CovidDeep: SARS-CoV-2/COVID-19 Test Based on Wearable Medical Sensors and Efficient Neural Networks

Shayan Hassantabar, Novati Stefano, Vishweshwar Ghanakota, Alessandra Ferrari, Gregory N. Nicola, Raffaele Bruno, Ignazio R. Marino, Kenza Hamidouche, and Niraj K. Jha, *Fellow, IEEE*

***Abstract*—The novel coronavirus (SARS-CoV-2) has led to a pandemic. The current testing regime based on Reverse Transcription-Polymerase Chain Reaction for SARS-CoV-2 has been unable to keep up with testing demands, and also suffers from a relatively low positive detection rate in the early stages of the resultant COVID-19 disease. Hence, there is a need for an alternative approach for repeated large-scale testing of SARS-CoV-2/COVID-19. The emergence of wearable medical sensors (WMSs) and deep neural networks (DNNs) points to a promising approach to address this challenge. WMSs enable continuous and user-transparent monitoring of physiological signals. However, disease detection based on WMSs/DNNs and their deployment on resource-constrained edge devices remain challenging problems. To address these problems, we propose a framework called CovidDeep that combines efficient DNNs with commercially available WMSs for pervasive testing of the virus and the resultant disease. CovidDeep does not depend on manual feature extraction. It directly operates on WMS data and some easy-to-answer questions in a questionnaire whose answers can be obtained through a smartphone application. We collected data from 87 individuals, spanning three cohorts including healthy, asymptomatic (to detect the virus), and symptomatic (to detect the disease) patients. We trained DNNs on various subsets of the features automatically extracted from six WMS and questionnaire categories to perform ablation studies to determine which subsets are most efficacious in terms of test accuracy for a three-way classification. The highest test accuracy obtained was 98.1%. The models were also shown to perform well on other performance measures, such as false positive rate, false negative rate, and F1 score. We augmented the real training dataset with a synthetic training dataset drawn from the same probability distribution to impose a prior on DNN weights and leveraged a grow-and-prune synthesis paradigm to learn both DNN architecture and weights. This boosted the accuracy of the various DNNs further and simultaneously reduced their size and floating-point operations. This makes the CovidDeep DNNs both accurate and efficient, in terms of memory requirements and computations. The resultant DNNs are embedded in a smartphone application, which has the added benefit of preserving patient privacy.**

***Index Terms*—COVID-19 test, deep neural network (DNN), grow-and-prune synthesis, Internet of Medical Things, SARS-CoV-2, smart healthcare, synthetic data generation, wearable online computing, wearable systems.**

I. Introduction

SARS-COV-2, also known as novel coronavirus, emerged in China and soon after spread across the globe. The World Health Organization (WHO) named the resultant disease COVID-19. COVID-19 was declared a pandemic on March 11, 2020 [1]. In its early stages, the symptoms of COVID-19 include fever, cough, fatigue, and myalgia. However, in more serious cases, it can lead to shortness of breath, pneumonia, severe acute respiratory disorder, and heart problems, and may lead to death [2]. It is of paramount importance to detect which individuals are infected at as early a stage as possible in order to limit the spread of disease through quarantine and contact tracing. In response to COVID-19, governments around the world issued social distancing and self-isolation orders. This led to a significant increase in unemployment across diverse economic sectors. As a result, COVID-19 triggered an economic recession in a large number of countries [3].

Reverse Transcription-Polymerase Chain Reaction (RT-PCR) is currently the gold standard for SARS-CoV-2 detection [4]. This test is based on viral nucleic acid detection in sputum or nasopharyngeal swab. Although it has high specificity, it has several drawbacks. The RT-PCR test is invasive and uncomfortable, and non-reusable testing kits have led to significant supply chain deficiencies. SARS-CoV-2 infection can also be assessed with an antibody test [5]. However, antibody titers are only detectable from the second week of illness

Manuscript received April 12, 2021; revised October 4, 2021; accepted November 19, 2021. Date of publication November 24, 2021; date of current version December 23, 2021. This work was supported by NSF under Grant CNS-1907381. *(Corresponding author: Shayan Hassantabar.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Institutional Research Board (IRB) approval (reference no.: P-20200044163) at San Matteo Hospital in Pavia, Italy.

Shayan Hassantabar and Niraj K. Jha are with the Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: seyedh@princeton.edu; jha@princeton.edu).

Novati Stefano and Alessandra Ferrari are with the Division of Infectious Diseases Unit, Fondazione IRCCS Policlinico San Matteo, 27100 Pavia, Italy (e-mail: s.novati@smatteo.pv.it; alessandra.ferrari@smatteo.pv.it).

Vishweshwar Ghanakota and Gregory N. Nicola are with NeuTigers Inc., Brooklyn, NY 11201 USA (e-mail: vishu@neutigers.com; greg@neutigers.com).

Raffaele Bruno is with the Division of Infectious Diseases Unit, Fondazione IRCCS Policlinico San Matteo, 27100 Pavia, Italy, and also with the Department of Medical, Surgical, Diagnostic and Pediatric Science, University of Pavia, 27100 Pavia, Italy (e-mail: raffaele.bruno@unipv.it).

Ignazio R. Marino is with the Department of Jefferson Health, Thomas Jefferson University, Philadelphia, PA 19107 USA (e-mail: ignazio.marino@jefferson.edu).

Kenza Hamidouche is with the Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: kenzah@princeton.edu).

Digital Object Identifier 10.1109/TCE.2021.3130228

onwards and persist for an uncertain length of time. The antibody test is also invasive, requiring venipuncture which, in combination with a several-day processing time, makes it less ideal for rapid mass screening. In the current economic and social situation, there is a great need for an alternative SARS-CoV-2/COVID-19 detection method that is easily accessible to the public for repeated testing with high accuracy.

To address the above issues, researchers have begun to explore the use of artificial intelligence (AI) algorithms to detect COVID-19 [6]. Initial work concentrated on CT scans and X-ray images [4], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21]. A survey of such datasets can be found in [22], [23]. These methods often rely on transfer learning of a convolutional neural network (CNN) architecture, pre-trained on large image datasets, on a smaller COVID-19 image dataset. However, such an image-based AI approach faces several challenges that include lack of large datasets and inapplicability outside the clinic or hospital. Another work [24] shows that it is difficult to distinguish COVID-19 pneumonia from influenza virus pneumonia in a clinical setting using CT scans. Thus, the work in this area is not mature yet.

CORD-19 [25] is an assembly of 59000 scholarly articles on COVID-19. It can be used with natural language processing methods to distill useful information on COVID-19-related topics.

AI4COVID-19 [26] performs a preliminary diagnosis of COVID-19 through cough sample recordings with a smartphone application. However, since coughing is a common symptom of two dozen non-COVID-19 medical conditions, this is an extremely difficult task. Nonetheless, AI4COVID-19 shows promising results and opens the door for COVID-19 diagnosis through a smartphone.

The emergence of wearable medical sensors (WMSs) offers a promising way to tackle these challenges. WMSs can continuously sense physiological signals throughout the day [27]. Hence, they enable constant monitoring of the user's health status. Training AI algorithms with data produced by WMSs can enable pervasive health condition tracking and disease onset detection [28]. This approach exploits the knowledge distillation capability of machine learning algorithms to directly extract information from physiological signals. Thus, it is not limited to disease detection in clinical scenarios.

We propose a framework called CovidDeep for daily detection of SARS-CoV-2/COVID-19 based on off-the-shelf WMSs and compact deep neural networks (DNNs). It bypasses manual feature engineering and directly distills information from the raw signals captured by available WMSs. It addresses the problem posed by small COVID-19 datasets by relying on intelligent synthetic data generation from the same probability distribution as the training data [29]. These synthetic data are used to pre-train the DNN architecture in order to impose a prior on the network weights. To cut down on the computation and storage costs of the model without any loss in accuracy, CovidDeep leverages the grow-and-prune DNN synthesis paradigm [30], [31]. This not only improves accuracy, but also shrinks model size and reduces the computation costs of the inference process.

A. Novel Contributions of This Study

Next, we summarize the major contributions of this article:

- We propose CovidDeep, an easy-to-use, accurate, and pervasive SARS-CoV-2/COVID-19 detection framework. It combines features extracted from physiological signals using WMSs and simple-to-answer questions in a smartphone application-based questionnaire with efficient DNNs.
- It uses an intelligent synthetic data generation module to obtain a synthetic dataset [29], labeled by decision rules. The synthetic dataset is used to pre-train the weights of the DNN architecture.
- It uses a grow-and-prune DNN synthesis paradigm that learns both an efficient architecture and weights of the DNN at the same time [30], [31].
- It provides a solution to the daily SARS-CoV-2/COVID-19 detection problem. It captures all the required physiological signals non-invasively through comfortably-worn WMSs that are commercially available.
- The CovidDeep DNNs are embedded in a smartphone application (app) to enable edge inference.

The rest of the article is organized as follows. Section II reviews background material. Section III describes the CovidDeep framework. Section IV provides implementation details. Section V presents experimental results. Section VI provides a short discussion on CovidDeep and possible directions for future research. Finally, Section VII concludes the article.

II. Background

In this section, we discuss background material related to the CovidDeep framework. This includes recent methods for synthesizing and training efficient DNN architectures, as well as related work on smart healthcare platforms.

Use of efficient building blocks leads to compact DNNs/CNNs and significantly reduces computational costs and storage needs. For example, inverted residual blocks used in MobileNetV2 [32] reduce the number of parameters and the floating-point operations (FLOPs) greatly. In addition, spatial convolution is one of the most computationally expensive operations in CNN architectures. To address this issue, ShuffleNet-v2 [33] uses depth-wise separable convolutions and channel-shuffling operations. Furthermore, Shift [34] addresses this problem by using shift-based modules that combine shifts and point-wise convolutions. Neural architecture search (NAS) is also used in the literature to automatically generate compact architectures. For example, FBNetV2 [35] uses a differentiable NAS approach to synthesize compact CNN architectures. Efficient performance predictors, e.g., for accuracy, latency, and energy, are also used to accelerate the DNN search process [36], [37]. FBNetV3 [38] also takes into account the training recipe (i.e., training hyperparameters) in NAS, leading to higher accuracy-recipe combinations.

