Journal of Data Science 19 (2), 293–313
April 2021
DOI: 10.6339/21-JDS1007
Computing in Data Science

AICov: An Integrative Deep Learning Framework for COVID-19 Forecasting with Population Covariates

Geoffrey C. Fox[1], Gregor von Laszewski[1,*], Fugang Wang[1], and Saumyadipta Pyne[2,3]

[1]*Digital Science Center, Indiana University, Bloomington, Indiana, USA*
[2]*Public Health Dynamics Laboratory, and Department of Biostatistics, University of Pittsburgh, Pittsburgh, Pennsylvania, USA*
[3]*Health Analytics Network, Pennsylvania, USA*

Abstract

The COVID-19 (COrona VIrus Disease 2019) pandemic has had profound global consequences on health, economic, social, behavioral, and almost every major aspect of human life. Therefore, it is of great importance to model COVID-19 and other pandemics in terms of the broader social contexts in which they take place. We present the architecture of an artificial intelligence enhanced COVID-19 analysis (in short AICov), which provides an integrative deep learning framework for COVID-19 forecasting with population covariates, some of which may serve as putative risk factors. We have integrated multiple different strategies into AICov, including the ability to use deep learning strategies based on Long Short-Term Memory (LSTM) and event modeling. To demonstrate our approach, we have introduced a framework that integrates population covariates from multiple sources. Thus, AICov not only includes data on COVID-19 cases and deaths but, more importantly, the population's socioeconomic, health, and behavioral risk factors at their specific locations. The compiled data are fed into AICov, and thus we obtain improved prediction by the integration of the data to our model as compared to one that only uses case and death data. As we use deep learning our models adapt over time while learning the model from past data.

Keywords *Cloudmesh; comorbidities; prediction; risk factors*

1 Introduction

The COVID-19 pandemic has caused an enormous health and humanitarian crisis worldwide. It is unlike any other single phenomenon that has occurred in modern history since the end of World War II. The pandemic's effects have spanned over a range that is so vast over space, and yet so condensed over time, that the dual blows of intensity and rapidity have exposed myriad systemic vulnerabilities in many societies around the world.

The first known case was traced back to 17 November 2019. Since the emergence of a cluster of cases in Wuhan, China, on 31 December 2019, COVID-19 has spread rapidly worldwide. Just six months later, as of 23 June 2020, there are 8,993,659 COVID-19 cases and 469,587 deaths globally. On 30 January 2020, the World Health Organization (WHO) (Welt Health Organization, 2020) declared COVID-19 to be a Public Health Emergency of International Concern and subsequently, on 11 March 2020, a pandemic.

*Corresponding author. Email: laszewski@gmail.com.

Received July 1, 2020; Accepted January 23, 2021

In the United States, the first case of COVID-19 was confirmed on 20 January 2020 in Washington State, and as of 23 June 2020, there were over 2.3 million confirmed cases and 121,167 deaths (New York Times, 2020a). Although initial efforts to reduce the spread took place, recent data shows a resurgence at an increased scale. In the absence of a vaccine or treatment to effectively combat the disease, non-pharmaceutical policy interventions such as social distancing and lockdowns are recommended by health experts to prevent further transmission. To gain insights into the possible impact of such measures on COVID-19 outcomes, we depend upon the ability to accurately forecast the spread of reported cases and confirmed deaths and recoveries. Naturally, the accuracy of forecasting relies on the availability of current, reliable data and historical data to determine estimates of uncertainty.

During outbreaks of epidemics, initially data on cases and deaths could be scarce, and the quality of data annotation, validation, and aggregation might be uncertain. For instance, changes in clinical data entry, such as the addition of a new category clinically diagnosed to the existing lab-confirmed category, may likely have been reflected in the reporting (Petropoulos and Makridakis, 2020). Initial fears about COVID-19 among the general population with regards to the trajectory of the pandemic could also affect administrative reporting. Diverse media sources and differences in local and federal policies may add to the general uncertainty about disease progression. Nevertheless, a data-driven approach to forecasting can offer valuable insights into the disease dynamics, and thereby an ability to objectively plan for the near future.

Unlike earlier global outbreaks, during COVID-19, the current worldwide digital ecosystem allows for real-time data collection, which in turn is used by artificial intelligence (AI) and deep learning systems to understand healthcare trends, model risk associations, and predict outcomes (Ting et al., 2020). In addition to traditional public health surveillance strategies available to most countries, a variety of static and dynamic data types may be integrated to model the scale and dynamics of this pandemic (Pyne et al., 2015). Several organizations, including the Johns Hopkins University's Center for Systems Science and Engineering, the New York Times, the Atlantic's COVID-19 tracking project, among others have developed real-time tracking maps for following COVID-19 cases around the world using data from the U.S. Centers for Disease Control and Prevention (CDC), WHO, and other international health agencies. Notably, the CDC receives forecasts from several modeling groups that use a wide variety of approaches ranging from SEIR (Susceptible-Exposed-Infectious-Recovered) to Agent-based to Bayesian models to understand the non-pharmaceutical policy interventions (or the lack thereof) to predict disease dynamics and its impact on human lives (CDC, 2020b).

