection detection, contact tracing, and drug development [26]. Deep learning-based COVID-19 detection systems using Computed Tomography (CT) scans [27,28] and chest X-ray images [29–32] have shown promising performance. Moreover, Bogu et al. [33] took their previous statistical work [22] further and proposed a Long Short-Term Memory (LSTM)-based autoencoder network to detect abnormal RHR individually among the COVID-19-infected subjects. However, in contrast to their previous work, they only focused on the RHR of 25 COVID-19-infected individuals, and the model could predict the aberrant RHR among 23 of them (14 during the presymptomatic stage and 9 after the symptom onset). Although in preprint, this work marks the effectiveness of deep learning methods with wearable sensor data for detecting COVID-19 infection. More recently, Liu et al. [34] proposed a combination of Convolutional autoencoder with the contrastive loss on 19 multiple sclerosis patients' RHR recorded by Fitbit smartwatches. With the addition of contrastive loss, their model is claimed to perform better than traditional CNN (Convolutional Neural Network) or LSTM autoencoders. On the other hand, Chharia et al. proposed a deep learning-based Computer-Aided Diagnosis (CAD) system to predict whether an immerging pathogen might create a pandemic [35]. Furthermore, owing to the advancements in deep learning research, several advanced architectures have shown improved performances compared to the traditional DenseNet, CNN, and LSTM in a variety of domains [36–38].

This study presents PCovNet, a deep learning-based anomaly detection framework to detect the aberrant RHR before the symptom onset of COVID-19 infection. The underlying model is a Long Short-Term Memory (LSTM) Variational Autoencoder (VAE) leveraging the benefit of VAE, which is capable of learning smooth latent state representations of the input data and producing a reconstructed signal. Based on the reconstruction loss, PCovNet can detect anomalies and thus, identifies probable COVID infection even before the symptom onset. We evaluated the PCovNet framework on the RHR of 25 COVID-19-infected subjects recorded via Fitbit smartwatches. The dataset is a part of the phase-1 study by a group from the Genetics Department of Stanford University [22]. Their second study [33] also used the same dataset and used an LSTM autoencoder model, but this framework improves the overall result by using the Variational Autoencoder (VAE) architecture which often learns better latent state representations compared to the vanilla autoencoders [39]. Overall, the contributions of this work are as follows.

Firstly, we introduced PCovNet, an LSTM Variational Autoencoder-based framework, for detecting anomalous RHR with two separate configurations with different application scenarios.

Secondly, we validated the use of smartwatch-based RHR monitoring systems as a secondary diagnostic tool for continuous health monitoring and anomalous RHR detection even before the onset of COVID-19 symptoms.

2. Material and methods

2.1. Dataset

The research group from Stanford University Genetics Department led a study among a cohort of 5262 subjects who finished surveys of respiratory disease symptoms, diagnosis, and severity with the explicit date via the REDCap (Research Electronic Data Capture) survey web platform. Among these subjects, the study also included the smartwatch data from 4642 subjects where Fitbit users constitute the majority of 3325 individuals. However, among the COVID-19 infected Fitbit users, only 32 individuals' data were available during their symptoms and diagnosis period [22,40].

This study is based on the publicly available dataset which comprised these 32 subjects with COVID-19 infection, 15 with other respiratory diseases, and 73 healthy individuals, with additional demographic and symptom information [40]. Additionally, each subject is identified with a particular ID assigned during the data collection. As this study focused on the presymptomatic detection of COVID-19 infection, it targeted only the COVID-positive subject data from the dataset. For these subjects, sleep duration and stage information are also provided with RHR and step counts.

In addition to the sensor data channels and symptom onset date, other crucial factors for such a study are the baseline RHR calculation and the infectious period determination. The incubation period is the

number of days between the day of infection and the symptom onset. Several researchers calculated the median incubation period of SARS-CoV-2 to vary in different sample groups from 3 days (ranging from 0 to 24 days) [41] to 5 days (ranging from 1 to 19 days) [42]. In a separate study, Li et al. [43] also reported a median incubation period of 5 days with a 95% Confidence Interval (CI) of 4.1–7.0 days. On the other hand, viral shedding denotes the release of viral particles by an asymptomatic or symptomatic host. Similar to the incubation period, the duration of viral shedding also varies in different studies and can be as long as 83 days after the symptom onset [44]. However, the viral load (numerical expression of the amount of viral pathogen in a host fluid) gradually decreased to the detection limit at about 21 days of symptom onset [45].

As PCovNet is built for analyzing the RHR abnormality individually, we established two criteria for this study.

A. Each subject must have HR and steps data.
B. The data collection must span from at least 20 days before the symptom onset to 21 days afterward.

Here, criterion A is essential for RHR calculation, criterion B ensures the evaluation of baseline RHR, which is the reference for the model. Albeit all data satisfied criterion A, 7 COVID-19, 5 Other illness, and 6 Healthy subjects did not fulfill criterion B. As a result, the working dataset for PCovNet shrunk to 25 COVID-19, 10 Other illness, and 67 Healthy subjects. The dataset overview is shown in Fig. 1.

2.2. Data preprocessing

The raw data provided in the dataset is passed through five separate steps and prepared for the anomaly detection model. The raw HR and steps are merged to derive the RHR. Afterward, the derived RHR is passed through the filtering and resampling stage which filters out the daily activity-induced RHR fluctuations. The RHR sequence is split into train and test sets in the third stage. The next two stages further process the train and test sets to prepare for the model. After training the anomaly detector in the sixth step, the losses are used to find out the anomalous portions of the signal. The methodology overview is shown given in Fig. 2.

2.2.1. Resting Heart Rate calculation

As mentioned before, our channel of interest for this study was the Resting Heart Rate (RHR), which is not provided in the dataset. Therefore, in the very first preprocessing step, our goal was to merge the separated raw HR and steps data based on the provided timestamps. However, albeit processed within the Fitbit devices, the provided wearable data do not have the same timestamps and equal frequency. In order to fix the intermittent nature of the signal, we used a 1 min rolling average filter on the HR and steps channels and then merged HR and step channels [22,33].

Resting heart rate denotes the heart beats per minute during a resting state. In this case, we identified a resting state if a person does not have any step count for 12 consecutive minutes, which denotes inactivity [33]. Hence, we filtered out the HR unless the step count is zero for 12 consecutive minutes. Although this step filters out a good portion of the data points, it also filters the activity-induced HR elevation, which is beneficial for sensing any underlying health condition.

2.2.2. Filtering and resampling

Looking closely at 1 min resampled RHR, it was observed that it contained numerous short-term RHR fluctuations, which did not persist for hours. As our measure of the inactive state was based on the null step count for 12 consecutive minutes, the RHR calculated in such a way might contain an abrupt elevation due to some other factors, e.g., watching a horror movie, playing video games, or having a stressful conversation. Moreover, having a festive mood during the holiday might cause a temporary RHR elevation [22,33]. Therefore, further filtering was necessary to determine the effect of an underlying condition causing abnormal RHR. In this step, we filtered the RHR further with a rolling average of 400 samples (6 h 40 min) and resampled the filtered data into a 1-h period.

2.2.3. Dataset split

Mishra et al. divided the RHR of the COVID-19 infection timeline into four distinct regions [22]. We identified the spans of these regions considering the domain knowledge of symptom onset, incubation period, viral load, and viral shedding from previous literature [41–45]. These periods are shown in Fig. 3.

1. *Baseline Period:* It is the region before 20 days of the symptom onset. RHR during this period is considered normal.
2. *Non-infectious Period:* This region spans from 20 days to 10 days before the symptom onset. Despite having similar RHR to the baseline period for the majority of the symptomatic individuals, a handful might show viral shedding during this period [45].
3. *Infectious Period:* This region is most likely to contain anomalous RHR, which spans from 7 days before the symptom onset to 21 days after that.

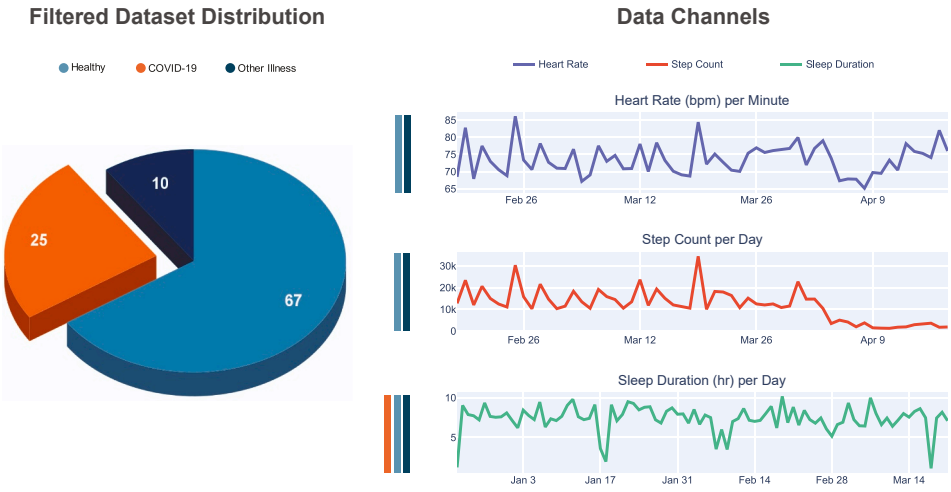


Fig. 1. The dataset distribution for the study is shown in the left pie chart. Three available data channels are shown in the right plot. There are colorful strips in between the two charts denoting the availability of the channel immediately to the right in the certain subject class indicated by the color.