DNN compression methods remove redundancy in DNN models. Network pruning [39] removes redundancy from both CNN and multilayer-perceptron architectures. ESE [40] uses pruning to also remove redundancy in recurrent neural networks. Dai *et al.* [30], [41] combine network growth

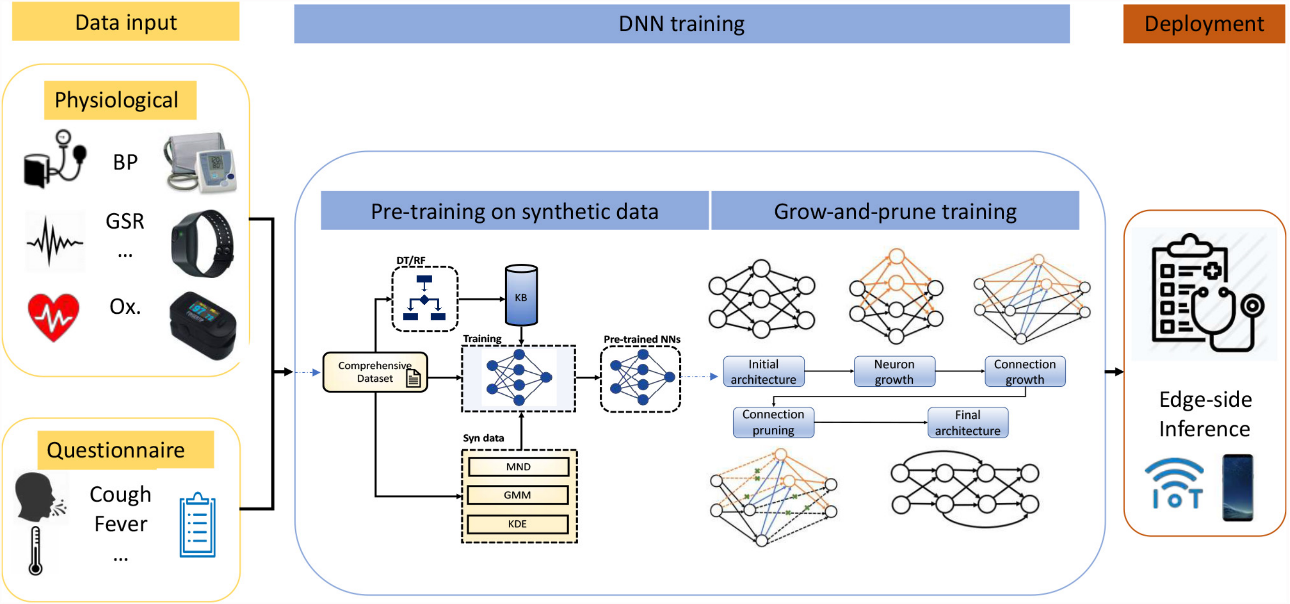


Fig. 1. Schematic diagram of the CovidDeep framework (GSR: Galvanic skin response, Ox.: oxygen saturation, BP: blood pressure, DT/RF: decision tree/random forest, NN: neural network, KB: knowledge-base, MND: multi-variate Normal distribution, GMM: Gaussian mixture model, KDE: kernel density estimation).

with pruning to generate efficient CNNs and long short-term memories. SCANN [31] combines feature dimensionality reduction with grow-and-prune synthesis to generate very compact models that can be easily deployed on edge devices and Internet-of-Things sensors.

Orthogonal to the above works, low-bit quantization of DNN weights can also be used to reduce computations in a network with little to no accuracy drop [42].

Smart healthcare is another emerging area that is related to CovidDeep. It has led to the development of a variety of advanced devices and systems for monitoring various biomedical signals and diagnosing several health conditions. A health decision support system is described in [27] to embed machine learning models at various tiers of healthcare. SoDA uses machine learning models to detect and alleviate the stress level of the user using WMSs [43]. SaYoPillow [44] also uses a smart wearable to monitor stress during sleep. Stress-Lysis [45] is a DNN-based approach for monitoring stress based on three sources of data, namely, body temperature, rate of motion, and sweat during physical activity. DiabDeep enables detection of Type I/II diabetes using WMSs and DNNs [28]. Another application of smart healthcare is in the design of edge devices to accurately detect seizure episodes based on electroencephalography (EEG) signals [46], [47], [48]. Monitoring user's cardiac activity [49], blood glucose level [50], and mental health [51] are among other use cases of smart healthcare platforms. Developing methods for faster data preparation and processing [52] and efficient inference on the edge [53] will be useful for further progress in this area.

III. Methodology

In this section, we present the CovidDeep framework. First, we give an overview of the entire framework. Then, we describe the DNN architecture that is used in CovidDeep for inference. We also describe how synthetic data generation can be used to impose a prior on the DNN weights and then use the DNN grow-and-prune synthesis paradigm to boost the test accuracy further and ensure the computational efficiency of the model.

A. Framework Overview

The CovidDeep framework is shown in Fig. 1. It obtains data from two different sources: physiological signals and questionnaire. It has two flows: one that does not use synthetic data and another one that does. When synthetic data are not used, the framework just uses the real dataset divided into three categories: training, validation, and test. It trains the DNNs with the training dataset and picks the best one for the given set of features based on the validation dataset, and finally tests this DNN on the test dataset to obtain the test accuracy. However, when the real training dataset size is small, it is often advantageous to draw a synthetic dataset from the same probability distribution. CovidDeep uses synthetic data generation methods to increase the dataset size and use such data to pre-train the DNN architecture. Then, it uses grow-and-prune synthesis to generate inference models that are both accurate and computationally-efficient. The models generated by CovidDeep are efficient enough to be deployed on the edge, e.g., the smartphone or smartwatch, for SARS-CoV-2/COVID-19 inference.

Next, we discuss the data input, model training, and model inference details.

- *Data input:* As mentioned above, physiological signals and a questionnaire are the two sources of data input to the model. The physiological signals are derived from WMSs embedded in a smartwatch as well as a discrete pulse oximeter and blood pressure monitor. These signals can be easily obtained in a non-invasive, passive, and user-transparent manner. The list of these signals includes Galvanic skin response (GSR), inter-beat interval (IBI) that indicates the heart rate, skin temperature, oxygen saturation, and blood pressure: systolic and diastolic. In the questionnaire, we ask the following yes/no questions: immune-compromised, chronic lung disease, cough, shortness of breath, chills, fever, muscle pain, headache, sore throat, smell-taste loss, and diarrhea. We collected data on age, gender, weight, height, and smoking/drinking (yes/no), but did not find them to be useful either because of overfitting or being unrepresentative. All the relevant data sources are aggregated into a comprehensive data input for further processing.

- *Model training:* CovidDeep uses different types of DNN models: (i) those trained on the raw data only, (ii) those trained on raw data augmented with synthetic data to boost accuracy, and (iii) those subjected to grow-and-prune synthesis for both boosting accuracy further and reducing model size. The first type of DNN model uses a few hidden layers. The second type of DNN model is trained based on a system called TUTOR [29] and is suitable for settings where data availability is limited. It provides the DNN with a suitable inductive bias. The third type of DNN model is based on the grow-and-prune DNN synthesis paradigm and employs three architecture-changing operations: neuron growth, connection growth, and connection pruning. These operations have been shown to yield DNNs that are both accurate and efficient [31].
- *Model inference:* CovidDeep enables the users to have SARS-CoV-2/COVID-19 detection decision on their edge device on demand.

Next, we discuss the CovidDeep DNN architecture.

B. Model Architecture

Fig. 2 shows the processing pipeline of the CovidDeep framework. The architecture takes the data inputs (shown at the bottom) and generates a prediction, i.e., the detection decision, (shown at the top). The pipeline consists of four steps: data pre-processing, synthetic data generation and architecture pre-training, grow-and-prune synthesis, and output generation through softmax.

In the data pre-processing stage, data normalization and data alignment/aggregation are done.

- *Data normalization:* This step is aimed at changing feature values to a common scale. While data normalization is not always required, it is highly beneficial in the case of datasets that have features with very different ranges. It leads to better noise tolerance and improvement in model accuracy [54]. Data normalization can be done in several ways, such as min-max scaling and standardization. In this work, we use min-max scaling to map each data input to the [0, 1] interval. Scaling can be done as follows:

$$x_{scaled} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

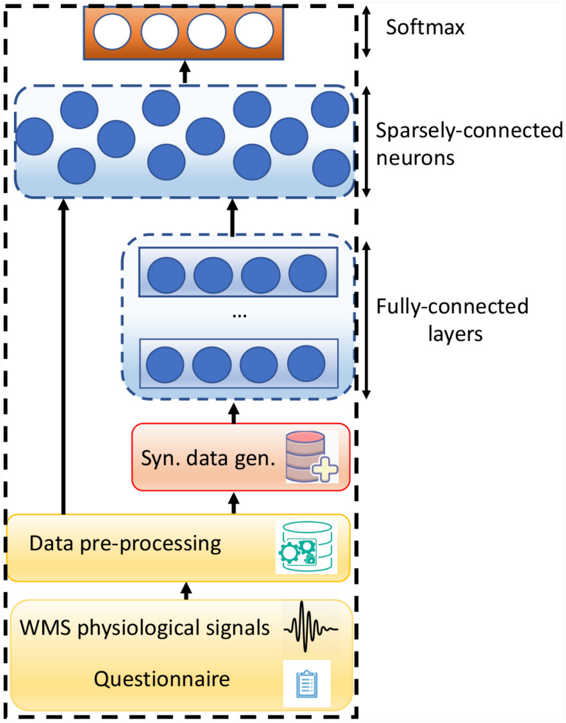


Fig. 2. An illustration of the CovidDeep processing pipeline to generate predictions from data inputs.

- *Data alignment/aggregation:* The data from different WMSs may have different start times and frequencies. In order to merge them into a dataset, we need to synchronize the data streams based on their timestamps. The answers to the questions in the questionnaire are also added to the final dataset.