For precise and contextually relevant modeling of COVID-19 dynamics for a given population or community, we use data integration to combine community-specific health and behavioral risk factors, demographic and socioeconomic variables that are available for the corresponding county along with the cases and deaths data for COVID-19 while leveraging the recently developed and increasingly popular deep learning strategies to forecast COVID-19 dynamics using our model. Towards this, we develop a parallel computing platform called AICov standing for AI-driven Platform for COVID-19 to implement the different components of our framework and obtain the needed resources through multi-cloud interfaces in a robust manner.

In this study, while we incorporate information from community and county-specific covariates to inform our forecasting model for each metropolitan area, we want to avoid the so-called ecological fallacy in drawing inferences about individual disease outcomes based on large area-level aggregated risk factors. Our objective is also to underscore the nuanced roles played by pre-existing socioeconomic and other prevailing conditions of a given community that can act as determinants of its health outcomes and possible disparities, especially under such sudden

stresses to its current systems as those felt during any pandemic. It is important to note that the underlying models are learned directly from the data instead of being projected by a theoretical epidemiological model that relies on simulation populations. This allows quick adaptation and self-adaptation in regards to changing behaviors and regional differences.

The paper is structured as follows. In Section 2, we provide a short overview of deep learning-based time series forecasting that we use in AICov. In Section 3, we outline our architecture that enables the user to conduct powerful time series forecasting by integrating static and dynamic information on population risk factors with time-series disease data. In Section 4, we describe the input data for AICov. Next, we demonstrate our analysis involving the population risk factors and identify those that contribute to our prediction accuracy. Finally, we present our conclusion.

2 LSTM

This section presents a short overview of the recurrent neural network algorithms used in this work.

2.1 LSTM Background

In this study, we base our work on a traditional LSTM algorithm for our predictions (Keras, 2015; Hochreiter and Schmidhuber, 1997). An LSTM is a Recurrent Neural Network (RNN) (Rumelhart et al., 1986) with feedback connections allowing the use of data input sequences to predict data output sequences. LSTMs have been applied to many application areas from handwriting (Graves and Schmidhuber, 2009), image, feature detection, and time series prediction (Schmidhuber et al., 2005). LSTM's have an internal state that is used to prevent the vanishing gradient problem (Hochreiter, 1991) during the training of RNNs.

Different variants of LSTM algorithms exist. Ours is based on an LSTM cell depicted in Figure 1.

It has an input gate, output gate, and a forget gate. The cell is maintaining how a subsequent value is calculated. This includes (a) how input values are influencing the cell via the input gate, (b) how the forget gate influences a memory value within the cell via the forget gate, and (c) how

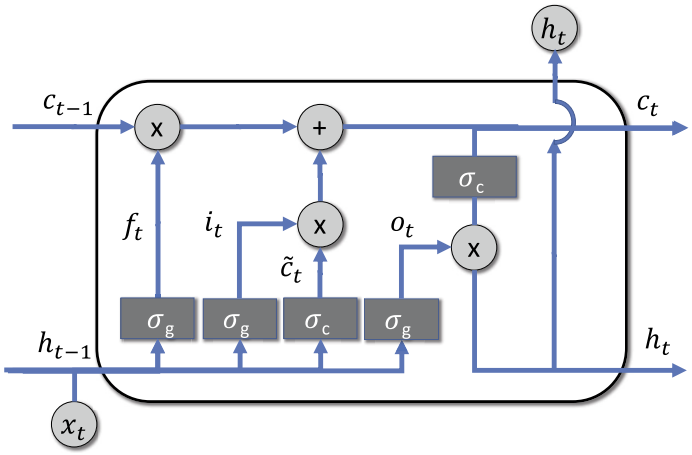


Figure 1: LSTM Unit Diagram.

the output gate influences the output activation of the LSTM unit. A cell has a number of inputs and outputs that are weighted. During a training step, these weights are learned to be reused in a prediction process between input and output sequences. The LSTM network will have a number of input values represented as vector x and a number of output values in the last layer of the network predicting a number of future values (Greff et al., 2017). For our experiments this includes the numbers of days we use as input to our prediction and the number of days we use as output of our prediction for a single cell. In our implementation the number for inputs and outputs can be adjusted. We repeat this prediction over all input sequences to create the output forecast sequences. Next we list the equations (1)–(6) used in LSTM. We use in the equations $\circ$ as the Hadamard product, σ_g is the sigmoid function for the gate. With D being the number of days as input features and H being the number of hidden layers, the inputs and outputs are defined by $x_t \in \mathbb{R}^D$ the input vector to the LSTM unit and $h_t \in \mathbb{R}^H$ the hidden state vector e.g., the last of which is the output vector of the LSTM. In our case the output contains the days we predict based on the input features. The equations are

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f), \tag{1}$$
$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i), \tag{2}$$
$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o), \tag{3}$$
$$\tilde{c}_t = \sigma_c(W_c x_t + U_c h_{t-1} + b_c), \tag{4}$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t, \tag{5}$$
$$h_t = o_t \circ \sigma_h(c_t), \tag{6}$$

where $f_t \in \mathbb{R}^H$ is the forget gate's activation vector, $i_t \in \mathbb{R}^H$ is the input/update gates activation vector, $o_t \in \mathbb{R}^H$ output gate's activation vector, $\tilde{c}_t \in \mathbb{R}^H$ the cell input activation vector $c_t \in \mathbb{R}^H$ the cell state vector, σ_c is the cell activation function, $W \in \mathbb{R}^{H\times D}$, $U \in \mathbb{R}^{H\times H}$ and $b \in \mathbb{R}^H$ being the weight matrices and bias vector parameters which are learned during training. In particular we denote W_q and U_q contain the weights of the input and recurrent connections, where the subscript denotes either the input gate i, the output gate o, the forget gate f or the memory cell c. For a more detailed discussion about LSTM's we refer to Greff et al. (2017).