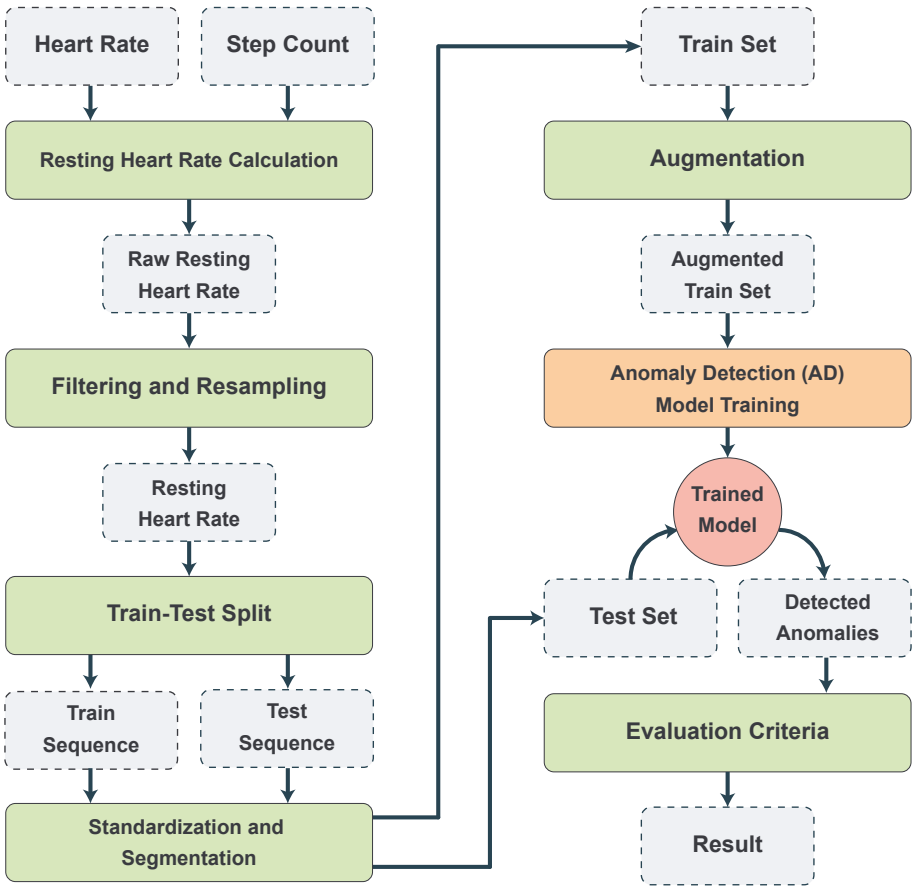


Fig. 2. Methodology overview of the anomaly detection framework using PCovNet.

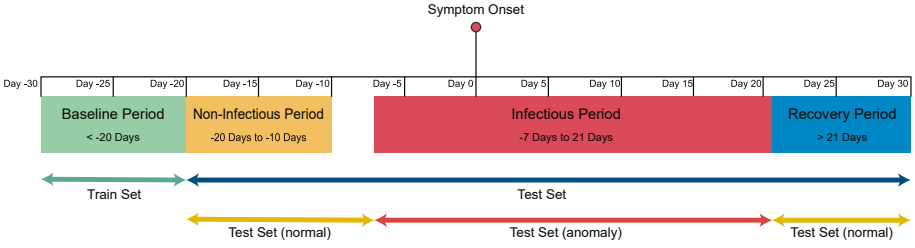


Fig. 3. Four stages of COVID-19 infections are shown here - baseline period, non-infectious period, infectious period, and recovery period. Only the data from the baseline period is kept in the training set and the rest constitutes the test set. The test set contains both anomalous and normal data. For evaluation, samples only within the infectious period are considered anomalous.

4. *Recovery Period:* In most cases, after the SARS-CoV-2 infection, the RHR comes back to the baseline after 21 days from symptom onset, which is marked by the recovery period. However, the possibility of viral shedding and having prolonged aberrant RHR is not zero; hence, this region is not labeled normal in this study.

We employed personalized training for each user where the model is trained on the normal RHR data of a user and evaluated for anomalous RHR for that user only. Based on the characteristics of the four above-mentioned regions, the whole dataset is divided into two sets - train set and test set. The train set for each subject contains only the Baseline Period data which represents the normal RHR of that user. On the other hand, the test set contains the other three periods, which may include normal RHR as well as anomalous RHR. Moreover, an additional validation set is separated from the train set just before the model training, which constitutes 5% of the train set chosen randomly.

2.2.4. Standardization and segmentation

Standardization is the process of removing the mean from a dataset and scaling it to unit variance. Before training the PCovNet model, the whole dataset is standardized based on the train set. For a train set, X_{train}, the mean (μ_{train}) and the standard deviation (σ_{train}) of the train set are calculated at first. Then the test set, X_{test}, is transformed based on the mean and standard deviation of the train set, which is shown in Equation (1).

$$X'_{test} = \left(\frac{X_{test} - \mu_{train}}{\sigma_{train}}\right) \tag{1}$$

To implement it in the PCovNet framework, we used the *StandardScaler* module from the Scikit-Learn Python package [46]. After the standardization of the datasets, the RHR sequences are divided into smaller segments by shifting a window of the same size by 1 h. This step serves two purposes - it shapes the input sequence into a fixed-length sample and increases the sample size for training the model by

making several sub-sequences from a single long sequence.

2.2.5. Augmentation

Augmentation is the process of creating new training data from the existing dataset. It is a widely used technique in the image domain for fixing the class imbalance and creating additional data for the data-hungry neural nets. In contrast to the image domain, time-series augmentation has rarely proven any significant development in a model's performance, rather causing declination. For example, rotating a cat picture still results in a cat picture but rotating an audio signal can completely change the morphology. Although time-series augmentation is more likely to cause unrepresentative data, the data processing steps in this study ensured a potential usage for the augmentation. We employed seven techniques to augment the original train dataset [47, 48].

1. *Scaling:* A value from a Gaussian distribution (mean 1, standard deviation 0.1) is chosen and multiplied with the signal to change its original amplitude.
2. *Rotation:* An arbitrary rotation is applied to the original signal which causes horizontal mirroring.
3. *Permutation:* It creates perturbation in the original signal by slicing it into 1 to 4 same length segments and randomly permuting them.
4. *Magnitude Warping:* This technique multiplies the original signal with a curve generated by a cubic spline with four knots. The knots are chosen randomly from a Gaussian distribution (mean 1, standard deviation 0.2).
5. *Time Warping:* Like Magnitude Warping, a curve is generated by a cubic spline, but in this case, the curve is used to perturb the temporal locations of the original data points of the sequence.
6. *Window Warping:* This method takes a portion of the original window and changes its frequency by a factor of 2 or 0.5.
7. *Window Slicing:* It takes 90% of the original window by slicing it randomly.

These augmentation techniques increase our number of training samples by seven times for each subject.

2.3. Anomaly detection model

This study approaches the presymptomatic COVID-19 detection task as an anomaly detection task. The use of an autoencoder is a widely adopted approach learn the representations from a feature map, where the encoder learns the latent attributes, and the decoder learns to decode the original signal from the latent attributes [49]. The low dimensionality of the latent space works as a bottleneck for the model to reconstruct the input. Whereas the conventional autoencoders return points in the latent space based on the input, the Variational Autoencoders (VAE) return a distribution. One problem of autoencoders is the discrete latent space, which sometimes occurs from overfitting. On the other hand, a regularization is introduced in the VAE due to the architecture, which forces the latent space distribution towards a normal distribution. Such regularization checks the overfitting and creates continuity in the latent space. Hence, the PCovNet model inherits Variational Autoencoder (VAE) architecture with LSTM units in contrast to the Long Short-Term Memory Networks-based autoencoder (LAAD) architecture employed in the previous study [33].

For the encoder and decoder blocks, PCovNet adopts LSTM-based architecture, which is often preferred in a wide variety of time-series classification and regression problems due to its sequence learning capabilities [50]. To facilitate the sequence input capability of LSTM units, the input data for the model is structured in a three-dimensional array, where the dimensions respectively denote the number of samples, number of time steps in a sequence (window), and number of channels (features).

2.3.1. LSTM-VAE architecture

The architecture of the PCovNet model is primarily divided into two major blocks - encoder and decoder. Moreover, we used LSTM layers in both blocks with tanh as activation and sigmoid as recurrent activation.

1. *Encoder Block:* This block contained a cascade of LSTM layers, where the unit size denoted the number of extracted features in each LSTM layer. These LSTM layers with decreasing unit numbers ensured higher-level feature encoding by the network. Each LSTM layer, except the last one, returned the same number of timesteps (sequence) as the input, and the last encoder LSTM layer returned two vectors (mean vector and variance vector) with the same dimension as the latent vector. A sampling function was used to make the final latent vector from these two. This sampling function employed a reparameterization technique, where a randomly sampled vector with the latent space dimension was taken from a unit Gaussian distribution. Afterward, this vector was used to transform the mean and variance vectors into the final latent vector. The reason for such a sampling method was to ensure the back-propagation process to learn the appropriate parameters for the mean and variance vector during the training steps which would result in a Gaussian-like latent space.
2. *Decoder Block:* Generally, a decoder block of an autoencoder is the mirror image of the encoder block. Likewise, for this VAE architecture, the LSTM layers were mirrored in shape. As opposed to the encoder block, all the LSTM layers were configured to return sequences with the same length as the input timesteps. After the last LSTM layer, a Dense layer with a Time-Distributed wrapper was used to denote the fully-connected operation on the last LSTM layer output sequence. As a result, the output shape was converted back to the input shape.

According to Fig. 4, for an input sequence x, the LSTM units extract features in successive encoder layers and output two vectors $\mu_x = g(x)$ and $\sigma_x = h(x)$ with the same dimension as the latent vector. These two vectors are sampled with the random normal vector ε to get the final latent vector z, which is shown in Equation (2).

$$z = \mu_x + \sigma_x \times \varepsilon \tag{2}$$

2.3.2. Loss calculation

In general, the anomaly detection task employs only the reconstruction loss; however, due to the VAE architecture and the Gaussian nature of the latent space, a second loss was employed in this study.