Synthetic data generation: The training dataset generated in the above manner is next used to generate a synthetic dataset that is used to pre-train the DNN. These synthetic data and pre-training steps are based on the TUTOR framework [29]. The schematic diagram of the training scheme based on synthetic data is shown in Fig. 3. The synthetic dataset is generated in three different ways in TUTOR:

- Using multi-variate Normal distribution (MND): In this approach, the real training dataset, i.e., the one obtained as a fraction of the data obtained from the WMSs and questionnaire, is modeled as a normal distribution to generate the synthetic data.
- Using Gaussian mixture model (GMM): This approach uses a multi-dimensional GMM to model the data distribution. The optimal number of GMM components is obtained with the help of a validation dataset. Subsequently, the synthetic dataset is generated from this GMM.
- Using kernel density estimation (KDE): This approach uses non-parametric density estimation to estimate the probability distribution as a sum of many kernels. In our implementation, KDE is based on the Gaussian kernel function. The synthetic data are generated based on samples generated from this model.

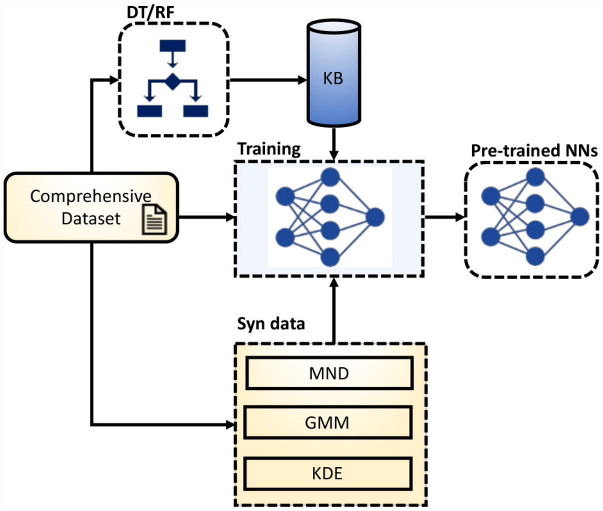


Fig. 3. The schematic diagram for pre-training of the DNN model with the synthetic dataset (DT/RF: decision tree/random forest, NN: neural network, KB: knowledge-base).

Building a knowledge base (KB): After generation of the synthetic data, we need to label the data points. To this end, we build a KB from the real training dataset. Decision tree (DT) and random forest (RF) are two classical machine learning methods that are inherently rule-based. In fact, each decision path in a decision tree, from the root to a leaf, can be thought of as a rule. Therefore, we aim to identify the set of rules that best describes the data. We use such a model as a KB to label the generated synthetic dataset.

Training with synthetic data: We use the labeled synthetic data to impose a prior on the DNN weights. To accomplish this, we pre-train the DNN model by using the generated synthetic dataset. This provides the network with an appropriate inductive bias and helps the network to “get underway." This helps improve accuracy when data availability is limited.

C. Grow-and-Prune DNN Synthesis

In this section, we discuss the grow-and-prune synthesis paradigm [30], [31]. The approach presented in [31] allows the depth of the DNN to grow during synthesis. Thus, a hidden neuron can receive inputs from any neuron activated before it (including input neurons) and can feed its output to any neuron activated after it (including output neurons). As a result, the depth of the model is determined based on how the hidden neurons are connected, enabling the depth to be changed during training. We use three basic architecture-changing operations in the grow-and-prune synthesis process that are discussed next.

Connection growth: This activates the dormant connections in the network. The weights of the added connections are set to 0 and trained later. We use two different methods for connection growth:

- *Gradient-based growth:* This approach was first introduced by Dai *et al.* [30]. Algorithm 1 shows the process of gradient-based growth. Each weight matrix has a corresponding binary mask of the same size. This mask is used to disregard the inactive connections. The algorithm adds connections to reduce the loss function $\mathcal{L}$ significantly. To this end, the gradients of all the dormant connections are evaluated and their effectiveness ranked based on this metric. During a training epoch, the gradients of all the weight matrices for all the data mini-batches are captured in the back-propagation step. An inactive connection is activated if its gradient magnitude is large relative to the gradients in its associated layer.
- *Full growth:* This connection growth restores all the dormant connections in the network to make the DNN fully-connected.

Algorithm 1 Connection Growth Algorithm

```
Input: $W \in R^{M \times N}$: weight matrix of dimension $M \times N$ (connecting layer with $M$ neurons to layer with $N$ neurons); $Mask \in R^{M \times N}$: weight mask of the same dimension as the weight matrix; Network $P$; $W.grad$: gradient of the weight matrix (of dimension $M \times N$); data $D$; $\alpha$: growth ratio
    if full growth then
        $Mask_{[1:M,1:N]} = 1$
    else if gradient-based growth then
        Forward propagation of data $D$ through network $P$ and then back-propagation
        Accumulation of $W.grad$ for one training epoch
        $t = (\alpha \times MN)^{th}$ largest element in the $|W.grad|$ matrix
        for all $w.grad_{ij}$ do
            if $\left|w.grad_{ij}\right| > t$ then
                $Mask_{ij} = 1$
            end if
        end for
    end if
    $W = W \otimes Mask$
Output: Modified weight matrix $W$ and mask matrix $Mask$
```

Algorithm 2 Connection Pruning Algorithm

```
Input: Weight matrix $W \in R^{M \times N}$; mask matrix $Mask$ of the same dimension as the weight matrix; $\alpha$: pruning ratio
    $t = (\alpha \times MN)^{th}$ largest element in $|W|$
    for all $w_{ij}$ do
        if $\left|w_{ij}\right| < t$ then
            $Mask_{ij} = 0$
        end if
    end for
    $W = W \otimes Mask$
Output: Modified weight matrix $W$ and mask matrix $Mask$
```

Connection pruning: Connection pruning deactivates the connections that are smaller than a specified threshold. Algorithm 2 shows this process.

Neuron growth: This step adds neurons to the network and thus increases network size. This is done by duplicating existing neurons in the architecture. To break the symmetry, random noise is added to the weights of all the connections related to the newly added neurons. The neurons to be duplicated are

Algorithm 3 Neuron Growth Algorithm

```
Input: Network $P$; weight matrix $W \in R^{M \times N}$; mask matrix
    $Mask$ of the same dimension as the weight matrix; data
    $D$; candidate neuron $n_j$ to be added; array $A$ of activation
    values for all hidden neurons
    if activation-based selection then
        forward propagation through $P$ using data $D$
        $i = argmax\ (A)$
    else if random selection then
        randomly pick an active neuron $n_i$
    end if
    $Mask_{j\cdot} = Mask_{i\cdot}, Mask_{\cdot j} = Mask_{\cdot i}$
    $w_{j\cdot} = w_{i\cdot} + noise, w_{\cdot j} = w_{\cdot i} + noise$
Output: Modified weight matrix $W$ and mask matrix $Mask$
```

either selected randomly or based on higher activation values. The process is explained in Algorithm 3.

We apply connection pruning after neuron growth and connection growth in each iteration. Grow-and-prune synthesis starts from a fully connected architecture (mask values set to 1) and runs for a pre-defined number of iterations. Finally, the architecture that performs the best on the validation dataset is chosen.

IV. Implementation Details

In this section, we first explain how the data were obtained from 87 individuals and how various datasets were prepared from the data. We also provide implementation details of the CovidDeep DNN model.

A. Data Collection and Preparation

We collected physiological signals and questionnaire data with Institutional Research Board (IRB) approval at San Matteo Hospital in Pavia, Italy. 30 individuals were healthy (referred to as Cohort 1) and the remaining were SARS-CoV-2-positive with varying levels of disease severity. The SARS-CoV-2-positive cases were categorized into two other cohorts: asymptomatic (Cohort 2 with 27 individuals) and symptomatic (Cohort 3 with 30 individuals). Distinguishing among these cohorts is important to ascertain who may be spreading the virus unknowingly and to determine whether medical support is needed for symptomatic individuals. Hence, we train DNN models that can perform three-way classification.

To collect the physiological signals, we used commercially available devices: Empatica E4 smartwatch (sensors we found useful: GSR, IBI, skin temperature), a pulse oximeter, and a blood pressure monitor. Alongside the physiological signals, we employed a questionnaire to collect information about possible COVID-19-related symptoms from all the individuals. We also collected data about age, gender, weight, height, and smoking/drinking (yes/no), but did not rely on these features as they were not necessarily representative of the larger population. Table I shows all the data types that we found to be useful. The smartwatch data capture the physiological state of the user. GSR measures continuous variations in the electrical characteristics of the skin, such as conductance, which can be caused by variations in body sweat. IBI correlates with cardiac health. Furthermore, skin acts as a medium for insulation, sweat, and control of blood flow. Although it is not a clear indicator of internal body temperature, skin temperature helps assess skin health. The pulse oximeter indirectly measures blood oxygen saturation. It is a comfortable and painless way of measuring how well oxygen is being sent to parts of the body furthest from the heart, such as the arms and legs. Blood pressure exposes various underlying health problems. Last, but not the least, the questionnaire elicits information that may help improve COVID-19 detection accuracy. From all these sources of data, we derive various subsets as datasets for use in the CovidDeep framework to see which data features are the most beneficial to obtaining a high detection accuracy. In addition, the various sensor subsets have different costs. Hence, our results also let one take test accuracy vs. cost into consideration.

TABLE I

Data Types Collected in the CovidDeep Framework

Data type	Data source
Immune-compromised	Questionnaire
Chronic lung disease	
Shortness of breath	
Cough	
Fever	
Muscle pain	
Chills	
Headache	
Sore throat	
Smell/taste loss	
Diarrhea	
Galvanic skin response (μS)	Smartwatch
Skin temperature ($^\circ C$)	
Inter-beat interval (ms)	
Oxygen saturation (%)	Pulse oximeter
Systolic blood pressure (mmHg)	Blood pressure monitor
Diastolic blood pressure (mmHg)	Blood pressure monitor

Before data collection commences, we inform the participants about the procedure. We then collect some relevant information and COVID-19-related symptoms in response to a questionnaire. We place the pulse oximeter on the index finger of the user for blood oxygen measurement. We also obtain the systolic/diastolic blood pressure measurements. We place the smartwatch on the participant's wrist. Data collection lasts for at most one hour for each participant, during which time we collect sensor data from the smartwatch. We stream the data from the smartwatch to the smartphone over Bluetooth in real-time using a smartphone application. This application collects the data and performs basic validation to ensure data integrity.