2.2 Covariate Enhanced LSTM

We often think of the laws of physics described by operators that evolve the system given sufficient initial conditions and in this language, we have shown how to represent Newton's law operator by a recurrent network (Kadupitiya et al., 2020). We expect that the time dependence of many complex systems: Covid pandemics, Southern California earthquakes, traffic flow, security events can be described by deep learning operators that both capture the dynamics and allow predictions.

In this paper, we adopt this approach to describe the time dependence of COVID-19 infection and fatality counts in different regions. We intend to process the two time-series (infections, fatalities/deaths) associated with each city plus the covariates (fixed in time but dependent on region) to build a model for the time evolution of the data. There is a nontrivial architecture design choice as to the method of handling the time-independent covariates. We could combine a recurrent network for the time series with some fixed network for the covariates. However, that does not fit with our concept of deriving an evolution operator for COVID-19 data that naturally depends on the covariates for each city. Thus instead, we feed the covariates into the recurrent network as input in addition to the days.

3 AICov Architecture

The motivation for creating this architecture is based on our experience with deep learning toolkits such as Keras and PyTorch. While these systems provide the necessary APIs to integrate time series solutions, they target primarily more general user communities. In particular, they do not consider existing data sets or specific needs and analysis options to evaluate them for the purpose of COVID-19 risk factor data integration. We describe below the different requirements that motivate the design of our architecture framework, AICov, and how we address each of these.

Requirement 1 (Ease of Use)**.** A major design criterion for AICov is that the framework must be easy to use, and extensions can be made to allow usability of both the Application Programming Interfaces (API) as well as the user interface. To address these requirements, we are devising a specialized but easy to use time series API for COVID-19 that integrates common tasks such as automated filtering and normalization of the data.

Requirement 2 (Interactive Access)**.** Due to the experimental nature of analyzing the data, AICov must support the exploration of the data in an interactive fashion. As many data scientists use Jupyter Notebooks to integrate their work interactively, AICov integrates such notebooks. Jupyter notebooks allow rapid modification of the analysis workflow through the ease of Python as programming language. Additionally, they enable us to formulate easily sophisticated scientific analytics workflows with the help of Python and Jupyter Notebooks.

Requirement 3 (Expandable API)**.** Assuming that new models and other analysis algorithms will be developed over time, AICov must allow the integration of such models through programs and APIs. To address this requirement, we are developing an abstract API for data, analysis, and metadata parameter adjustments. External services can be integrated with the help of REST (REpresentational State Transfer) services. The REST services are automatically generated from function specifications (von Laszewski et al., 2020a).

Requirement 4 (Flexible Data Source Integration)**.** The integration of various sources of data is a key part of AICov. To achieve this goal, we provide a number of abstractions and data manipulation functions to extract needed data from established sources. Furthermore, the data can be combined, and additional data wrangling by external groups can be integrated. Through such abstractions, it is possible to replace, add, and correct data sets that are part of the analysis. This allows us to compare different results created from different data sets. Moreover, we include data selection directly in our framework to easily and quickly reuse them in a data mashup. An important example is that the data need not be static over the disease's progression but can be continuously updated. This also has a significant impact on our analytics algorithms.

Requirement 5 (Automated Update of the Analytics)**.** As the data can change, previous analyses may have to be updated. Our forecast is not just a single script but includes a mechanism to register multiple workflows of continuously integrated (CI) analysis that are automatically rerun when new data are made available to the system. Previous results are maintained. Metadata with these runs can retrieve the versioned data sets used to calculate and store the versioned analytics workflow.

Requirement 6 (Flexible Model definition)**.** We want to allow experimentation with various models in AICov that may be easily and flexibly definable as workflows. We give functional abstraction definitions that enables us to define the models ourselves or integrate third party models describing the spread of the disease.

Requirement 7 (Deep Learning Forecast)**.** As the forecasting models depend on changing data that are made available daily, it is important for AICov to utilize the newly available data. Many models can be applied to this, including moving averages based models or models that are purely derived from deep learning. To address this requirement, AICov allows for the integration of both. Furthermore, in our deep learning framework, we can integrate an automated search for important risk factors, which is the focus of this study.

Requirement 8 (Model Orchestration)**.** To coordinate the different model predictions and the generation of the deep learning forecasts, we need the ability to orchestrate them while applying a number of parameters. To address this requirement, the AICov architecture includes the ability to integrate parameter sweeps to, for example, identify hyperparameters, or integrate different data sets as parameters to identify suitable forecast models that can deliver the one which fits the best.