Reconstruction Loss: The most commonly used loss for any representation learning model is the reconstruction loss, where a Mean Squared Error (MSE) is calculated between the input and the reconstructed output. For the input sequence x and reconstructed sequence $\hat{x} = f(z)$, Equation (3) shows the reconstruction loss.

$$Loss_{reconstruction} = MSE[x, \hat{x}] = |x - f(z)|^2 \tag{3}$$

KL Divergence Loss: Kullback-Leibler (KL) Divergence is a statistical method of determining the difference between two distributions. A VAE architecture tends to convert the latent space distribution to a Gaussian one, so the KL Divergence method can be used to determine the distribution loss (Equation (4)). Moreover, employing the KL loss along with the reconstruction loss adds regularization to the model.

$$Loss_{KLD} = KLD[\boldsymbol{N}(\mu_x, \sigma_x) - \boldsymbol{N}(0, 1)] = KLD[\boldsymbol{N}(g(x), h(x)), \boldsymbol{N}(0, 1)] \tag{4}$$

Total Loss: The total loss of the model is the summation of these two losses shown in Equations (3) and (4).

$$\begin{aligned} Loss_{total} &= Loss_{reconstruction} + Loss_{KLD} \\ &= |x - f(z)|^2 + KLD[\boldsymbol{N}(g(x), h(x)), \boldsymbol{N}(0, 1)] \end{aligned} \tag{5}$$

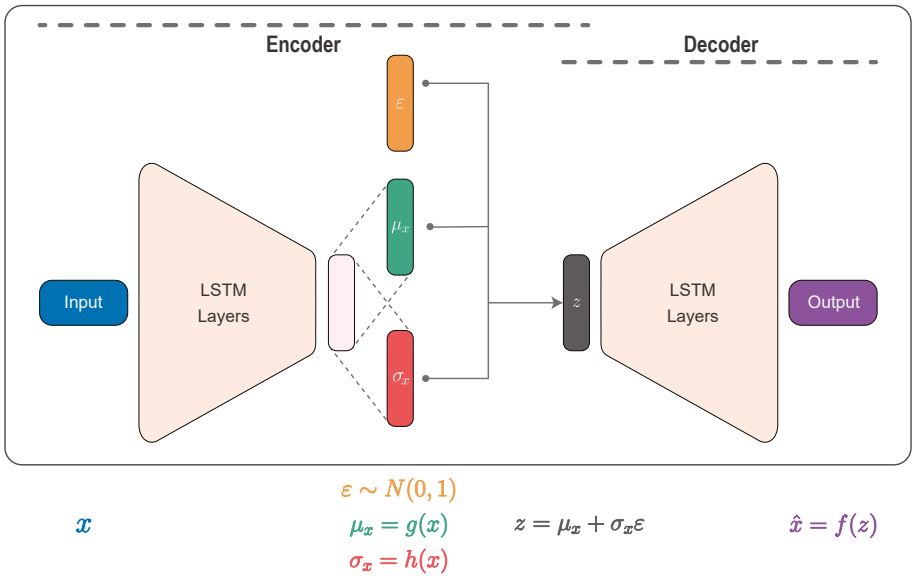


Fig. 4. The backbone of the LSTM Variational Autoencoder. The encoder block spans from the LSTM layers (which takes input sequence, x) to the latent vector, z. The decoder layers take z as input and output the reconstructed sequence, $\hat{x}$.

2.4. Threshold calculation

The anomaly detection model is trained individually and a 5% validation set is randomly chosen from the train set for model validation. After training, the whole train loss is calculated by taking the mean absolute loss value of the data points per sample. Based on the losses from each sample, the framework calculates the threshold in two ways.

1. *Statistical Threshold Estimation (STE):* In this method, the threshold was determined to be the three standard deviations above the mean of the training loss.
2. *MinMax Threshold Estimation (MTE):* The maximum value in the training loss was set to be the threshold in this method. As the absolute values of individual datapoint losses were taken for training loss determination, all the losses are positive. Hence, the minimum threshold is redundant in our case.

In both threshold methods, if the test loss (calculated in the same way as the training loss) was greater than the threshold, the sample was marked as an anomalous one.

3. Experimental results

Despite choosing LSTM-VAE as the anomaly detector, we conducted several experiments to achieve the best performance from the PCovNet framework. As described in Subsection 2.3, we evaluated the PCovNet framework with two threshold methods and each showed a different performance than the other one in terms of the evaluation metrics. However, apart from the threshold calculation techniques, the anomaly detection model has been the same for each user. Moreover, a separate anomaly detection model was trained and evaluated for each user. The overall number of samples in the train and test dataset per subject are shown in Supplementary Appendix 1.

3.1. Experimental setup

During the training process, *Adam* optimizer [51] was used with learning rate, $\alpha = 10^{-4}$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. We used a batch size of 64 and the maximum epoch was set to 1000 along with an early stopping callback criterion, which was defined with the patience of 50 epochs on validation loss to avoid overfitting. Besides, another callback criterion was used to get back to the last checkpoint with minimum validation loss.

The deep learning part of the experiment was conducted in Python programming language with *Keras* [52] and *TensorFlow* [53] packages. Moreover, for data analysis and processing, we used *Numpy* [54] and *Pandas* [55,56] packages of Python. For training purposes, we used *Google Colab* with GPU support. The overall hardware specification was - Processor: Intel(R) Xeon(R) CPU @ 2.30 GHz, RAM: 13 GB, GPU: *NVIDIA Tesla K80*. Moreover, we generated all the figures in this work using *Matplotlib* [57] and *Seaborn* [58] libraries.

3.2. Evaluation metrics

This study had a well-defined train set, which constitutes the baseline RHR determined by the literature and the subject-reported symptom onset date. However, due to the lack of ground truth, we had to rely on the literature to determine the infectious period of each subject. Moreover, the goal of this study compelled us to consider the rate of pre-symptomatic detection along with the model performance for anomaly detection. Hence, we relied on two sets of evaluation metrics:

1. *Statistical Metrics:* Based on the domain knowledge, we have assumed that all samples within the infectious period are anomalous ones. Therefore, the number of detected anomalies within the infectious period is True Positive (TP) and outside the infectious period is False Positive (FP). On the other hand, the number of timestamps labeled as normal by the model, if within the infectious period, is denoted as the False Negative (FN). The remaining predicted samples are marked as True Negative (TN). Based on these four parameters, Precision and Recall are calculated.

$$Precision = \frac{\sum TP}{\sum (TP + FP)} = \frac{\#Detected\ anomalous\ samples\ within\ the\ infectious\ period}{\#Total\ detected\ anomalous\ samples} \tag{6}$$

$$Recall = \frac{\sum TP}{\sum (TP + FN)} = \frac{\#Detected\ anomalous\ samples\ within\ the\ infectious\ period}{\#Total\ samples\ within\ infectious\ period} \tag{7}$$

However, this assumption to mark all samples within the average infectious period is not completely accurate, since the incubation period might vary for each individual. Nevertheless, such an assumption gives us a good insight into the usability of this framework in our purpose. As there is no exact ground truth provided, we considered Precision in this regard, which quantifies the anomaly detection capability of the model within the infectious period. To compare with the LAAD framework [33], we also considered F-beta, weighing Precision heavily over Recall ($\beta = 0.1$), where F-beta is represented as follows:

$$F\text{-}beta = \frac{(1 + beta^2)\, x\, Precision\, x\, Recall}{beta^2\, x\, Precision + Recall} \tag{8}$$

2. *Problem-Specific Metric:* In this study, the infectious period is considered to span from 7 days before the symptom onset to 21 days after the symptom onset. For presymptomatic detection of COVID infection, another significant metric to consider is the presymptomatic detection rate. It is the percentage of subjects the model identified as COVID-19-infected before the symptom onset.

3.3. Experiments on model parameters

Four experiments were done to determine the best properties for the PCovNet framework. Details specifications of these experiments are described in the following part.

3.3.1. Network depth

Increasing the number of layers in a neural network often results in more complex feature learning, but the model itself becomes prone to overfitting. We trained several networks with an increasing number of LSTM layers in both encoder and decoder blocks of the PCovNet model. For each case, we used augmentation on training data with the segmentation of 8 samples per window, and the model latent vector dimension is kept constant at 16. We evaluated 1 to 4 layers of depth for encoder and decoder LSTM layers with both thresholding methods. The detailed result of this experiment is given in Table 1.

From the above observations, we concluded that the MTE threshold method gave the best statistical metrics at depth 2 and the STE method showed the best problem-specific metrics (presymptomatic detection rate) at depth 2. So, we used PCovNet with 2 LSTM layers in both encoder and decoder blocks, and depending on the objective (presymptomatic COVID-19 detection or only anomalous RHR detection), the thresholding method was varied.

3.3.2. Latent space dimension

Similar to an autoencoder depth, setting the optimum dimension of the latent space is another significant experimental task for training a successful anomaly detector. The role of the latent space is to encode the underlying features from the training samples, which are used by the decoder to reconstruct the original input. Hence, increasing the dimension of the latent space creates more encoded features that tend to overfit the model. Although the VAE architecture has inherent regularization property, increasing the dimension of the latent space beyond a certain value still might not give the best performance. Conversely, decreasing the latent space dimension results in fewer encoded features, which beyond a certain point fails to represent the input data well enough; hence, the model performance degrades. Moreover, we used augmentation on the training dataset segmented at 8 samples per window and MTE as the thresholding method.