Next, we pre-process the raw data to generate a comprehensive dataset. To this end, we first synchronize the WMS data streams. We then divide the data streams into 15-second data windows. We then split the participants into three different sets: training, validation, and test. The training set contains data from 52 individuals, approximately 60% of all the participants. Among the 52 individuals represented in the training set, 18 are healthy, 16 are asymptomatic (but virus-positive), and 18 are symptomatic (and virus-positive). The validation set consists of data from 17 individuals, approximately 20% of all the participants, with 6, 5, and 6 individuals from Cohorts 1, 2, and 3, respectively. The test set contains data from 18 individuals, approximately 20% of all the participants, with 6 individuals from each of the three cohorts. This data partitioning ensures that all the data collected from any individual are limited to just one of the three sets. Furthermore, the data instances extracted from each individual have no time overlap. In addition, in order to conduct ablation studies to gauge the impact of different data streams, we create different datasets, with various subsets of all the features.

B. Model Implementation

We have implemented the CovidDeep framework in PyTorch. We perform DNN training on the Nvidia Tesla P100 data center accelerator, with 16GB of memory. We use cuDNN library to accelerate GPU processing. Next, we give the details of the implemented DNN architectures trained on the different datasets.

We train various DNNs (with different numbers of layers and different numbers of neurons per layer) and verify their performance on the validation dataset. In general, a four-layer architecture with 256, 128, 128, and 3 neurons, respectively, performs the best. The number of neurons in the input layer depends on which subset of features is selected for training the DNN. In the case of the full dataset, the input layer has 194 neurons, which indicates the dataset dimension. We obtain the features of the dataset from the 15-second data window as follows. Sensor data collected from the smartwatch in the data window consist of 180 signal readings, hence 180 features, from the three data streams running at 4Hz. We derive 11 features from the 11 questionnaire questions. Finally, we append the pulse oximeter oxygen saturation measurement and systolic/diastolic blood pressure measurements to obtain a feature vector of length 194.

We use leaky ReLU as the nonlinear activation function in all the DNN layers. As explained in Section III, we generate three DNNs for each dataset: (i) DNN trained on the real training dataset, (ii) DNN pre-trained on the synthetic dataset and then trained on the real training dataset, and (iii) DNN synthesized and trained with the grow-and-prune synthesis paradigm.

C. Network Training

We use the stochastic gradient descent optimizer for DNN training, with a learning rate of 5e-3 and batch size of 256. We use 100000 synthetic data instances to pre-train the network architecture. Moreover, in the grow-and-prune synthesis phase, we train the network for 20 epochs each time the architecture changes. We apply network-changing operations over five iterations. In this step, we use pruning to achieve a predefined number of connections in the network, chosen based on performance on the validation set.

V. Experimental Results

In this section, we analyze the performance of CovidDeep DNN models. We target three-way classification among the three cohorts described earlier. In addition, we perform an ablation study to analyze the impact of different subsets of features as well as different steps of CovidDeep DNN synthesis. We also describe the processing flow of the CovidDeep smartphone app that we have developed for edge-side diagnosis.

TABLE II
Confusion Matrix for the Most Accurate Three-Way Classification Model

Label↓\Prediction→	C1	C2	C3	Total
C1	1066	9	0	1075
C2	54	1152	0	1206
C3	0	0	975	975
Total	1120	1161	975	3256

The CovidDeep DNN models are first evaluated with four different metrics: test accuracy, false positive rate (FPR), false negative rate (FNR), and F1 score. These terms are based on the following terms:

- True positive (negative): SARS-CoV-2/COVID-19 (healthy) data instances classified as SARS-CoV-2/COVID-19 (healthy).
- False positive (negative): healthy (SARS-CoV-2/ COVID-19) data instances classified as SARS-CoV-2/ COVID-19 (healthy).

These metrics evaluate model performance from different perspectives. Test accuracy evaluates its overall prediction power. It is simply the ratio of all the correct predictions on the test data instances and the total number of such instances. The FPR is defined as the ratio of the number of negative, i.e., healthy, instances wrongly categorized as positive (false positives) and the total number of actual negative instances. The FNR is the ratio of positives that yield different test outcomes. Thus, there is an FNR for both Cohorts 2 and 3. Because of the three-way classification, the F1 score we report is the Macro F1 score.

A. Model Performance Evaluation

We obtained the highest test accuracy with a DNN model trained with the grow-and-prune synthesis paradigm on the dataset that contained features from four categories: GSR, pulse oximeter (Ox), blood pressure (BP), and questionnaire (Q). Table II shows the confusion matrix for three-way classification among the three cohorts: Cohort 1 (healthy), Cohort 2 (asymptomatic-positive), Cohort 3 (symptomatic-positive), denoted as C1, C2, and C3, respectively. CovidDeep DNN achieves a test accuracy of 98.1%. The model achieves an FPR of only 0.8%. The low FPR means that the model does not raise many false alarms. It results in a 4.5% FNR for Cohort 2 and a 0.0% FNR for Cohort 3, denoted as FNR(2) and FNR(3), respectively (each FNR refers to the ratio of the number of false predictions for that cohort divided by the total number of data instances of that type). The low FNRs demonstrate the ability of the DNN model to not miss virus-positive cases. Moreover, the Macro F1 score of the DNN model is also high: 98.2%.

Next, we compare the three DNN models, trained on the real training dataset, with the aid of synthetic data, and with the aid of grow-and-prune synthesis, for the most accurate case in Table III. From this comparison, we see that the use of synthetic data and then grow-and-prune synthesis is able to

TABLE III
TEST ACCURACY, FPR, FNRS, AND F1 SCORE (ALL IN %) FOR THE THREE DNN MODELS OBTAINED FOR THE MOST ACCURATE CASE

DNN model trained on	Acc.	FPR	FNR(2)	FNR(3)	F1 Score
Real training dataset	79.9	22.5	34.2	0.0	80.9
Real+synthetic training dataset	84.8	14.1	28.4	0.0	85.5
Real+synthetic training dataset + grow-prune	98.1	0.8	4.5	0.0	98.2

TABLE IV
TEST ACCURACY, FPR, FNRS, AND F1 SCORE (ALL IN %) FOR TWO DNN MODELS OBTAINED FOR FEATURE SUBSETS FROM ONE, TWO OR THREE DATA CATEGORIES

	DNN Model 1					DNN Model 2				
Data category	Acc.	FPR	FNR(2)	FNR(3)	F1 Score	Acc.	FPR	FNR(2)	FNR(3)	F1 Score
GSR	54.2	22.1	23.3	99.6	44.6	54.2	22.1	23.4	99.5	44.7
Temp	57.2	31.5	60.3	33.4	57.5	58.6	32.2	60.2	28.2	58.7
IBI	66.6	55.1	24.0	21.1	65.6	66.8	53.1	25.1	21.1	66.0
Ox	45.4	56.2	59.6	46.7	45.5	45.4	56.2	59.6	46.7	45.5
BP	44.3	96.3	60.3	5.2	36.4	44.3	96.3	60.3	5.2	36.4
Q	61.4	0.0	100.0	5.2	53.5	63.0	0.0	100.0	0.0	54.7
GSR+Temp	57.2	33.4	60.3	31.4	57.3	76.9	6.4	44.1	15.4	76.5
GSR+IBI	74.9	3.2	34.6	37.4	74.3	76.1	3.6	31.9	36.3	75.5
GSR+Ox	52.7	29.0	44.2	71.3	51.3	47.5	44.3	44.7	71.3	46.1
GSR+BP	55.2	70.7	53.8	5.2	52.7	64.1	46.4	51.2	5.2	63.7
GSR+Q	89.1	6.8	23.3	0.0	89.6	89.2	6.7	23.3	0.0	89.7
Temp+IBI	68.1	19.3	53.9	18.8	68.4	68.2	19.9	52.9	18.9	68.6
Temp+Ox	48.3	26.3	78.4	46.7	46.5	49.3	24.2	77.7	46.7	47.3
Temp+BP	50.3	84.5	54.7	5.2	45.9	53.7	74.0	54.7	5.2	50.9
Temp+Q	68.9	26.5	60.4	0.0	69.8	69.0	26.3	60.3	0.0	69.9
IBI+Ox	48.1	60.4	68.0	22.7	49.8	49.0	58.3	68.0	22.1	50.7
IBI+BP	47.8	92.8	54.0	5.2	44.8	48.5	89.8	54.9	5.2	46.3
IBI+Q	80.9	19.5	34.2	0.0	81.8	80.9	17.8	35.8	0.0	81.7
Ox+BP	59.6	56.2	54.8	5.2	59.1	66.9	56.2	35.0	5.2	66.8
Ox+Q	50.2	56.2	80.2	5.2	52.5	50.2	56.2	80.2	5.2	52.5
BP+Q	51.8	56.2	80.1	0.0	49.9	57.6	56.2	60.3	5.2	56.8
GSR+Temp+IBI	70.5	11.5	54.7	17.9	70.8	76.6	3.5	46.0	17.2	76.7
GSR+Temp+Ox	69.1	22.1	33.5	37.2	70.0	69.7	23.1	27.1	42.4	70.2
GSR+Temp+BP	57.0	64.0	54.8	5.2	55.4	67.0	34.2	54.4	5.2	66.4
GSR+Temp+Q	83.6	0.2	44.2	0.0	83.9	91.3	0.2	23.3	0.0	91.7
GSR+IBI+Ox	64.8	14.0	45.4	45.8	64.8	70.8	19.1	43.2	23.0	71.7
GSR+IBI+BP	60.2	34.4	52.8	29.5	61.5	64.3	32.2	43.7	29.5	64.8
GSR+IBI+Q	87.7	11.2	23.3	0.0	88.3	88.8	7.7	23.3	0.0	89.4
GSR+Ox+BP	71.3	40.7	37.1	5.2	71.2	81.9	23.1	4.1	29.8	82.1
GSR+Ox+Q	69.9	22.9	56.7	5.2	71.0	75.5	22.7	41.8	5.2	76.7
GSR+BP+Q	63.9	26.5	73.8	0.0	62.3	64.1	25.9	73.8	0.0	62.4
Temp+IBI+Ox	57.4	38.9	62.4	22.2	57.5	61.8	30.7	57.8	22.2	61.8
Temp+IBI+BP	55.8	71.6	51.2	5.2	53.9	55.3	70.0	54.0	5.2	53.0
Temp+IBI+Q	73.6	17.2	51.8	5.0	74.5	77.1	9.0	53.6	0.0	77.5
Temp+Ox+BP	70.6	34.5	44.2	5.4	72.1	72.3	33.9	40.4	5.2	73.7
Temp+Ox+Q	53.3	56.2	71.8	5.2	55.8	53.4	56.2	71.4	5.2	55.9
Temp+BP+Q	47.9	46.6	94.9	5.2	43.5	49.9	40.8	94.7	5.2	45.1
IBI+Ox+BP	65.0	59.1	37.5	5.2	66.1	64.1	60.8	38.4	5.2	65.0
IBI+Ox+Q	54.8	56.2	67.8	5.2	57.2	55.0	56.2	67.2	5.2	57.4
IBI+BP+Q	55.9	56.2	65.2	4.6	55.0	53.4	56.2	71.6	5.2	52.3
Ox+BP+Q	66.9	56.2	35.0	5.2	68.2	66.9	56.2	35.0	5.2	68.2