Requirement 9 (Compute Resource Mapping)**.** For other uses of AICov, we need to integrate a flexible resource utilization framework. To address this requirement, we leverage from our earlier work and utilize through Cloudmesh and National Institute of Standards and Technology (NIST) Big Data Working Group (NBDIF) definitions cloud services into the architecture (Chang and von Laszewski, 2019). This includes containers but also infrastructure services in the cloud. Through the service integration, we also provide access to Graphics Processing Units (GPUs).

Requirement 10 (Convenient Interfaces)**.** As we have a wide variety of users, we need to enable interfaces that are used by the various communities. To address this requirement, we provide a number of APIs, REST services and make them available via Jupyter Notebooks. In addition, we developed some custom widgets for the notebooks that are specifically targeted towards data integration, parameter manipulation, and visualization of the results.

Putting all these requirements together results in an architecture as shown in Figure 2.

In summary, the AICov architecture allows the inclusion of new data sources and models. An orchestrator enables modifying parameters and selecting data sources and models used in the analysis. The framework can be accessed via the interface layer that includes APIs, REST, and

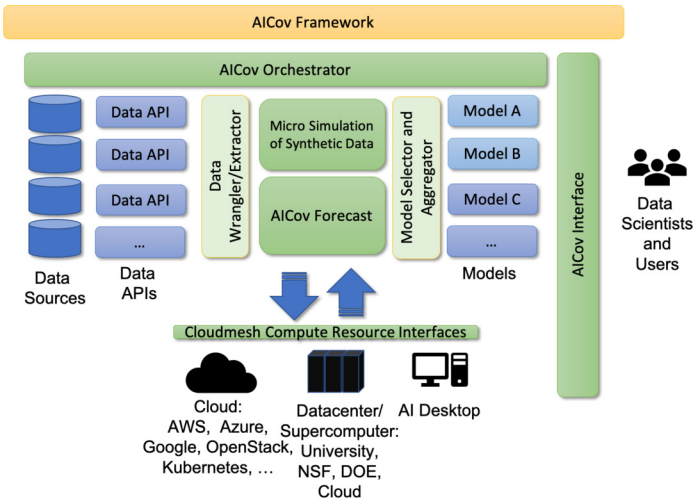


Figure 2: AICov Architecture.

Jupyter Notebooks. Compute interfaces are provided either by direct access of GPUs from local resources or through the staging of automatically generated REST services via our Cloudmesh REST Service integrator (von Laszewski et al., 2020a,b; von Laszewski, 2020). Additionally, we can also access through abstractions cloud services that are offered by the various cloud providers and are targeting our application domain. Different models can be easily integrated into our framework to verify our forecast or enhance our forecast based on the users' models and data.

4 Data for the COVID-19 Analysis

While the U.S. has seen an overall case fatality rate of 5.14%, some U.S. locations have experienced a disproportionate number of deaths compared to others. This disparity of deaths among different communities could be attributed to a diversity of local risk factors and socioeconomic determinants of health that now constitute a very active area of investigation. Researchers are studying the effects of different community-specific socioeconomic variables, underlying health conditions and comorbidities, infrastructure, and systemic responses. Given the complexity of such underlying conditions that can determine individual disease outcomes, where and under what conditions people get infected by the virus, and then recover or die take place not in isolation but rather in an interplay of diverse factors that are characteristic to a local community, which need to be synthesized. In other words, a basic requirement for effective modeling of COVID-19 dynamics is that the information regarding a particular community's vulnerabilities and strengths pertaining to the disease must be as comprehensive and integrative as possible. The regular reporting of accurate Coronavirus outbreaks data from local health departments has been understandably difficult, especially in the areas that are hardest hit.

We use data on community-specific health and behavioral risk factors aggregated at the county level from the large-scale longitudinal CDC Behavioral Risk Factor Surveillance System (BRFSS) surveys (CDC, 2020a) and demographic and socioeconomic data from the U.S. Census Bureau US Census Bureau (2020a) for each county, along with the cases and deaths data compiled by the CDC National Center for Health Statistics (NCHS) (CDC, 2020c) and integrate them with other data sources such as the American Hospital Directory (American Hospital Directory, 2020).

4.1 Data Collection and Processing

Among the different sources of data on daily cases and deaths data collected and aggregated at different spatial levels, especially for the U.S., many are either freely available repositories such as GitHub or accessible via specifically developed online resources such as the New York Times (New York Times, 2020b) and the Johns Hopkins Coronavirus Resource Center (Johns Hopkins Coronavirus Resource Center, 2020). For this study, we generated two data sets based on the latter resource upon our checking for the locations and times at which cumulative counts of COVID-19 cases and deaths on any given day were no less than those on the previous day. This led us to compile daily coronavirus cases and deaths data for 345 U.S. counties from 32 states and the District of Columbia and matched by five-digit Federal Information Processing Standards (FIPS) code or county name to dynamic and static variables from additional data sources.