The experiment agreed with the prior theoretical assumption with varying latent space dimensions. According to Table 2, the best statistical and problem-specific metrics were found at the latent space dimension of 16. Moreover, increasing and decreasing the dimension of the latent layer from 16 degraded both metrics. Hence, we adopted the latent space dimension of 16 in our PCovNet LSTM-VAE model.

3.3.3. Window size

Setting the appropriate window size not always depends on the model, but varies based on the application domain. As mentioned in Subsection 3.2, the PCovNet framework resamples RHR into a 1-h period. We varied the window period per segment from 4 h to 12 h and evaluated the results. In each case, the model dimension (2 LSTM layers with a latent layer dimension of 16) and the thresholding method (MTE) are kept constant.

From the results shown in Table 3, we found that decreasing the window interval increased the Precision, Recall, and F-beta; however, the problem-specific metrics did not follow the same trend. The number of patients where the framework failed to detect was minimum with the window interval of 8 h, which declined with either the increase or decrease in the window size. On the other hand, window size 8 showed close performance to the best performance in terms of statistical metrics. As the primary target of this study was anomalous RHR detection, we set the window size to 8 h.

3.3.4. Effects of augmentation

The main challenge of using augmentation techniques in the time-series domain is the change in the inherent properties of the signal. On the other hand, deep learning models tend to learn better with the increasing volume of relevant train data. The PCovNet model is trained on individual subject data, where each sample window of the training dataset has 8 data points with a sampling interval of 1 h. As a result, for each train set, the number of raw samples is very low. Therefore, we experimented with 7 augmentation techniques which increased the train set volume by 7 times. For this experiment, we chose the window size of

Table 2
Evaluation of the framework varying the latent space dimension.

Latent layer	Precision	Recall	F-beta	Early Detection	Delayed Detection	Failed Detection
4	0.933	0.245	0.907	10	10	5
8	0.944	0.239	0.917	11	10	4
12	0.942	0.242	0.915	14	6	5
16	0.946	0.234	0.918	11	11	3
32	0.941	0.238	0.914	10	8	7
64	0.923	0.259	0.899	11	10	4

Table 1
Evaluation of the framework varying LSTM block depth and threshold method.

LSTM Layers	Threshold Method	Precision	Recall	F-beta	Early Detection	Delayed Detection	Failed Detection
128	MTE	0.94	0.217	0.909	12	9	4
128, 64	MTE	0.946	0.234	0.918	11	11	3
128, 64, 32	MTE	0.924	0.261	0.902	10	12	3
128, 64, 32, 16	MTE	0.911	0.276	0.890	12	11	2
128	STE	0.924	0.24	0.899	18	7	0
128, 64	STE	0.904	0.274	0.883	20	5	0
128, 64, 32	STE	0.906	0.284	0.887	15	10	0
128, 64, 32, 16	STE	0.891	0.3	0.874	18	7	0

Table 3
Evaluation of the framework varying the window size.

Window size	Precision	Recall	F-beta	Early Detection	Delayed Detection	Failed Detection
4	0.952	0.208	0.919	12	7	6
6	0.947	0.215	0.916	13	7	5
8	0.946	0.234	0.918	11	11	3
10	0.917	0.250	0.893	13	6	6
12	0.888	0.296	0.870	15	6	4
16	0.89	0.329	0.876	11	9	5

8 and kept the PCovNet model properties constant (2 LSTM layers with 128 and 64 units in both encoder and decoder, latent space dimension of 16).

The results in Table 4 showed that augmentation increased all metrics for both threshold methods. In our task of aberrant RHR detection, the anomaly was mostly dependent on the RHR value fluctuations compared to the baseline RHR.

Prior literature [17,18] suggested that COVID-19 causes elevation, declination, and abrupt changes in one's RHR from the baseline. Therefore, we assumed that the baseline RHR modified within a very limited range by the augmentation techniques was less likely to misrepresent the baseline RHR. On the other hand, such careful augmentation would result in a larger train set aiding the model performance. The above comparison was certainly commensurate with the prior assumptions.

3.4. PCovNet model

Based on the results of our experiments, we built our PCovNet framework with the LSTM layer depth of 2, latent space dimension of 16, and the window size of 8 for making the train and test set, and employed the augmentation techniques. Moreover, as mentioned in Section 3.3.1, we considered both MTE and STE threshold methods for evaluation because each showed superiority in performance over the other based on the metric type. Furthermore, there were 225 trainable and 6 non-trainable parameters in the model. The detailed architecture of our PCovNet LSTM-VAE model is illustrated in Fig. 5.

As an example of the anomalous RHR detection process and the model training, Fig. 6 illustrated the RHR, anomaly detection model losses, loss distribution, and loss curve of a COVID-19 infected subject (id - ASFODQR). Fig. 6a and b were aligned along the x-axis as per date to relate the RHR changes with the model losses. From top to bottom, the figures depicted the RHR, model losses with MTE threshold, and model losses with STE threshold, respectively. The anomaly predictions for all COVID-19 subjects for the MTE and the STE methods are respectively shown in Supplementary Appendix 7 and 8. Fig. 6c illustrated the loss distribution, which showed both threshold methods and how the anomalous RHR was divided from the normal one based on the losses. Although the MTE threshold line was leading for this subject, the STE threshold also led in several subjects (Supplementary Appendix 6). Although the maximum epoch was set to 1000, according to Fig. 6d, the model trained a little over 400 epochs for this subject. This was the result of the early stopping criterion to avoid overfitting. The actual number of epochs also varied from subject to subject, which was shown in Supplementary Appendix 5.

3.5. Experiments on PCovNet framework performance

As mentioned in Section 2.1, our study consisted of 25 COVID-19, 10 Other illness, and 67 Healthy subjects. We evaluated the performance of PCovNet with both MTE and STE threshold methods on these three groups.

3.5.1. COVID-19 group

As per the literature, the RHR of COVID-19 infected individuals tends to deviate from the baseline RHR. The PCovNet framework also found notable anomalous RHR per day for most of the COVID-19 subjects during the infectious period. The anomalous RHR detection rate per day was illustrated in Fig. 6a along with the infectious period and the symptom onset date. The STE threshold method was more sensitive to the MTE one, which resulted in more presymptomatic detection as shown in Table II. Conversely, the MTE method showed a less anomalous RHR detection rate but failed to detect any anomaly for 3 subjects (Fig. 7a, Table 1). Anomaly predictions for the Other illness and Healthy groups were shown in Supplementary Appendix 2 and 3 respectively.

Fig. 7b illustrated the first anomaly detection for each COVID-19 subject in the span of the infectious period for both the MTE and the STE threshold method. According to the figure, the MTE method showed 44% presymptomatic (11 of 25), 44% post-symptomatic (11 of 25), and 12% (3 of 25) failed cases. On the other hand, the STE method resulted in 80% presymptomatic (20 of 25), 20% post-symptomatic (5 of 25), and no failed classes.

3.5.2. COVID-19 vs other illness vs healthy groups

We trained and evaluated PCovNet for 25 COVID-19 and 10 Other illness subjects along with 67 Healthy subjects individually. The COVID-19 and the Other illness groups had predefined symptom onset dates in the provided dataset; however, the Healthy group does not have symptom onset dates. To evaluate, whether the model misidentified healthy RHR sequences as anomalous, we took a baseline period of 23 days from the beginning of the RHR sequences of each Healthy subject. Afterward, we set the symptom onset at the 66.7 percentile of the remaining sequence. Fig. 8 showed the combined F-beta scores for each group for MTE and STE threshold methods. The box plot showed that the median of the COVID-19 subjects is the greatest of the three groups. On the other hand, the Healthy group showed F-beta spanning from 0 to 1 for both MTE and STE threshold methods with a median approximately at 0.5. Such performance signified that the model does not effectively detect anomalies within our imaginary infectious period with normal RHR, which is desired for a good anomaly detector.

Moreover, Fig. 9a illustrated the elevated and decreased RHR from the baseline periods of three groups, which were calculated based on the RHR sequences of the subjects. This figure showed that COVID-19 subjects had the greatest median elevated RHR, whereas the lowest median decreased RHR was observed for the Other illness group. Additionally, Fig. 9c showed the RHR difference from the baseline for each subject within these two groups. We can that only one subject (AV2GF3B) among the COVID-19 group showed a significantly high RHR elevation compared to all other subjects. For all other subjects within these two groups, the RHR differences seemed comparable. This proves that only RHR differences might not be a good metric to differentiate COVID-19 from Other illness with the given data.

Table 4
Evaluation of the framework based on augmentation and threshold method.

Threshold	Augmentation	Precision	Recall	F-beta	Early Detection	Delayed Detection	Failed Detection
MTE	No	0.915	0.268	0.893	10	10	5
MTE	Yes	0.942	0.237	0.915	11	12	2
STE	No	0.892	0.289	0.874	16	8	1
STE	Yes	0.917	0.271	0.896	20	5	0

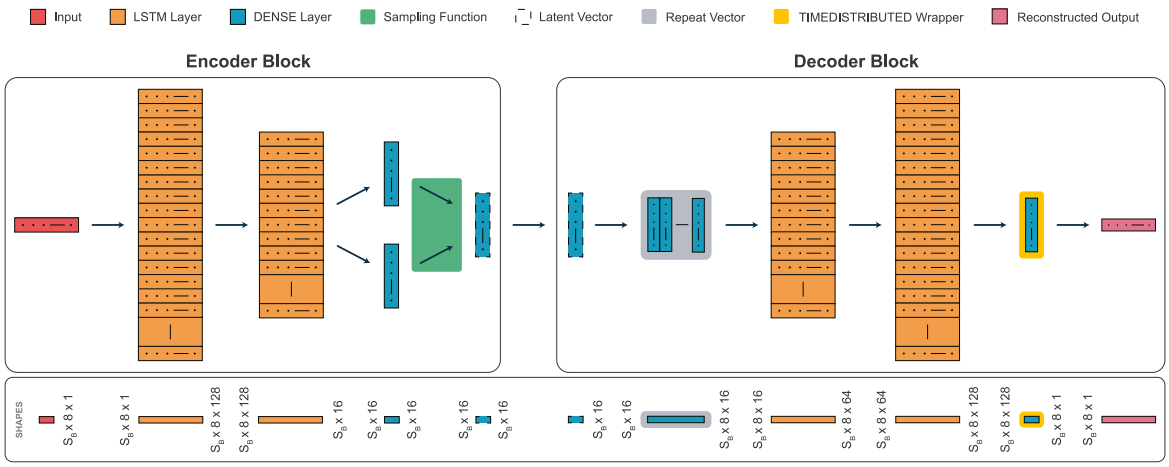


Fig. 5. The detailed architecture of PCovNet LSTM-VAE architecture.