boost the test accuracy compared to the DNN model trained on just the real dataset. In addition, we see improvements in the FPR and FNR values. The F1 score also follows the same trend, increasing with the use of synthetic data, and even more with the use of grow-and-prune synthesis.

B. Ablation Studies

In this section, we report results on various ablation studies. We begin by considering DNN models trained on features obtained from subsets of the six data categories (five sensors and the questionnaire). This helps us understand the impact of each of these categories and their various combinations. Then, we analyze the impact of different parts of the CovidDeep training process, pre-training with synthetic data, and grow-and-prune synthesis.

Since there are six data categories from which the corresponding features are obtained, there are 64 subsets. However, one of these subsets is the null subset. Thus, we evaluate the remaining 63 subsets. For these evaluations, we only consider the first two types of DNN models, referred to as DNN Models 1 and 2. We consider grow-and-prune synthesis-based models later. The results shown in Table IV correspond to the case when features from only one, two or three data categories are chosen, and in Table V when features from four, five or six data categories are chosen. In these tables, Temp denotes skin temperature.

TABLE V
TEST ACCURACY, FPR, FNRS, AND F1 SCORE (ALL IN %) FOR TWO DNN MODELS OBTAINED FOR FEATURE SUBSETS FROM FOUR, FIVE OR SIX DATA CATEGORIES

	DNN Model 1					DNN Model 2				
Data category	Acc.	FPR	FNR(2)	FNR(3)	F1 Score	Acc.	FPR	FNR(2)	FNR(3)	F1 Score
GSR+Temp+IBI+Ox	76.6	23.3	27.0	19.2	77.3	74.5	28.5	28.3	18.8	75.2
GSR+Temp+IBI+BP	62.5	27.1	53.4	29.2	62.4	73.3	13.6	44.0	19.8	73.4
GSR+Temp+IBI+Q	87.1	0.2	34.7	0.0	87.5	89.1	1.6	27.9	0.0	89.6
GSR+Temp+Ox+BP	77.6	24.2	34.7	5.2	77.8	93.6	1.7	11.4	5.2	93.7
GSR+Temp+Ox+Q	80.7	22.5	27.8	5.2	81.7	81.2	22.5	26.4	5.2	82.2
GSR+Temp+BP+Q	60.0	11.5	93.4	5.2	53.2	61.8	11.5	93.0	0.0	54.5
GSR+IBI+Ox+BP	75.0	23.3	42.6	5.2	76.1	76.8	24.2	37.0	5.2	77.8
GSR+IBI+Ox+Q	69.8	32.2	48.5	5.2	71.4	76.1	40.4	24.5	4.9	77.1
GSR+IBI+BP+Q	59.3	32.6	80.3	0.8	57.1	66.2	3.4	84.5	4.6	60.7
GSR+Ox+BP+Q	79.9	22.5	34.2	0.0	80.9	84.8	14.1	28.4	0.0	85.5
Temp+IBI+Ox+BP	59.2	52.9	58.9	5.2	61.1	66.9	53.8	37.2	5.2	67.9
Temp+IBI+Ox+Q	63.1	48.5	52.2	5.2	65.1	62.1	56.2	48.0	5.2	64.0
Temp+IBI+BP+Q	54.5	31.9	90.3	5.2	49.8	54.7	30.7	90.7	5.1	49.8
Temp+Ox+BP+Q	67.1	56.2	34.5	5.2	68.3	66.8	56.2	35.3	5.2	68.1
IBI+Ox+BP+Q	66.9	56.2	35.0	5.2	68.2	66.9	56.2	35.0	5.2	68.2
GSR+Temp+IBI+Ox+BP	77.1	29.1	31.8	5.2	78.2	83.3	34.2	10.3	5.2	83.7
GSR+Temp+IBI+Ox+Q	67.2	5.8	79.1	5.2	65.3	83.1	20.1	23.5	5.2	83.9
GSR+Temp+IBI+BP+Q	64.3	4.7	88.2	5.1	57.8	69.0	15.7	65.8	4.7	67.0
GSR+Temp+Ox+BP+Q	83.8	0.4	39.1	5.2	84.2	83.8	0.4	39.1	5.2	84.2
GSR+IBI+Ox+BP+Q	71.8	37.5	38.5	5.2	73.3	75.3	23.8	41.1	5.2	76.6
Temp+IBI+Ox+BP+Q	62.5	44.8	57.0	5.2	64.5	66.6	48.8	42.4	5.2	68.3
GSR+Temp+IBI+Ox+BP+Q	77.8	18.3	39.4	5.2	78.8	83.7	26.9	15.9	5.2	84.1

TABLE VI
COMPARISON OF THE THREE DNN MODELS (ALL PERFORMANCE METRICS IN %) FOR VARIOUS FEATURE SETS

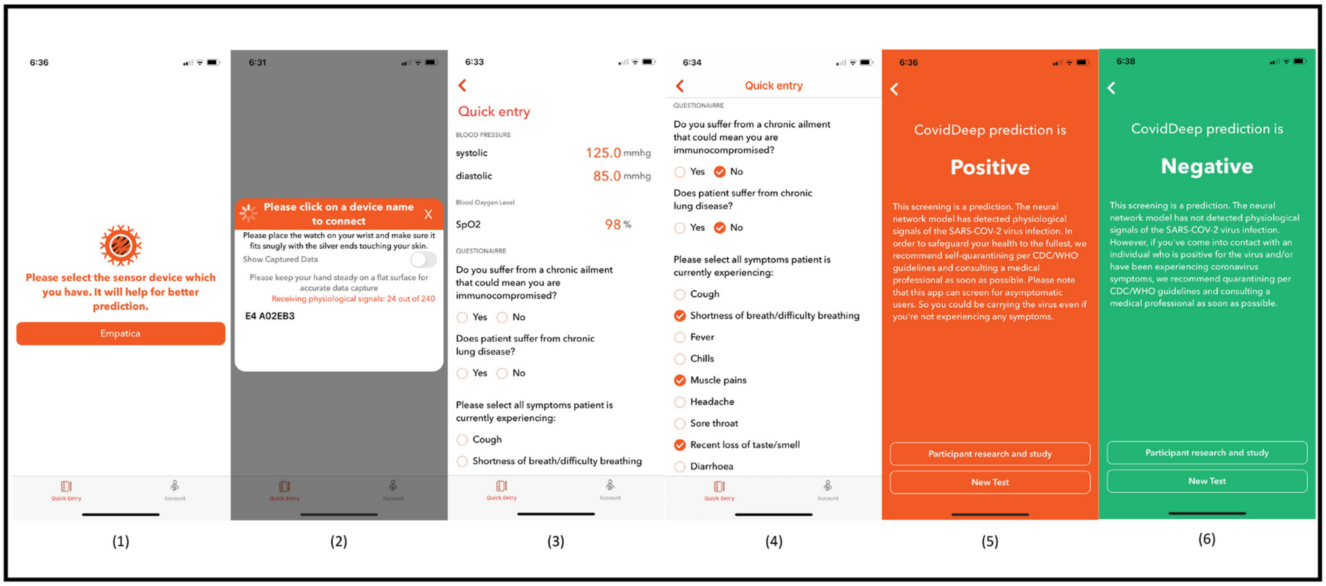
	DNN Models 1 and 2				DNN Model 3						
Data category	Acc.(1)	Acc.(2)	FLOPs	#Param.	Acc.	FLOPs	#Param	FPR	FNR(2)	FNR(3)	F1 Score
GSR+Ox+BP+Q	79.9	84.8	136.4k	68.5k	98.1	19.5k	10.0k	0.8	4.5	0.0	98.2
GSR+IBI+Q	87.7	88.8	165.6k	83.1k	91.5	39.5k	20.0k	1.3	21.9	0.0	91.9
GSR+Q	89.1	89.2	134.9k	67.7k	91.3	9.5k	5.0k	0.2	23.2	0.0	91.7
GSR+Temp+Q	83.6	91.3	165.6k	83.1k	91.3	151.5k	76.0k	0.2	23.3	0.0	91.7
GSR+Temp+IBI+Q	87.1	89.1	196.3k	98.4k	90.7	19.5k	10.0k	0.2	20.7	5.2	91.0
GSR+Temp+Ox+Q	80.7	81.2	166.1k	83.3k	87.7	119.5k	60.0k	0.3	28.7	5.2	88.1
GSR+Temp+IBI+Ox+Q	67.2	83.1	196.8k	98.7k	86.4	59.5k	30.0k	11.3	22.6	5.2	87.0
GSR+Temp+IBI+Ox+BP	77.1	83.3	192.2k	96.4k	84.6	59.5k	30.0k	29.5	11.2	5.2	85.1
GSR+Ox+BP	71.3	81.9	130.8k	65.7k	82.4	89.5k	45.0k	23.8	2.1	29.8	82.5
GSR+Temp+Ox+BP	77.6	93.6	161.5k	81.0k	82.3	129.5k	65.0k	25.2	21.0	5.2	82.8
IBI+Q	80.9	80.9	134.9k	67.7k	81.7	19.5k	10.0k	29.3	23.3	0.0	82.5