The static data in this repository was collected from the U.S. Census Bureau and the Centers for Disease Control and Prevention (CDC). The different fields cover the domains of

behavioral and health risk factors, hospital capacity, and socioeconomic and demographic conditions. Data on health outcomes, prevention measures, and unhealthy behaviors were extracted from the CDC's 500 Cities Local Data for Better Health program (Centers for Disease Control and Prevention, 2020a) in the form of crude or age-adjusted prevalence of conditions such as respiratory disease, obesity or smoking. These values were aggregated from the census tract to the county level using the first five digits of the eleven-digit tract FIPS code. Because a single county may consist of multiple individual cities, we include the list of all city labels within each aggregate group to represent a greater metropolitan area. 110 of such metropolitan areas that had more than 500 reported cases of COVID-19 by April 15, 2020, were selected for this study.

Notably, a greater metropolitan area such as New York City could be spread across more than one county, which can lead to the complexity of aggregating counts and other variables. Time series data for the 5 individual counties (boroughs) in New York City were not available in the Johns Hopkins CSSE (Center for Systems Science and Engineering) data set. Rather, the total number of cases and deaths for the entire city of New York are reported and assigned the FIPS code of New York County (Manhattan). To ensure consistent geography across the static and dynamic data, we compute the population-weighted sum or median of each covariate over the assigned counties.

Demographic variables were gathered from the Census QuickFacts (US Census Bureau, 2020b) online resource using an automated web scraping algorithm and cover relevant areas such as age, race, income, and population density. Additional socioeconomic variables include the Gini Index, which measures economic inequality and CDC Social Vulnerability Index (SVI) (Centers for Disease Control and Prevention, 2020b). The SVI was created to guide public health officials and disaster response efforts by identifying the communities across the United States most likely to need support during a crisis. Census tracts and corresponding counties are ranked across 15 social factors, which are grouped into four themes: socioeconomic status, household composition and disability, minority status and language, and housing and transportation.

Lastly, the total number of general acute care, critical access, and military hospitals within each county are included in the data. Such variables included the number of relevant hospitals per county, and the estimated number of beds (total known bed counts added to the number of hospitals in the county with missing data times the average number of beds per hospital in that state) per 1,000 people using the American Hospital Directory (American Hospital Directory, 2020). The list of covariates is presented in Table 1.

5 Analysis

We have conducted a number of analysis efforts to evaluate the validity of our approach. We focus on the following questions:

Question 1. Can we apply deep learning strategies that self-learn the behavior of a model in order to provide future predictions? See Section 5.1.

Question 2. Can we use real time data as input and apply deep learning strategies instead of models that self learn and provide future predictions? See Section 5.2.

Question 3. Does the inclusion of covariates improve the prediction quality while including geospatial and socio-economic risk factors? See Section 5.2.

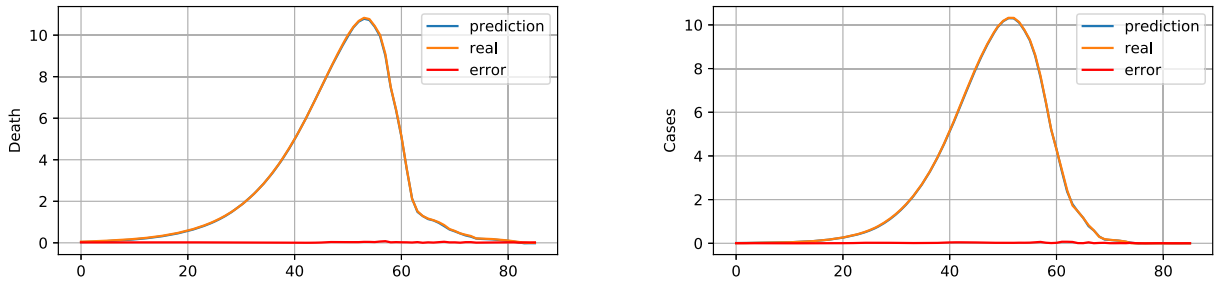
Question 4. Are there some impact factors that influence the prediction more than others? See Section 5.2.

Table 1: Covariate Risk factor abbreviations as used in this study.

Risk Factor	Description
ARTHRITIS	Percent reported with arthritis
BINGE	Percent reported with binge drinking
BLACK	Percent of Blacks in the population
BPHIGH	Percent reported with high blood pressure
BPMED	Percent reported taking blood pressure medication
CANCER	Percent reported as cancer patients
CASTHMA	Percent reported with current asthma
CHD	Percent reported with coronary heart disease
CHECKUP	Percent reported with health checkup
CHOLSCREEN	Percent reported with cholesterol screening
COPD	Percent reported with chronic obstructive pulmonary disease
CSMOKING	Percent reported as currently smoking
DIABETES	Percent reported with diabetes
ESTBEDS	Number of estimated beds in hospitals
HIGHCHOL	Percent reported with high cholesterol
HISPANIC	Percent of Hispanic in the population
INSURANCE	Percent reported with insurance
KIDNEY	Percent reported with kidney disease
LPA	Percent reported with no leisure-time physical activity
MHLTH	Percent reported with not good mental health
NBEDS	Number of beds in hospitals
NBEDS/1000	Number of beds per 1000 people
NHOSP	Number of hospitals
NONE	No risk factor used
OBESITY	Percent reported with population that is obese
PHLTH	Percent reported with physical health issues
POP_DENSITY_2010	Population Density from the 2010 Census
POVERTY	Percent of poverty in the population
SENIOR	Percent of seniors in the population
STROKE	Percent reported with stroke
SVI_MINORITY	Percent of social vulnerability index in the minority population
SVI_OVERALL	Percent of social vulnerability index in the overall population