However, Healthy group box plots in Fig. 9a showed both elevated and decreased RHR, but they were closer to the baseline compared to the other two groups. A detailed RHR difference of the Healthy group was illustrated in Supplementary Appendix 4 which showed a very little deviance of RHR from the baseline. The distributions of the detected anomalous RHR duration by the model were depicted in Fig. 9c. Two sides of the violin plots showed the distributions for MTE and STE threshold methods. The violin plot of the COVID-19 group stretched the farthest. On the other hand, whereas the medians of COVID-19 and Other illness distributions were within 50–100, the medians of the Healthy distributions were close to zero. This reconfirmed the model's ability not to label the normal RHR during the imaginary infectious period as an anomalous one. Moreover, 9d illustrated the total anomalous RHR hours per subject during the infectious period for COVID-19 and Other illness groups. Although total anomalous RHR was higher for 6 COVID-19 subjects, others did not show any significant differences from the Other illness group. This further reconfirms the model's inability to detect COVID-19 exclusively among different illnesses. On the other hand, the detected anomalous hours are far less for the Healthy subjects than for the rest (Supplementary Appendix 4).

4. Discussion

The PCovNet framework was trained on a subset of a phase 1 study performed by the Genetics Department of Stanford University, where among the cohort of 5262 subjects only a fraction of subjects had smartwatch data during COVID-19 infection. However, they only made the 32 Fitbit users' data with COVID-19 infection publicly available with the symptom onset information [40]. However, among the given 32 COVID-19 infected subjects, only 25 met our study criterion mentioned in Section 2.1. For the train set, we identified the baseline RHR based on prior literature. Since there was no clear way of determining the anomalous ground truths, we took the infectious period, defined in several prior research, as the anomalous period [41–45]. As a result, we prioritized the Precision score, which quantifies the model's performance in detecting anomalies within the infectious period [33]. As the actual infectious period for each subject may vary, the Recall scores do not represent a good insight into the model's performance, and in practice, they were quite low. Hence, we calculated the F-beta score penalizing the Recall instead of F1. However, compared to the other deep learning-based work on this dataset, PCovNet (MTE) showed an increase in Precision, F-beta scores, and PCovNet (STE) showed a higher presymptomatic anomaly detection rate.

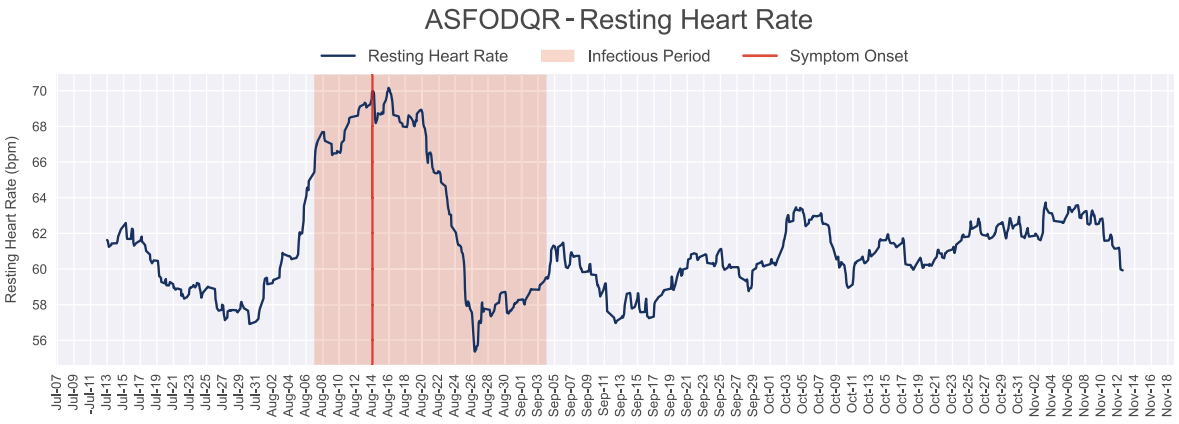
4.1. Comparison with previous study

We compared the PCovNet framework with the first deep learning-based presymptomatic COVID detection framework, LAAD [33], which was also individually trained on the 25 COVID-positive subjects of the same dataset. At first, the statistical matrices, namely TP, FP, TN, FN, Precision, Recall, and F-beta were calculated for each subject individually. For reporting the overall score, the authors of LAAD averaged all 25 instances but ignored the ones where the framework failed to make any successful anomaly detection. In our investigation based on their per-subject reported metrics, those two failed cases result in zero or undefined values. Hence, ignoring these failed cases would make the overall reporting unrepresentative of the whole study. To circumvent this problem, we took an alternative approach of calculating the overall TP, FP, TN, and FN by summing them for each subject and then calculating Precision, Recall, and F-beta from those four matrices. For a fair comparison, we recalculated the statistical metrics based on the per-subject TP, FP, TN, and FN scores provided in the LAAD article [33]. We also reproduced the results of the LAAD model from their pipeline provided on GitHub. Although we could not get the same result due to the stochastic nature of the deep learning models, our reproduced results were close to their reported ones.

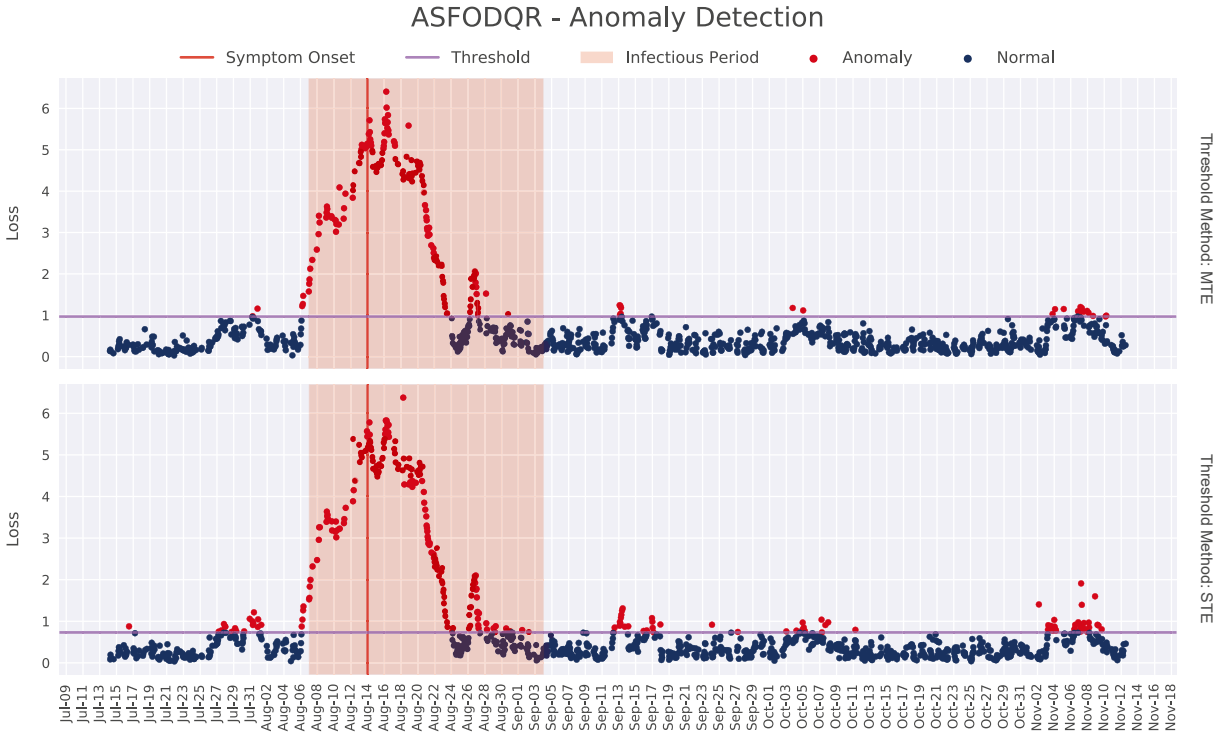
The dataset only provided the symptom onset date for each COVID-19 infected individual, but no actual infection date. Hence, we relied on the literature [41–45] to derive the infectious period and formulated the experiments accordingly. However, in reality, this infectious period may vary for each subject. Based on the definitions of Precision and Recall mentioned in Equations (6) and (7), our objective is to optimize the model so that most of the predictions are made in the infectious period. Therefore, among the statistical metrics, we put more weight on Precision than on Recall and calculated F-beta accordingly.

From the comparison presented in Table V, the PCovNet model with the MTE threshold method performed the best in statistical metrics. Moreover, it resulted in an above 5% Precision increase and above 2% F-beta increase from the reported LAAD results. Conversely, the STE threshold method showed a slight increase in Precision, but a slight decrease in Recall and F-beta. In terms of failed detection cases, PCovNet with the STE threshold method detected anomalous RHR for all subjects, whereas the LAAD failed for two subjects. Moreover, the presymptomatic detection was also increased by 6 subjects compared to LAAD. On the other hand, the MTE threshold method showed a decline in presymptomatic detection by three subjects and failed detection by one. Overall, the MTE threshold method with PCovNet showed a notable increase from LAAD in terms of statistical metrics and the STE threshold method showed significant development in presymptomatic detection.