We first notice that DNN Model 2 generally performs better than DNN Model 1 across the various performance metrics. This underscores the importance of using synthetic data when the available dataset size is not large. Second, we observe that since this is a three-way classification, only 33.3% accuracy is possible by randomly predicting one of the three Cohorts. Thus, even single data categories (GSR, Temp, IBI, Ox, BP, Q) enable much better prediction than by chance. These single data categories are still only weak learners of the correct label, when used in isolation. Third, DNN models, in general, tend to perform better on the various performance metrics when more data categories are used. However, this is not always true. For example, we obtain the highest accuracy of 93.6% with DNN Model 2 when only features from four (GSR, Temp, Ox, BP) of the six categories are used. Adding features based on IBI or Q or both to these four categories actually reduces the test accuracy. This may be due to the curse of dimensionality. When the number of features increases, in general, the dataset size needs to be increased to obtain a good accuracy. For a fixed dataset size, this curse indicates that the number of features should be reduced. However, throwing out informative features would also reduce accuracy. In addition, some features are interactive, i.e., work synergistically to increase accuracy. Hence, a balance has to be found between accuracy and the number of features. Finally, when not all sensors are available (perhaps due to cost reasons), a suitable set that still provides reasonable accuracy can be chosen based on the given cost budget. This may help a broader cross-section of the population access the technology.

To illustrate the effect of the different parts of the CovidDeep training process, we compare 11 CovidDeep DNN models, trained based on the different DNN synthesis and training steps. We chose these models from different accuracy ranges. Table VI shows comparison results for the three-way classification task. We have already compared various performance metrics for DNN Models 1 and 2 earlier. Hence, here, we just report their accuracy, FLOPs, and number of model parameters (#Param). The best DNN Model 3 was obtained with the help of the validation dataset. This enabled us to find the best #Param. value. Only this model was tested

TABLE VII
TEST ACCURACY (%) COMPARISONS WITH TRADITIONAL MACHINE LEARNING MODELS FOR VARIOUS DATA CATEGORIES

Data category	AdaBoost	Naive Bayes	Random Forest	Decision Tree	SVM	*k*-NN	Our DNN
GSR+Ox+BP+Q	59.2	59.6	85.5	57.4	82.6	79.4	98.1
GSR+IBI+Q	67.2	61.4	76.4	77.0	80.8	81.6	91.5
GSR+Q	73.9	59.7	77.9	75.4	78.2	74.1	91.3
GSR+Temp+Q	63.8	63.4	70.4	66.2	79.0	73.8	91.3
GSR+Temp+IBI+Q	63.7	65.6	67.0	72.6	81.8	74.6	90.7
GSR+Temp+Ox+Q	78.9	63.4	71.3	58.1	78.1	74.0	87.7
GSR+Temp+IBI+Ox+Q	74.7	65.6	68.6	66.9	84.4	75.2	86.4
GSR+Temp+IBI+Ox+BP	68.9	44.3	63.9	53.6	67.8	68.8	84.6
GSR+Ox+BP	66.6	29.0	69.2	50.6	72.4	65.0	82.4
GSR+Temp+Ox+BP	66.7	34.8	55.2	51.5	70.9	71.8	82.3
IBI+Q	66.4	68.0	78.4	76.1	78.2	74.1	81.7



Please select the sensor device which you have. It will help for better prediction.

Please click on a device name to connect

Please place the watch on your wrist and make sure it fits snugly with the silver ends touching your skin.

Show Captured Data

Please keep your hand steady on a flat surface for accurate data capture

Receiving physiological signals: 24 out of 240

E4 A02EB3

BLOOD PRESSURE

systolic 125.0 mmhg

diastolic 85.0 mmhg

Blood Oxygen Level

SpO2 98 %

QUESTIONAIRRE

Do you suffer from a chronic ailment that could mean you are immunocompromised?

○ Yes ○ No

Does patient suffer from chronic lung disease?

○ Yes ○ No

Please select all symptoms patient is currently experiencing:

○ Cough

○ Shortness of breath/difficulty breathing

QUESTIONAIRRE

Do you suffer from a chronic ailment that could mean you are immunocompromised?

○ Yes ✓ No

Does patient suffer from chronic lung disease?

○ Yes ✓ No

Please select all symptoms patient is currently experiencing:

○ Cough

✓ Shortness of breath/difficulty breathing

○ Fever

○ Chills

✓ Muscle pains

○ Headache

○ Sore throat

✓ Recent loss of taste/smell

○ Diarrhoea

CovidDeep prediction is

Positive

This screening is a prediction. The neural network model has detected physiological signals of the SARS-COV-2 virus infection. In order to safeguard your health to the fullest, we recommend self-quarantining per CDC/WHO guidelines and consulting a medical professional as soon as possible. Please note that this app can screen for asymptomatic users. So you could be carrying the virus even if you're not experiencing any symptoms.

CovidDeep prediction is

Negative

This screening is a prediction. The neural network model has not detected physiological signals of the SARS-COV-2 virus infection. However, if you've come into contact with an individual who is positive for the virus and/or have been experiencing coronavirus symptoms, we recommend quarantining per CDC/WHO guidelines and consulting a medical professional as soon as possible.

Fig. 4. Screenshots of the CovidDeep app user experience.

on the test dataset. Acc.(1) and Acc.(2), respectively, refer to the accuracy of DNN Models 1 and 2. The FLOPs and #Param. for these two models are identical. We report all the performance metrics for DNN Model 3 that is generated by grow-and-prune synthesis using both real and synthetic data. Thus, the starting point for DNN Model 3 synthesis is DNN Model 2. Next, we compare DNN Model 3 with the other two models based on various measures and show why it is suitable for deployment on the edge devices.

- *Smaller model size:* It contains 3.4× fewer parameters on an average (geometric mean) than DNN Models 1 and 2, thus significantly reducing the memory requirements.
- *Less computation:* It reduces FLOPs per inference by 3.5× on an average (geometric mean) relative to DNN Models 1 and 2, thus facilitating more efficient inference on the edge devices.
- *Better performance:* It improves accuracy on an average by 7.8% (1.9%) relative to DNN Model 1 (2), while also lowering FPR and FNRs, in general.

Next, we compare the performance of CovidDeep DNNs with that of traditional machine learning models on the same 11 data categories evaluated in Table VI. Table VII compares the test accuracies. As we can see, CovidDeep DNNs outperform the other models for all 11 data categories. This highlights the difficulty traditional models face for classification based on raw data.

C. CovidDeep Smartphone Application

The CovidDeep DNN models can be deployed on a smartphone for edge inference by embedding them inside a smartphone app. We have developed the CovidDeep app for both iOS and Android devices. It uses Bluetooth to send sensor data to the smartphone. It includes fours steps: (i) capturing of sensor data from the Empatica E4 smartwatch, (ii) entering of data from the questionnaire and discrete sensors, (iii) data normalization, and (iv) DNN processing for diagnosis.

Fig. 4 shows several screenshots that depict user experience with the CovidDeep app. First, to capture physiological signals, the user is instructed to put on the Empatica E4 smartwatch on the wrist and turn it on. The smartwatch has to be paired with the mobile device using Bluetooth discovery. Once the smartwatch is paired, it can be controlled using the Empatica mobile software development kit. In this app, physiological signals are collected for one minute, resulting in four 15-second window data instances. Since not all sensors start reporting data at the same time, the elapsed time is typically slightly longer than a minute. Then, the user enters the questionnaire information and blood pressure/oxygen saturation values in the app. On iOS devices, if the user enables HealthKit integration and blood pressure and oxygen saturation values exist, they are auto-populated in the app. The normalization module normalizes the data using the minimum and maximum values employed in the training process. The inference module uses PyTorch application program interfaces to send the four 15-second data instances and obtains average prediction probabilities after DNN processing. Finally, a threshold is used with the average probabilities to predict the virus-positive or virus-negative status of the user.

VI. Discussion and Future Work

In this section, we discuss several points related to the CovidDeep framework. First, we discuss two inspirations we took from the human brain in the synthesis process of the CovidDeep DNNs. Next, we discuss future directions in medical research enabled by this framework.

CovidDeep took inspiration from the human brain in utilizing synthetic data in the DNN synthesis process. An interesting ability of the human brain is to efficiently solve novel problems in a new domain despite limited prior experience. Inspired by this human capability, CovidDeep uses the TUTOR [29] approach for synthetic data generation and labeling to help the neural network start from a better initialization point. Hence, it reduces the need for large datasets that are not readily available for SARS-CoV-2/COVID-19 AI research.

The CovidDeep DNN training process takes another inspiration from the human brain development process in the grow-and-prune DNN synthesis step. The human brain undergoes dynamic changes in its synaptic connections every second of its lifetime. Acquisition of knowledge depends on these synaptic rewirings [55]. Inspired by this phenomenon, CovidDeep utilizes the grow-and-prune synthesis paradigm to enable DNN architecture adaptation throughout training. CovidDeep DNNs obtained through grow-and-prune synthesis do not suffer from the situation faced by most current DNNs: fixed connections during training. This enables CovidDeep to generate very compact, yet accurate, models for SARS-CoV-2/COVID-19 detection.