5.1 Deep Learning Predictions of Model-based Empirical Fits

To establish the feasibility of using LSTM, we have, in our analysis, verified that through deep learning we can recreate the predictions identified by a model-based approach using empirical fits. In this comprehensive analysis, we obtain predictions through deep learning while using data generated from epidemiological models as discussed in Marsland and Mehta1 (2020). The time series we used was 100 days long, and a multi-layered LSTM recurrent network was used. Our prediction approach differs by learning not only from the demographics (fixed data for each city) and time-dependent data but by integrating the population model for the underlying prediction, as shown in Figure 3. Such model predictive integration capability is important in



(a) Prediction accuracy for deaths.

(b) Prediction accuracy for cases.

Figure 3: Cumulative empirical fits can be very accurately predicted with LSTM. The data is taken from 37 U.S. Cities. Our comparison showcases that given a model, we can replicate the prediction of such models using the model data over time. The x-axis represents the days since the first data was fed to the model.

any application with multiple time scales. For example, this allows us to integrate multi-scale time effects into the forecast, which could address the combination of general forecasts and the next time steps of choice, such as days, weeks, months, or longer. For this analysis, we used 37 of the 110 cities with reliable empirical fits (that we derived from an epidemiological model and not deep learning (Marsland and Mehta1, 2020)) for the cases and deaths data up to April 15, 2020. Our analysis identified a sophisticated single deep learning time evolution operator that can describe these 37 separate data sets, and smooth fitted data leads to very accurate deep learning descriptions via LSTM as depicted in Figure 3 that are smaller than 1% from the model predictions.

5.2 Realtime-Data Covariate LSTM Prediction

In this section, we establish that using deep learning can not only be applied to model predictions but also from real time data. As we want to evaluate the influence of the risk factors on our prediction framework using not only models but deep learning algorithms, we have devised a parameter sweep framework based on LSTM that conducts the prediction for 110 cities while also considering the risk factors as listed in Table 1. This framework can query any combination of risk factors as well as not including any, in which case it defaults to the traditional LSTM algorithm as introduced in Section 2. Hence, we will answer Questions 2 and 3.

To answer these questions, we chose for our next analysis a date range between 2020-02-01 and 2020-05-25. We start with a basic LSTM operator with two layers of LSTM, and initial and final fully connected layers (Kadupitiya et al., 2020). For LSTM, we have selected the activation function based on Rectified Linear Units (RELU). For the last layer sigmoid is chosen. A drop out rate of 0.2 is used, we use 16 input nodes, and have a batch size of 110. The batch size is the subset size of the training sample (e.g. 110 out of 12540 for a 3 day input length) that is used to train the LSTM during its learning process. The maximum epoch is chosen to be 200. This is the number of steps after we interrupt the learning process. The number of nodes internally is 32. The maximum total number of samples is 12540 when using 3 days as input length. We use a two-layer LSTM, as shown in Table 2. Within our system, we have the ability to set a number of input days. For this analysis, we have chosen three different input lengths, namely 3, 4, and 5 days. The term “None” in the output shape means this dimension is variable.

Table 2: The deep learning parameters used for the model sweep creation in as exported by Keras using 1 risk factor. The activation function is RELU and the recurrent activation function is sigmoid.

Layer	Type	Output Shape	Parameters for 0 risk factor	Parameters for 1 risk factor	Output Shape	Parameters for 33 risk factors	Output Shape	Parameters for 37 risk factors
dense	Dense	(None, 1, 32)	96	128	(None, 5, 32)	1152	(None, 5, 32)	1280
LSTM	LSTM	(None, 1, 16)	3136	3136	(None, 5, 16)	3136	(None, 5, 16)	3136
LSTM 1	LSTM	(None, 16)	2112	2112	(None, 16)	2112	(None, 16)	2112
Dense 1	Dense	(None, 32)	544	544	(None, 32)	544	(None, 32)	544
Dense 2	Dense	(None, 30)	990	990	(None, 30)	990	(None, 30)	990

The LSTM contains a total of 6,910 trainable parameters in case we include one risk factor. In the case, we do not use any risk factor it is 6878, and when we include 33 risk factors, it is 7934.

We have run the training on all 110 cities from our data that represented each FIPS while not including any risk factors (Question 2) and running them with a single risk factor while iterating over all risk factors (Question 3). We then summed up all absolute errors for the predicted data points over the same time period and repeated each experiment 10 times. We summarized all the results in the form of a box-whisker diagram each for deaths and for cases. We sorted the diagram in such a fashion that the risk factor that has the model with the lowest error appears first. We preceded the graph with the experiment labeled as "NONE" that which does not use any risk factors to provide a comparison. The graphs are depicted in Figures 4 for cases, and 5 for deaths.