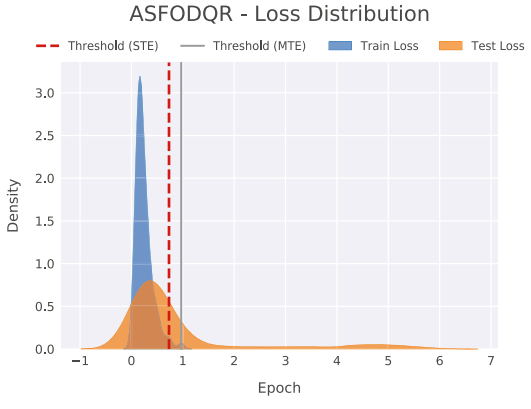
From the architectural point of view, in contrast to the autoencoder



(a)



(b)

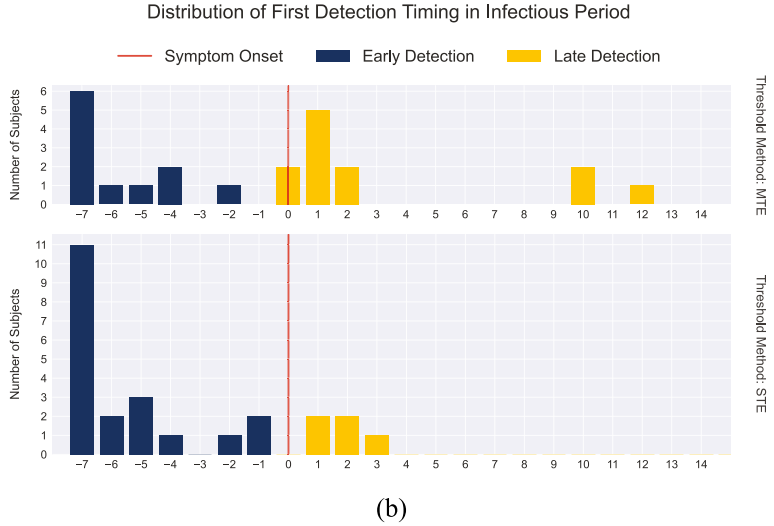


(c)

(d)

Fig. 6. Training summary of subject ID ASFODQR (a) The Resting Heart Rate (b) Anomaly Detector Losses for MTE and STE threshold (c) Training and testing loss distribution (d) Loss curve during the LSTM-VAE training.

(a)



(b)

Fig. 7. (a) The top and bottom heatmaps illustrate the anomalous RHR detection rate per day for each user for MTE and STE threshold methods, respectively. (b) The bar charts showed the number of subjects identified with anomalous RHR in the infectious period for the first time. The presymptomatic and post-symptomatic detection are indicated in varying colors.

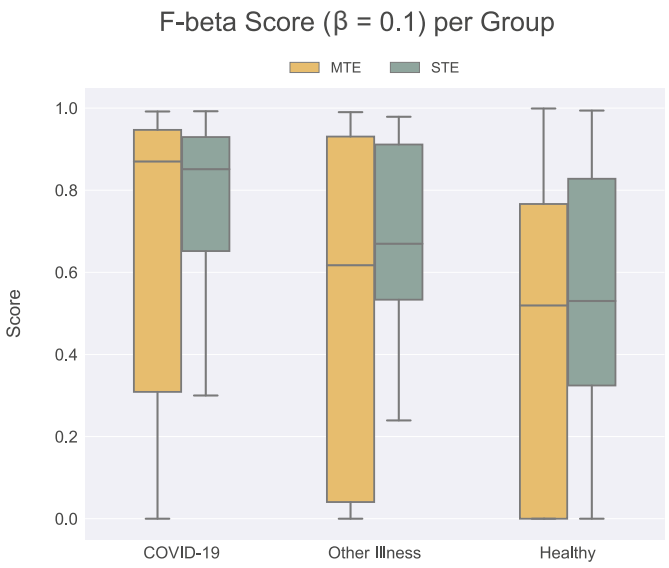


Fig. 8. Comparison of F-beta Scores among three group.

(AE) in the LAAD framework, the variational autoencoder in PCovNet introduces regularization and continuous latent space representation. Although the AE and VAE are generative networks by architecture, and both have encoder and decoder units at their core, their latent vector (encoder output) are different. For AE, the latent vector often contains discrete sets of values resulting in an irregular latent space. During decoding, any value from the irregular latent space can result in meaningless output. Moreover, this phenomenon often results in overfitting. Conversely, the VAE encoder maps the latent vector into a gaussian distribution resulting in a continuous latent space. Alternatively, such continuity introduces explicit regularization when the model is in the training phase [59]. Thus, due to the inherent regulation of the VAE architecture in PCovNet, the model learns the important features better and shows better decoding during the testing process in contrast to the LAAD. This is validated by the results in Table V.

Although a few other recent studies have validated smartwatch-based COVID-19 detection systems, their study objective, experiment setup, evaluation criteria, and dataset are not the same as this experiment. Hence, we could not compare these quantitative results with ours. In one such study, Liu et al. proposed a contrastive convolutional autoencoder-based system on the HRV data. The method achieved unweighted average specificity of 90.6%, sensitivity of 100%, and recall of 95.3% [34]. In another study, Hijazi et al. extracted both time and frequency domain features from the HRV data collected by smartwatches and predicted COVID-19 infection at least two days before the symptom onset. The combination of HRV and survey-based assessment features reached maximum Precision, Recall, and F-score of 0.91, 0.88, and 0.89, respectively [60]. While these studies aimed at the abnormal HRV detection associated with COVID-19 infection, the experiment on pre-symptomatic detection and detailed investigation of Other Illness and Healthy groups are missing.

4.2. Utility of PCovNet

According to the results shown in Section 3.5, RHR elevation was spotted among the COVID-19 and Other Illness groups as mentioned in prior studies [17,18,22]. However, the results in Section 3.5.1 showed the performances of the two threshold methods and among them, STE showed higher sensitivity compared to MTE. The high sensitivity of the STE method might result in false alarming, which could be filtered by the anomaly strength or the anomaly duration for an alarming system. On the other hand, the high Precision rate is generally desirable for a detection system not used with duration-based post-processing. Hence, based on the objective of an application, either MTE or STE methods can be adopted.

Moreover, the experiment with COVID-19, Other Illness, and Healthy groups signifies that, despite the elevation and decrease in RHR for the Healthy group within the demo infectious period, the elevation pattern is more prominent in the other two groups. However, similar to the previous studies [22,33], the RHR elevation patterns in COVID-19 and Other Illness groups are not conclusive of a certain infection in PCovNet. This signifies that PCovNet, like other wearable-based systems, is not useable as the sole diagnosis tool for COVID-19, and certainly cannot replace laboratory-based testing, i.e., RT-PCR.

However, as an everyday companion, smartwatches can monitor the RHR continuously and the results showed potential for PCovNet to detect anomalous RHR directly related to COVID-19 infection. Thus, this system can be used as a tool for continuous monitoring and contact tracing, which is the primary concern with laboratory-based systems. Upon getting the alerts from PCovNet enabled smartwatches, the individuals can take the laboratory test and isolate themselves, which has immense utility for prioritizing laboratory testing during resource-constraint scenarios as well.

4.3. Future directions

Although we have achieved a promising result to use PCovNet with smartwatches for detecting COVID-19 before the symptom onset, there are several future directions that can be explored to make the system more robust.

Firstly, a larger dataset is needed to make such a framework more reliable. The greater dataset volume should include a larger baseline sequence as well. Due to the nature of the study, a large number of annotated data having the COVID-19 infection timeframe is a mammoth task. For example, we got only 25 subject data to work with among 3325 Fitbit users in the cohort. Hence a collaborative approach is necessary among the leading wearable device manufacturers and researchers.

Secondly, this study included only the COVID-19 infected individuals who experienced aberrant RHR due to SARS-CoV-2 infection. However, a good number of infections might cause RHR abnormalities. Prior works [22,33] showed, and we verified that subjects with COVID-19 infection and other illnesses show a similar elevation, declination, and abrupt fluctuations in RHR. The probable reason behind this failure might be the HR sensing ability of the wearable device sensors or the lack of infection effects on the HR. This can be further examined with a larger cohort; however, there is a financial and technological bottleneck for implementing high-quality HR sensors in wearable devices for such a large cohort.

Thirdly, although the dataset contained sleep information for the COVID-19-infected subjects, it was irregular and sparse. Moreover, the sleep stages can be a significant parameter for detecting diseases, which the wearable sensors often cannot determine reliably [22].

5. Conclusion

This study proposes an anomaly detection framework with LSTM-VAE to identify the underlying COVID-19 infection during the pre-symptomatic period based on the RHR of 25 COVID-positive subjects. The RHR was derived during the preprocessing steps of the framework using the HR and steps data collected by the Fitbit smartwatch. The model was trained and evaluated separately for each subject. However, the framework reported two thresholding parameters each showing better performance in two separate priorities - detecting anomalous RHR in the infectious period (presymptomatic and post-symptomatic detection) and detecting anomalous RHR before the symptom onset.

Overall, this study showed that despite the sensing constraint of wearable devices, they have the potential for continuous health monitoring. Although any diagnosis solely based on such devices is still not reliable, they can act as alerting companions for immediate diagnosis.