CovidDeep uses physiological signals obtained using commercially available sensors to achieve a high test accuracy. As a result, it provides a testing mechanism that is accurate, easily accessible to the general public, and easy for individuals to use. It demonstrates that WMS-based SARS-CoV-2/COVID-19 detection is feasible. We continue to collect more data across various countries for further validation of the CovidDeep models. The CovidDeep framework, alongside previous studies on diabetes diagnosis with the help of such sensors [28], gives us confidence that in the future WMS-based disease detection is feasible for a large number of diseases [27].

VII. Conclusion

In this article, we proposed a framework called CovidDeep to facilitate daily and pervasive detection of SARS-CoV-2/COVID-19. The framework combines off-the-shelf WMSs with efficient DNNs to achieve this goal. It uses synthetic data generation to alleviate the need for large datasets. Training of CovidDeep DNNs based on the grow-and-prune synthesis paradigm enables them to learn both the weights and the architecture during training. Hence, these DNNs can be easily deployed on edge devices (e.g., smartphones or smartwatches) as well as servers. CovidDeep was evaluated based on data collected from 87 individuals. The highest accuracy it achieves is 98.1%. We also obtained high enough test accuracies for many different sets of sensor/questionnaire data categories. Thus, users can choose the DNN model that is based on the sensors that are most conveniently accessible to them from the market. With more data collected from larger deployment scenarios, the accuracy of CovidDeep DNNs can be improved further through incremental learning.

Contributions

The SARS-CoV-2/COVID-19 detection project was conceived by Niraj K. Jha. He also supervised the dataset preparation and DNN model generation efforts. Shayan Hassantabar performed DNN synthesis and evaluation. Vishweshwar Ghanakota developed the smartphone application for data collection, authenticated the credentials of the application sending data, ensured data integrity, and ran pre-processing scripts. Gregory N. Nicola MD and Ignazio R. Marino MD defined the patient cohorts, and helped with the IRB approval process. Gregory N. Nicola MD, Ignazio R. Marino MD, and Bruno Raffaele decided which questions were to be placed in the questionnaire. Novati Stefano, Alessandra Ferrari, and Bruno Raffaele collected data from patients and healthy individuals and labeled the data. Kenza Hamidouche helped with the synthesis and evaluation of the DNN models. All co-authors helped with the revision and editing of the manuscript.

Acknowledgment

The project was facilitated by the tireless efforts of Bob Schena (CEO, Rajant Corp.) and Adel Laoui (CEO, NeuTigers, Inc.). Giana Schena and Maria Schena helped with buying and transporting the instruments as well as English-to-Italian translations of various documents. Joe Zhang helped initially with feature extraction from the raw dataset. Claudia Cirillo coordinated the administrative work and helped with translation of documents to Italian for the IRB application. Ravi Jha helped with proofreading of the manuscript. The Chief of Italian Police, Franco Gabrielli, helped ensure safe and fast entrance and transfer of U.S. researchers on Italian soil during the COVID-19 lockdown.

Competing Interests

Five of the co-authors of this article, Niraj K. Jha, Shayan Hassantabar, Vishweshwar Ghanakota, Gregory N. Nicola MD, and Kenza Hamidouche have equity in NeuTigers, Inc. NeuTigers, along with Rajant Corporation and Thomas Jefferson University and Jefferson Health, enabled data collection from San Matteo Hospital, Pavia, Italy.

References

[1] "Coronavirus disease 2019 (COVID-19): Situation report–72." World Health Organization and Others. 2020. [Online]. Available: https://apps.who.int/iris/bitstream/handle/10665/331685/nCoVsitrep01Apr2020-eng.pdf

[2] E. Mahase, "Coronavirus: COVID-19 has killed more people than SARS and MERS combined, despite lower case fatality rate," *Brit. Med. J.*, vol. 368, p. m641, Feb. 2020. [Online]. Available: https://www.bmj.com/content/368/bmj.m641

[3] M. Nicola *et al.*, "The socio-economic implications of the coronavirus pandemic (COVID-19): A review," *Int. J. Surg.*, vol. 78, pp. 185–193, Jun. 2020.

[4] C. Butt, J. Gill, D. Chun, and B. A. Babu, "Deep learning system to screen coronavirus disease 2019 pneumonia," *Appl. Intell.*, p. 1, Apr. 2020. [Online]. Available: https://link.springer.com/article/10.1007/s10489-020-01714-3

[5] K. Dheda *et al.*, "Diagnosis of COVID-19: Considerations, controversies and challenges in South Africa," *Wits J. Clin. Med.*, vol. 2, pp. 3–10, Apr. 2020.

[6] J. Bullock, A. Luccioni, K. H. Pham, C. S. N. Lam, and M. Luengo-Oroz, "Mapping the landscape of artificial intelligence applications against COVID-19," 2020, *arXiv:2003.11336*.

[7] M. Farooq and A. Hafeez, "COVID-ResNet: A deep learning framework for screening of COVID19 from radiographs," 2020, *arXiv:2003.14395*.

[8] L. Wang and A. Wong, "COVID-Net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images," 2020, *arXiv:2003.09871*.

[9] "SIRM COVID-19 Database." [Online]. Available: https://www:sirm:org/category/senza-categoria/covid-19/ (Accessed: Jul. 2020).

[10] "Diagnosi Radiologica e Prevenzione Della Diffusione di COVID-19 Nei Dipartimenti di Radiologia." [Online]. Available: https://www.sirm.org/wp-content/uploads/2020/03/SIRM-Covid-19.pdf (Accessed: Jul. 2020).

[11] J. Zhang *et al.*, "Viral pneumonia screening on chest X-ray images using confidence-aware anomaly detection," 2020, *arXiv:2003.12338*.

[12] A. Narin, C. Kaya, and Z. Pamuk, "Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks," 2020, *arXiv:2003.10849*.

[13] A. Abbas, M. M. Abdelsamea, and M. M. Gaber, "Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network," 2020, *arXiv:2003.13815*.

[14] L. O. Hall, R. Paul, D. B. Goldgof, and G. M. Goldgof, "Finding COVID-19 from chest X-rays using deep learning on a small dataset," 2020, *arXiv:2004.02060*.

[15] P. K. Sethy and S. K. Behera, "Detection of coronavirus disease (COVID-19) based on deep features," *Preprints*, 2020, doi: 10.20944/preprints202003.0300.v1.

[16] L. Li *et al.*, "Using artificial intelligence to detect COVID-19 and community-acquired pneumonia based on pulmonary CT: Evaluation of the diagnostic accuracy," *Radiology*, vol. 296, no. 2, pp. E65–E71, 2020.

[17] O. Gozes *et al.*, "Rapid AI development cycle for the coronavirus (COVID-19) pandemic: Initial results for automated detection & patient monitoring using deep learning CT image analysis," 2020, *arXiv:2003.05037*.

[18] I. D. Apostolopoulos and T. A. Mpesiana, "COVID-19: Automatic detection from X-ray images utilizing transfer learning with convolutional neural networks," *Phys. Eng. Sci. Med.*, vol. 43, pp. 635–640, Apr. 2020.

[19] S. Wang *et al.*, "A fully automatic deep learning system for COVID-19 diagnostic and prognostic analysis," *Eur. Respiratory J.*, vol. 56, no. 2, 2020, Art. no. 2000775.

[20] P. Afshar, S. Heidarian, F. Naderkhani, A. Oikonomou, K. N. Plataniotis, and A. Mohammadi, "COVID-CAPS: A capsule network-based framework for identification of COVID-19 cases from X-ray images," 2020, *arXiv:2004.02696*.

[21] S. Hassantabar, M. Ahmadi, and A. Sharifi, "Diagnosis and detection of infected tissue of COVID-19 patients based on lung X-ray image using convolutional neural network approaches," *Chaos Solitons Fractals*, vol. 140, Nov. 2020, Art. no. 110170.

[22] R. Kalkreuth and P. Kaufmann, "COVID-19: A survey on public medical imaging data resources," 2020, *arXiv:2004.04569*.

[23] J. P. Cohen, P. Morrison, and L. Dao, "COVID-19 image data collection," 2020, *arXiv:2003.11597*.

[24] L. Lin, G. Fu, S. Chen, and J. Tao, "CT manifestations of coronavirus disease (COVID-19) pneumonia and influenza virus pneumonia: A comparative study," *Amer. J. Roentgenol.*, vol. 216, no. 1, pp. 1–9, 2021.

[25] "COVID-19 open research dataset (CORD-19)," Semantic Scholar, 2020. [Online]. Available: https://pages:semanticscholar:org/Coronavirus-research.

[26] A. Imran *et al.*, "AI4COVID-19: AI enabled preliminary diagnosis for COVID-19 from cough samples via an app," 2020, *arXiv:2004.01275*.

[27] H. Yin and N. K. Jha, "A health decision support system for disease diagnosis based on wearable medical sensors and machine learning ensembles," *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 3, no. 4, pp. 228–241, Oct./Dec. 2017.

[28] H. Yin, B. Mukadam, X. Dai, and N. K. Jha, "DiabDeep: Pervasive diabetes diagnosis based on wearable medical sensors and efficient neural networks," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 3, pp. 1139–1150, Jul.–Sep. 2021.

[29] S. Hassantabar, P. Terway, and N. K. Jha, "TUTOR: Training neural networks using decision rules as model priors," 2020, *arXiv:2010.05429*.

[30] X. Dai, H. Yin, and N. K. Jha, "NeST: A neural network synthesis tool based on a grow-and-prune paradigm," *IEEE Trans. Comput.*, vol. 68, no. 10, pp. 1487–1497, Oct. 2019.

[31] S. Hassantabar, Z. Wang, and N. K. Jha, "SCANN: Synthesis of compact and accurate neural networks," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, early access, Sep. 29, 2021, doi: 10.1109/TCAD.2021.3116470.

[32] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 4510–4520.

[33] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," 2018, *arXiv:1807.11164*.

[34] B. Wu *et al.*, "Shift: A zero FLOP, zero parameter alternative to spatial convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 9127–9135.

[35] A. Wan *et al.*, "FBNetV2: Differentiable neural architecture search for spatial and channel dimensions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, USA, 2020, pp. 12965–12974.