We show the errors and the association with a risk factors in Tables 3 and 4. In the tables we include the following columns (a) the Risk Factor (b) the root mean square error of all models for this risk factor (c) the minimum cumulative error by risk factor and (d) the number of days that were used as input and identified to relate to the model with the minimum cumulative error. The table has in the first column the rank this particular model compares to all other models with minimal risk factor. The rank was obtained by sorting the best model for each risk factor. We display two tables. One for cases (Table 3) and one for deaths (Table 4).

We see that in our experiment, almost all risk factors lead to an increased model prediction accuracy for cases. We note that by including factors such as population density, physical health Insurance, population breakdown by ethnicity, number of hospitals, and diabetes, lead to better overall predictions for cases. We also see that many of these factors perform better on average.

Table 3: Best models for cases for each risk factor and their respective errors. The model that uses no risk factor is highlighted in grey. We present the Root Mean Square Error (RMSE), the cumulative error and how many days as input were used for this corresponding model.

Place/Rank	Risk Factor	RMSE Cases	Minimum Cumulative Error for Cases by Risk Factor	Number of Days as Input
1	POP_DENSITY_2010	0.0554	7.94	5
2	PHLTH	0.0548	8.01	3
3	BLACK	0.0550	8.12	4
4	HISPANIC	0.0561	8.22	5
5	NHOSP	0.0546	8.26	3
6	INSURANCE	0.0546	8.34	3
7	CHOLSCREEN	0.0547	8.37	3
8	DIABETES	0.0546	8.4	5
9	STROKE	0.0542	8.45	4
10	CHD	0.0540	8.46	5
11	CHECKUP	0.0549	8.51	5
12	NBEDS	0.0541	8.54	5
13	SVI_OVERALL	0.0548	8.58	5
14	KIDNEY	0.0546	8.61	5
15	CASTHMA	0.0551	8.63	5
16	BPMED	0.0547	8.74	3
17	LPA	0.0543	8.75	5
18	CSMOKING	0.0537	8.75	5
19	ARTHRITIS	0.0541	8.83	3
20	POVERTY	0.0550	8.84	3
21	ESTBEDS	0.0545	8.86	4
22	MHLTH	0.0549	8.9	3
23	COPD	0.0556	8.93	4
24	SENIOR	0.0551	8.93	3
25	CANCER	0.0545	8.99	3
26	NBEDS/1000	0.0544	9.05	3
27	BPHIGH	0.0546	9.13	4
28	NONE	0.0549	9.26	3
29	HIGHCHOL	0.0564	9.28	3
30	BINGE	0.0543	9.32	5
31	OBESITY	0.0545	9.5	3
32	SVI_MINORITY	0.0551	9.54	4

Table 4: Best models for deaths for each risk factor and their respective errors. The model that uses no risk factor is highlighted in grey. We present the Root Mean Square Error (RMSE), the cumulative error and how many days as input were used for this corresponding model.

Place/Rank	Risk Factor	RMSE Deaths	Minimum Cumulative Error for Deaths by Risk Factor	Number of Days as Input
1	SVI_OVERALL	0.0675	18.75	5
2	CSMOKING	0.0669	19.01	5
3	HISPANIC	0.0676	19.04	5
4	NONE	0.06696	19.09	4
5	POP_DENSITY_2010	0.0670	19.13	4
6	SENIOR	0.0685	19.27	3
7	CHOLSCREEN	0.0679	19.43	5
8	CASTHMA	0.0670	19.46	5
9	INSURANCE	0.0671	19.48	5
10	MHLTH	0.0687	19.53	3
11	LPA	0.0671	19.54	5
12	POVERTY	0.0673	19.57	5
13	OBESITY	0.0671	19.68	5
14	CANCER	0.0687	19.70	3
15	NBEDS/1000	0.0675	19.73	5
16	BLACK	0.0684	19.83	3
17	DIABETES	0.0676	19.84	5
18	ARTHRITIS	0.0669	19.94	4
19	NBEDS	0.0669	19.95	4
20	NHOSP	0.0674	19.95	5
21	BPHIGH	0.0678	19.99	5
22	PHLTH	0.0687	20.02	3
23	CHECKUP	0.0675	20.02	5
24	CHD	0.0687	20.15	3
25	STROKE	0.0669	20.16	4
26	HIGHCHOL	0.0686	20.35	3
27	ESTBEDS	0.0671	20.42	4
28	KIDNEY	0.0689	20.49	3
29	BINGE	0.0674	20.56	5
30	BPMED	0.0674	20.69	5
31	COPD	0.0675	20.98	5
32	SVI_MINORITY	0.0674	21.53	4