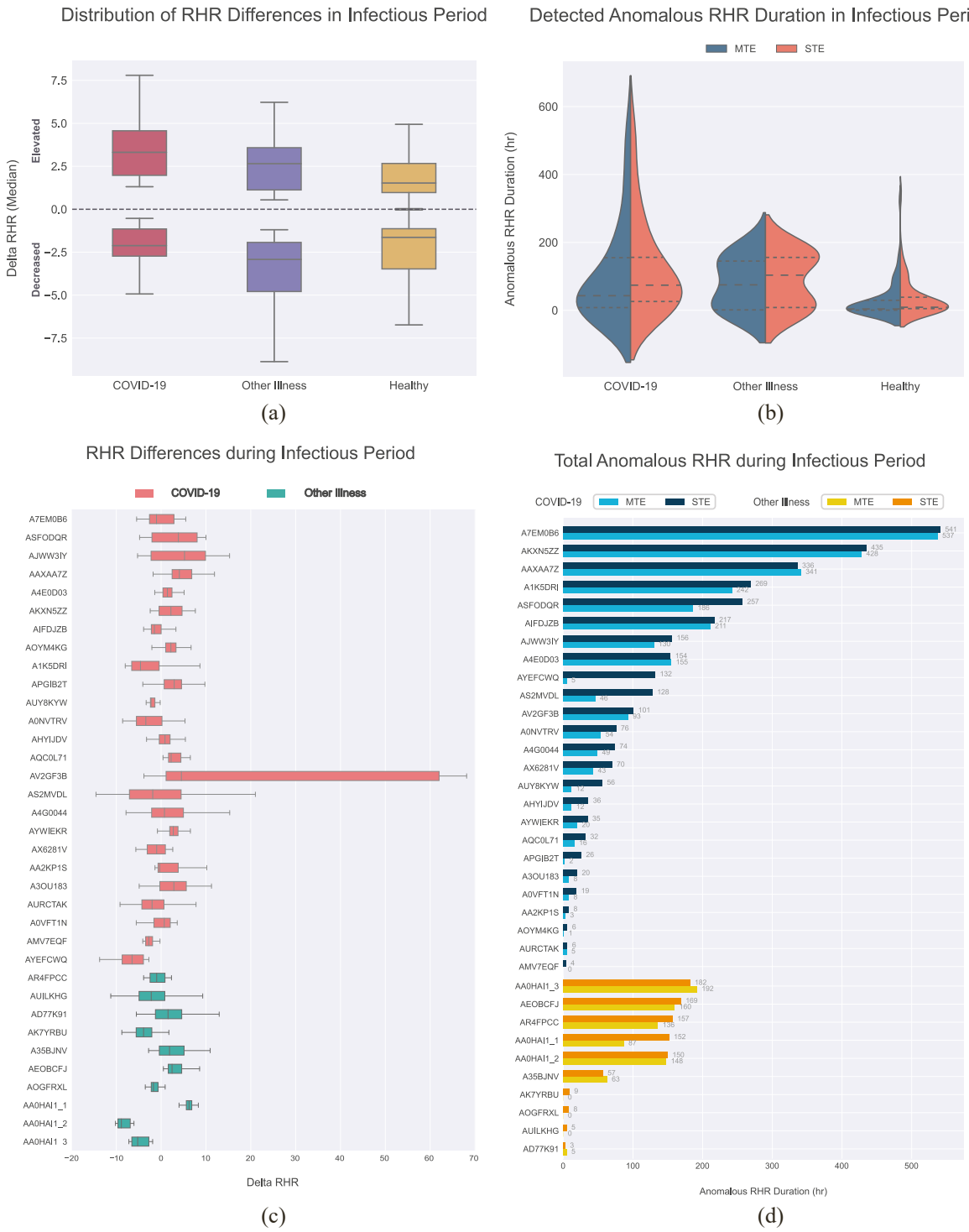


Fig. 9. (a) The distribution of elevated and decreased RHR during the infectious period for all three groups (b) Violin plot of the anomalous RHR hours of all subjects during the infectious period for all three groups (c) Box plot of the difference in RHR per subject of COVID-19 and Other illness group (d) Bar plot of the total anomalous RHR hours per subject of COVID-19 and Other illness group.

Table 5
Comparison of PCovNet with the previous work.

Models	Precision	Recall	F-beta	Early Detection	Delayed Detection	Failed Detection
LAAD	0.894	0.331	0.895	14	9	2
LAAD (Reproduced)	0.887	0.307	0.87	12	10	3
PCovNet (STE)	0.904	0.274	0.883	20	5	0
PCovNet (MTE)	0.946	0.234	0.918	11	11	3

We are optimistic that more comprehensive studies on wearable devices will enable us to contain infectious diseases more efficiently in the future.

Funding

This work was supported by the Qatar National Research Grant: UREP28-144-3-046. The statements made herein are solely the responsibility of the authors.

Statement of originality

The authors declare that they generated all figures used in this work.

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.

Acknowledgement

The authors would like to show our gratitude to Mishra et al. [22] for providing their study data.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.compbiomed.2022.105682.