[36] X. Dai *et al.*, "ChamNet: Towards efficient network design through platform-aware model adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, 2019, pp. 11390–11399.

[37] S. Hassantabar, X. Dai, and N. K. Jha, "STEERAGE: Synthesis of neural networks using architecture search and grow-and-prune methods," 2019, *arXiv:1912.05831*.

[38] X. Dai *et al.*, "FBNetV3: Joint architecture-recipe search using predictor pretraining," 2020, *arXiv:2006.02049*.

[39] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," 2015, *arXiv:1510.00149*.

[40] S. Han *et al.*, "ESE: Efficient speech recognition engine with sparse LSTM on FPGA," in *Proc. ACM/SIGDA Int. Symp. Field-Program. Gate Arrays*, 2017, pp. 75–84.

[41] X. Dai, H. Yin, and N. K. Jha, "Grow and prune compact, fast, and accurate LSTMs," *IEEE Trans. Comput.*, vol. 69, no. 3, pp. 441–452, Mar. 2020.

[42] C. Zhu, S. Han, H. Mao, and W. J. Dally, "Trained ternary quantization," 2016, *arXiv:1612.01064*.

[43] A. O. Akmandor and N. K. Jha, "Keep the stress away with SoDA: Stress detection and alleviation system," *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 3, no. 4, pp. 269–282, Oct./Dec. 2017.

[44] L. Rachakonda, A. K. Bapatla, S. P. Mohanty, and E. Kougianos, "SaYoPillow: Blockchain-integrated privacy-assured IoMT framework for stress management considering sleeping habits," *IEEE Trans. Consum. Electron.*, vol. 67, no. 1, pp. 20–29, Feb. 2021.

[45] L. Rachakonda, S. P. Mohanty, E. Kougianos, and P. Sundaravadivel, "Stress-lysis: A DNN-integrated edge device for stress level detection in the IoMT," *IEEE Trans. Consum. Electron.*, vol. 65, no. 4, pp. 474–483, Nov. 2019.

[46] M. A. Sayeed, S. P. Mohanty, E. Kougianos, and H. P. Zaveri, "eSeiz: An edge-device for accurate seizure detection for smart healthcare," *IEEE Trans. Consum. Electron.*, vol. 65, no. 3, pp. 379–387, Aug. 2019.

[47] M. A. Sayeed, S. P. Mohanty, E. Kougianos, and H. P. Zaveri, "Neuro-detect: A machine learning-based fast and accurate seizure detection system in the IoMT," *IEEE Trans. Consum. Electron.*, vol. 65, no. 3, pp. 359–368, Aug. 2019.

[48] I. L. Olokodana, S. P. Mohanty, E. Kougianos, and R. S. Sherratt, "EZcap: A novel wearable for real-time automated seizure detection from EEG signals," *IEEE Trans. Consum. Electron.*, vol. 67, no. 2, pp. 166–175, May 2021.

[49] S. Raj, "An efficient IoT-based platform for remote real-time cardiac activity monitoring," *IEEE Trans. Consum. Electron.*, vol. 66, no. 2, pp. 106–114, May 2020.

[50] A. M. Joshi, P. Jain, S. P. Mohanty, and N. Agrawal, "iGLU 2.0: A new wearable for accurate non-invasive continuous serum glucose measurement in IoMT framework," *IEEE Trans. Consum. Electron.*, vol. 66, no. 4, pp. 327–335, Nov. 2020.

[51] S. Hassantabar, J. Zhang, H. Yin, and N. K. Jha, "MHDeep: Mental health disorder detection system based on body-area and deep neural networks," 2021, *arXiv:2102.10435*.

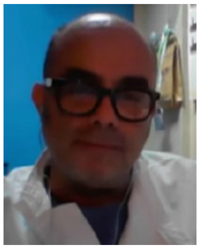
[52] S. Enayati, Z. Yang, B. Lu, and S. Vucetic, "A visualization approach for rapid labeling of clinical notes for smoking status extraction," in *Proc. Data Sci. Human Loop Lang. Adv.*, 2021, pp. 24–30.

[53] H.-G. Kim and G. Y. Kim, "Deep neural network-based indoor emergency awareness using contextual information from sound, human activity, and indoor position on mobile device," *IEEE Trans. Consum. Electron.*, vol. 66, no. 4, pp. 271–278, Nov. 2020.

[54] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[55] S. Grossberg, "Nonlinear neural networks: Principles, mechanisms, and architectures," *Neural Netw.*, vol. 1, no. 1, pp. 17–61, 1988.

Shayan Hassantabar received the B.S. degree in electrical engineering, digital systems focus, from the Sharif University of Technology, Iran, the M.Math. degree in computer science from the University of Waterloo, Canada, and the M.A. degree in electrical and computer engineering from Princeton University, where he is currently pursuing the Ph.D. degree. His research interests include automated neural network architecture synthesis, neural network compression, and smart healthcare.



Novati Stefano is the Medical Director with the University of Pavia with a school specialty in Infectious Diseases. In 1997, he benefited from IRCCS research fellowships with the Institute of Clinic of Infectious Diseases of Pavia. He then joined the Parasitology Laboratory and since 1999, has been serving with the outpatient departments of the Infectious Diseases Clinic of the same polyclinic, mainly dealing with the care of patients suffering from HIV and HCV infections. He is currently managing the Infectious Diseases Clinic that caters to patients with highly complex diagnostic/therapeutic pathways: HIV seropositive subjects and HSV positive co-infected.

Vishweshwar Ghanakota received the bachelor's degree from NIT, India, and the M.B.A. degree from INSEAD. He has over 18 years of experience designing and implementing large-scale real-time complex event processing systems. His last startup, Introspex, was acquired by Cisco-AppDynamics. Prior to that, he worked with healthcare startups to pull patient records from EMR systems using the FHIR protocol and make them portable using a mobile device. He is an Active Investor of Virtual Reality, Biotech, and Medical Device Startups. In his current role as a CTO with NeuTigers, he is working to commercialize latest Edge AI Technologies from Princeton University to transform patient care.

Alessandra Ferrari received the graduated degree in pharmacy from the University of Pavia, with a school specialty in Medical Pharmacology at the University of Milan. Subsequently, she attended numerous postgraduate courses on clinical trials as a "Biomedical Research Specialist," obtaining the qualification to carry out activities as a Clinical Research Associate and "Pharmacovigilance in the Course of Clinical Trials." She currently coordinates a Clinical Research Group, Fondazione IRCCS Policlinico San Matteo, managing clinical trials in various therapeutic areas.

Gregory N. Nicola serves as the Chief Medical Officer with NeuTigers Inc. He is a trained Radiologist subspecialized in neuroradiology. He currently serves as an Executive Leader of the Hackensack Radiology Group, Hackensack, NJ, USA, as well as a Board Member and the Finance Chair with the Hackensack Meridian Health Partners Clinically Integrated Network. He serves on his health systems care transformation team with particular interest in cost. He currently serves as the Chair of the ACR Commission on Economics. He is the past ASNR Advisor to the AMA RUC and a past two-time participant of the CMS/Acumen Resource Use Measure Committee. He has authored numerous peer-reviewed publications and is a frequent speaker on healthcare economics.

Raffaele Bruno received the graduated degree in medicine and surgery from the University of Pavia in 1991. In 2009, he was nominated as a Secretary of the AISF Italian Association for the Study of the Liver. He currently holds the position of a Full Professor of Infectious Diseases with the University of Pavia in agreement with the IRCCS Policlinico San Matteo-Pavia Foundation. He has authored over 250 publications in scientific journals (H-index SCOPUS 49). He is currently the Director of the Unit of Infectious Diseases, IRCCS Policlinico San Matteo di Pavia Foundation. In 1995, he was the winner of the competition for the University Researcher in the scientific disciplinary sector MED 17 Infectious Diseases.

Ignazio R. Marino is a Transplant Surgeon and the University Professor, and has published extensively as a clinician and an essayist. He trained with Thomas E. Starzl, who performed the first liver transplant in humans. In the United States, he has held chairs as a Professor of Surgery with the University of Pittsburgh and currently with the Thomas Jefferson University, Philadelphia. His main civil rights interests are those in favor of end-of-life decisions and same-sex marriages. He was also a Former Italian Senator and Mayor of Rome. He is currently an Executive Vice-President with Thomas Jefferson University and Jefferson Health, where he is leading the charge on healthcare delivery transformation. From 1992 to 1993, he was a member of the team that carried out the first baboon-to-human liver transplants in medical history. From 2006 to 2013, he has held a seat in the Italian Senate, where he chaired the Health Committee and the Investigative Committee on the National Health System.

Kenza Hamidouche received the bachelor's degree in computer science from the Ecole Nationale Supérieure d'Informatique, Algeria, the master's degree from the University of Rennes 1, France, and the Ph.D. degree in telecommunication from CentraleSupélec, France. In 2015, she was a Visiting Scholar with Virginia Tech. She is currently a Postdoctoral Researcher with Princeton University. Her research interests are in the broad areas of game theory, learning under uncertainty, and reinforcement learning. Applications of particular interest include wireless communications, Internet-of-Things, pricing, autonomous multiagent systems, security, and privacy.

Niraj K. Jha (Fellow, IEEE) received the B.Tech. degree in electronics and electrical communication engineering from the Indian Institute of Technology (I.I.T.) Kharagpur, Kharagpur, India, in 1981, and the Ph.D. degree in electrical engineering from the University of Illinois at Urbana–Champaign, Champaign, IL, USA, in 1985.

He has been a Faculty Member with the Department of Electrical and Computer Engineering, Princeton University, since 1987. He has coauthored five widely used books. He has published more than 450 papers (H-index: 81). His research interests include smart healthcare, cybersecurity, and machine learning algorithms/accelerators. He has received the Princeton Graduate Mentoring Award. His research has won 14 best paper awards, six award nominations, and 23 patents. He was given the Distinguished Alumnus Award by I.I.T., Kharagpur. He has served as the Editor-in-Chief for IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS and an associate editor of several journals. He has given several keynote speeches in the areas of nanoelectronic design/test, smart healthcare, and cybersecurity. He is a Fellow of ACM.