References

[1] WHO, Coronavirus (COVID-19) dashboard, n.d. https://covid19.who.int, April 6, 2022. accessed
[2] M. Al-Zinati, T. Almasri, M. Alsmirat, Y. Jararweh, Enabling multiple health security threats detection using mobile edge computing, Simulat. Model. Pract. Theor. 101 (2020), 101957.
[3] V. Haldane, C. De Foo, S.M. Abdalla, A.-S. Jung, M. Tan, S. Wu, A. Chua, M. Verma, P. Shrestha, S. Singh, others, Health systems resilience in managing the COVID-19 pandemic: lessons from 28 countries, Nat. Med. (2021) 1–17.
[4] H. Mazumder, M.M. Hossain, A. Das, Geriatric care during public health emergencies: lessons learned from novel corona virus disease (COVID-19) pandemic, J. Gerontol. Soc. Work 63 (2020) 257–258.
[5] Z. Wu, D. Harrich, Z. Li, D. Hu, D. Li, The unique features of SARS-CoV-2 transmission: comparison with SARS-CoV, MERS-CoV and 2009 H1N1 pandemic influenza virus, Rev. Med. Virol. 31 (2021), e2171.
[6] Z. Abdelrahman, M. Li, X. Wang, Comparative review of SARS-CoV-2, SARS-CoV, MERS-CoV, and influenza a respiratory viruses, Front. Immunol. 11 (2020) 2309.
[7] M. Gandhi, D.S. Yokoe, D.V. Havlir, Asymptomatic Transmission, the Achilles' Heel of Current Strategies to Control Covid-19, 2020.
[8] O. Byambasuren, M. Cardona, K. Bell, J. Clark, M.-L. McLaws, P. Glasziou, Estimating the extent of asymptomatic COVID-19 and its potential for community transmission: systematic review and meta-analysis, Official Journal of the Association of Medical Microbiology and Infectious Disease Canada 5 (2020) 223–234.
[9] Y. Bai, L. Yao, T. Wei, F. Tian, D.-Y. Jin, L. Chen, M. Wang, Presumed asymptomatic carrier transmission of COVID-19, JAMA 323 (2020) 1406–1407.
[10] K. Mizumoto, K. Kagaya, A. Zarebski, G. Chowell, Estimating the asymptomatic proportion of coronavirus disease 2019 (COVID-19) cases on board the Diamond Princess cruise ship, Yokohama, Japan, 2020, Euro Surveill. 25 (2020), 2000180.
[11] R. Kinoshita, A. Anzai, S. Jung, N.M. Linton, T. Miyama, T. Kobayashi, K. Hayashi, A. Suzuki, Y. Yang, A.R. Akhmetzhanov, others, Containment, contact tracing and asymptomatic transmission of novel Coronavirus disease (COVID-19): a modelling study, J. Clin. Med. 9 (2020) 3125.
[12] J. Hellewell, T.W. Russell, R. Beale, G. Kelly, C. Houlihan, E. Nastouli, A. J. Kucharski, Estimating the effectiveness of routine asymptomatic PCR testing at different frequencies for the detection of SARS-CoV-2 infections, BMC Med. 19 (2021) 1–10.
[13] A. Tahamtan, A. Ardebili, Real-time RT-PCR in COVID-19 detection: issues affecting the results, Expert Rev. Mol. Diagn. 20 (2020) 453–454.
[14] Y. Zhou, F. Pei, M. Ji, L. Wang, H. Zhao, H. Li, W. Yang, Q. Wang, Q. Zhao, Y. Wang, Sensitivity evaluation of 2019 novel coronavirus (SARS-CoV-2) RT-PCR detection kits and strategy to reduce false negative, PLoS One 15 (2020), e0241469.
[15] T. Rahman, A. Khandakar, M.E. Hoque, N. Ibtehaz, S.B. Kashem, R. Masud, L. Shampa, M.M. Hasan, M.T. Islam, S. Al-Maadeed, Others, development and validation of an early scoring system for prediction of disease severity in COVID-19 using complete blood count parameters, IEEE Access 9 (2021) 120422–120441.
[16] T. Rahman, F.A. Al-Ishaq, F.S. Al-Mohannadi, R.S. Mubarak, M.H. Al-Hitmi, K. R. Islam, A. Khandakar, A.A. Hssain, S. Al-Madeed, S.M. Zughaier, Others, mortality prediction utilizing blood biomarkers to predict the severity of COVID-19 using machine learning technique, Diagnostics 11 (2021) 1582.
[17] A. Ponomarev, K. Tyapochkin, E. Surkova, E. Smorodnikova, P. Pravdin, Heart Rate Variability as a Prospective Predictor of Early COVID-19 Symptoms, MedRxiv, 2021.
[18] R. Buchhorn, C. Baumann, C. Willaschek, Heart rate variability in a patient with coronavirus disease 2019, in: International Cardiovascular Forum Journal, 2020.
[19] A. Mahajan, G. Pottie, W. Kaiser, Transformation in healthcare by wearable devices for diagnostics and guidance of treatment, ACM Transactions on Computing for Healthcare 1 (2020) 1–12.
[20] L. Piwek, D.A. Ellis, S. Andrews, A. Joinson, The rise of consumer health wearables: promises and barriers, PLoS Med. 13 (2016), e1001953.
[21] S.M. Lee, D. Lee, Healthcare wearable devices: an analysis of key factors for continuous use intention, Service Business 14 (2020) 503–531.
[22] T. Mishra, M. Wang, A.A. Metwally, G.K. Bogu, A.W. Brooks, A. Bahmani, A. Alavi, A. Celli, E. Higgs, O. Dagan-Rosenfeld, others, Pre-symptomatic detection of COVID-19 from smartwatch data, Nature. Biomed. Eng. 4 (2020) 1208–1220.
[23] J.M. Radin, N.E. Wineinger, E.J. Topol, S.R. Steinhubl, Harnessing wearable device data to improve state-level real-time surveillance of influenza-like illness in the USA: a population-based study, The Lancet Digital Health 2 (2020) e85–e93.
[24] G. Quer, J.M. Radin, M. Gadaleta, K. Baca-Motes, L. Ariniello, E. Ramos, V. Kheterpal, E.J. Topol, S.R. Steinhubl, Wearable sensor data and self-reported symptoms for COVID-19 detection, Nat. Med. 27 (2021) 73–77.
[25] B.L. Smarr, K. Aschbacher, S.M. Fisher, A. Chowdhary, S. Dilchert, K. Puldon, A. Rao, F.M. Hecht, A.E. Mason, Feasibility of continuous fever monitoring using wearable devices, Sci. Rep. 10 (2020) 1–11.
[26] S. Lalmuanawma, J. Hussain, L. Chhakchhuak, Applications of machine learning and artificial intelligence for Covid-19 (SARS-CoV-2) pandemic: a review, Chaos, Solitons & Fractals 139 (2020), 110059.
[27] J. Chen, L. Wu, J. Zhang, L. Zhang, D. Gong, Y. Zhao, Q. Chen, S. Huang, M. Yang, X. Yang, others, Deep learning-based model for detecting 2019 novel coronavirus pneumonia on high-resolution computed tomography, Sci. Rep. 10 (2020) 1–11.
[28] Y. Qiblawey, A. Tahir, M.E. Chowdhury, A. Khandakar, S. Kiranyaz, T. Rahman, N. Ibtehaz, S. Mahmud, S.A. Maadeed, F. Musharavati, others, Detection and severity classification of COVID-19 in CT images using deep learning, Diagnostics 11 (2021) 893.
[29] Z. Wang, Y. Xiao, Y. Li, J. Zhang, F. Lu, M. Hou, X. Liu, Automatically discriminating and localizing COVID-19 from community-acquired pneumonia on chest X-rays, Pattern Recogn. 110 (2021), 107613.
[30] A. Tahir, Y. Qiblawey, A. Khandakar, T. Rahman, U. Khurshid, F. Musharavati, M. Islam, S. Kiranyaz, M. Chowdhury, Deep Learning for Reliable Classification of COVID-19, MERS, and SARS from Chest X-Ray Images, 2021.
[31] M. Yamaç, M. Ahishali, A. Degerli, S. Kiranyaz, M.E. Chowdhury, M. Gabbouj, Convolutional sparse support estimator-based COVID-19 recognition from X-ray images, IEEE Transact. Neural Networks Learn. Syst. 32 (2021) 1810–1820.
[32] J.D. Arias-Londoño, J.A. Gómez-García, L. Moro-Velázquez, J.I. Godino-Llorente, Artificial Intelligence applied to chest X-Ray images for the automatic detection of COVID-19. A thoughtful evaluation approach, IEEE Access 8 (2020) 226811–226827.
[33] G.K. Bogu, M.P. Snyder, Deep Learning-Based Detection of COVID-19 Using Wearables Data, MedRxiv, 2021.
[34] S. Liu, J. Han, E.L. Puyal, S. Kontaxis, S. Sun, P. Locatelli, J. Dineley, F.B. Pokorny, G. Dalla Costa, L. Leocani, Others, fitbeat: COVID-19 estimation based on wristband heart rate using a contrastive convolutional auto-encoder, Pattern Recogn. (2021), 108403.
[35] A. Chharia, R. Upadhyay, V. Kumar, C. Cheng, J. Zhang, T. Wang, M. Xu, Deep-precognitive diagnosis: preventing future pandemics by novel disease detection with biologically-inspired conv-fuzzy network, IEEE Access 10 (2022) 23167–23185.
[36] Y. Xue, Y. Wang, J. Liang, A. Slowik, A self-adaptive mutation neural architecture search algorithm based on blocks, IEEE Comput. Intell. Mag. 16 (2021) 67–78.
[37] Y. Xue, P. Jiang, F. Neri, J. Liang, A multi-objective evolutionary approach based on graph-in-graph for neural architecture search of convolutional neural networks, Int. J. Neural Syst. 31 (2021), 2150035.
[38] D. O'Neill, B. Xue, M. Zhang, Evolutionary neural architecture search for high-dimensional skip-connection structures on densenet style networks, IEEE Trans. Evol. Comput. 25 (2021) 1118–1132.
[39] D.P. Kingma, M. Welling, An Introduction to Variational Autoencoders, 2019. ArXiv Preprint ArXiv:1906.02691.
[40] COVID-19 Wearables Data, 2020. https://storage.googleapis.com/gbsc-gcp-project-ipop_public/COVID-19/COVID-19-Wearables.zip.
[41] W. Guan, Z. Ni, Y. Hu, W. Liang, C. Ou, J. He, L. Liu, H. Shan, C. Lei, D.S. Hui, Others, Clinical Characteristics of 2019 Novel Coronavirus Infection in China, MedRxiv, 2020.
[42] S.A. Lauer, K.H. Grantz, Q. Bi, F.K. Jones, Q. Zheng, H.R. Meredith, A.S. Azman, N. G. Reich, J. Lessler, The incubation period of coronavirus disease 2019 (COVID-19) from publicly reported confirmed cases: estimation and application, Ann. Intern. Med. 172 (2020) 577–582.
[43] Q. Li, X. Guan, P. Wu, X. Wang, L. Zhou, Y. Tong, R. Ren, K.S. Leung, E.H. Lau, J. Y. Wong, et al., Early Transmission Dynamics in Wuhan, China, of Novel Coronavirus–Infected Pneumonia, New England Journal of Medicine 382 (13)

(2020) 1199–1207, https://doi.org/10.1056/NEJMoa2001316. https://doi.org/10.1056/NEJMoa2001316.
[44] M. Cevik, M. Tate, O. Lloyd, A.E. Maraolo, J. Schafers, A. Ho, SARS-CoV-2, SARS-CoV, and MERS-CoV viral load dynamics, duration of viral shedding, and infectiousness: a systematic review and meta-analysis, The Lancet Microbe 2 (1) (2020) e13–e22, https://doi.org/10.1016/S2666-5247(20)30172-5.
[45] X. He, E.H. Lau, P. Wu, X. Deng, J. Wang, X. Hao, Y.C. Lau, J.Y. Wong, Y. Guan, X. Tan, others, Temporal dynamics in viral shedding and transmissibility of COVID-19, Nat. Med. 26 (2020) 672–675.
[46] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, others, Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[47] T.T. Um, F.M. Pfister, D. Pichler, S. Endo, M. Lang, S. Hirche, U. Fietzek, D. Kulić, Data augmentation of wearable sensor data for Parkinson's disease monitoring using convolutional neural networks, in: Proceedings of the 19th ACM International Conference on Multimodal Interaction, 2017, pp. 216–220.
[48] B.K. Iwana, S. Uchida, Time series data augmentation for neural networks by time warping with a discriminative teacher, in: 2020 25th International Conference on Pattern Recognition (ICPR), IEEE, 2021, pp. 3558–3565.
[49] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, L. Bottou, Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (2010).
[50] F.A. Gers, D. Eck, J. Schmidhuber, Applying LSTM to time series predictable through time-window approaches, in: Neural Nets WIRN Vietri-01, Springer, 2002, pp. 193–200.
[51] D.P. Kingma, J. Ba, Adam, A Method for Stochastic Optimization, 2014. ArXiv Preprint ArXiv:1412.6980.
[52] F. Chollet, Keras, 2015 others, https://github.com/fchollet/keras.
[53] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, et al., TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems, ArXiv Preprint ArXiv:1603.04467 (2016), https://doi.org/10.48550/arXiv.1603.04467. http://arxiv.org/abs/1603.04467.
[54] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, R. Kern, M. Picus, S. Hoyer, M.H. van Kerkwijk, M. Brett, A. Haldane, J.F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T.E. Oliphant, Array programming with NumPy, Nature 585 (2020) 357–362, https://doi.org/10.1038/s41586-020-2649-2.
[55] The Pandas Development Team, Pandas-Dev/pandas: Pandas, Zenodo, 2020, https://doi.org/10.5281/zenodo.3509134.
[56] W. McKinney, in: T.X. Austin (Ed.), Data Structures for Statistical Computing in python, 2010, pp. 51–56.
[57] J.D. Hunter, Matplotlib: a 2D graphics environment, Comput. Sci. Eng. 9 (2007) 90–95.
[58] M.L. Waskom, Seaborn: statistical data visualization, J. Open. Source. Soft 6 (2021) 3021.
[59] C. Doersch, Tutorial on Variational Autoencoders, 2016. ArXiv Preprint ArXiv: 1606.05908.
[60] H. Hijazi, M. Abu Talib, A. Hasasneh, A. Bou Nassif, N. Ahmed, Q. Nasir, Wearable devices, smartphones, and interpretable Artificial intelligence in combating COVID-19, Sensors 21 (2021) 8424, https://doi.org/10.3390/s21248424.