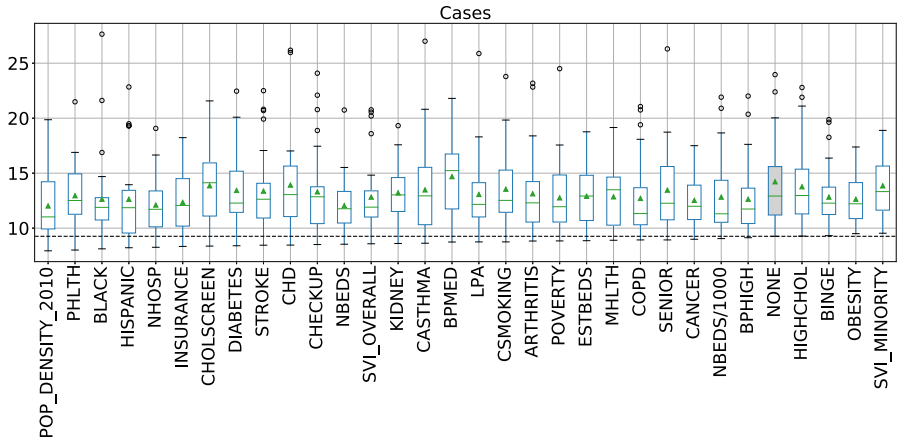


Figure 4: Influence of risk factors on the accuracy of the prediction of COVID-19 cases. The y-axis represents the cumulative error over all input data for the cities. The x-axis labels correspond to the order of models for minimum model error.

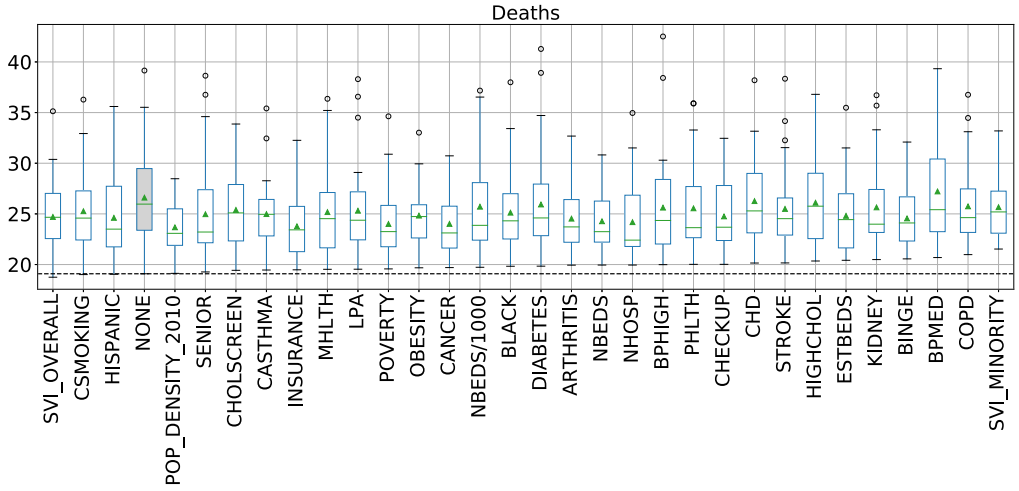
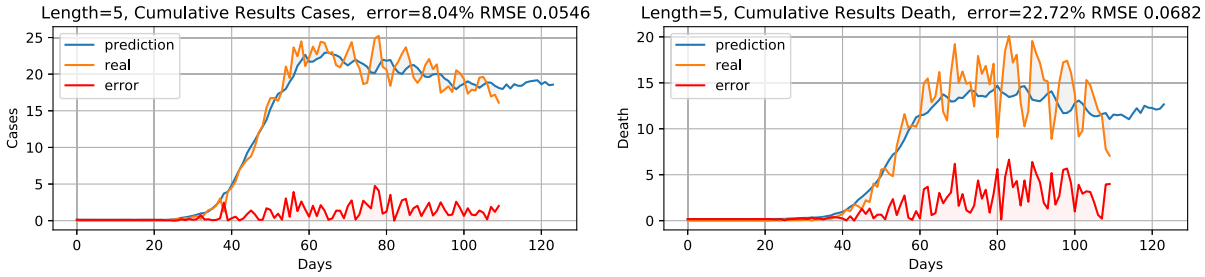


Figure 5: Influence of risk factors on the accuracy of the prediction of COVID-19 deaths. The y-axis represents the cumulative error over all input data for the cities. The x-axis labels correspond to the order of models for minimum model error.

We observed a different scenario for deaths, in which we only identified three risk factors, namely the Social Vulnerability Index (SVI), chain-smoking, and Hispanic that, when integrated into our deep learning model, leads to better prediction results. However, the accuracy of the overall prediction of deaths is far less precise than that of the cases. To showcase this, we have included Figures 6a and 6b that show the respective errors for example for the population density.

This can be explained by our input data distribution between cases and deaths in which we noted higher fluctuations relative to the overall value in cases of deaths. A mitigation to this issue would be using a seven day average over the analyzed period. However, we have not done this on purpose for this analysis as we wanted to identify how a covariate enhanced



(a) Model prediction error cases.

(b) Model prediction error deaths.

Figure 6: Model prediction error for cumulative deaths when including one risk factor POP_DENSITY_2010 in addition to the past number of cases and deaths.

LSTM behaves that includes risk factors based on the daily fluctuations. This was done to avoid and showcase any needed preprocessing on the data and to identify the capability of the deep learning framework while adapting to the fluctuations without any special activities. Previously, we have already shown in Section 5.1 that for smooth data inputs the prediction is very accurate. Using fluctuation data allows our framework for an automated ingest of data on a daily basis to enable data fusion of new cases and deaths information. Hence, we can re-train the model with new incoming data every day. Through repeated training experiments, we found the LSTM hyperparameters such as epochs and dropout values that work well for our experiments.

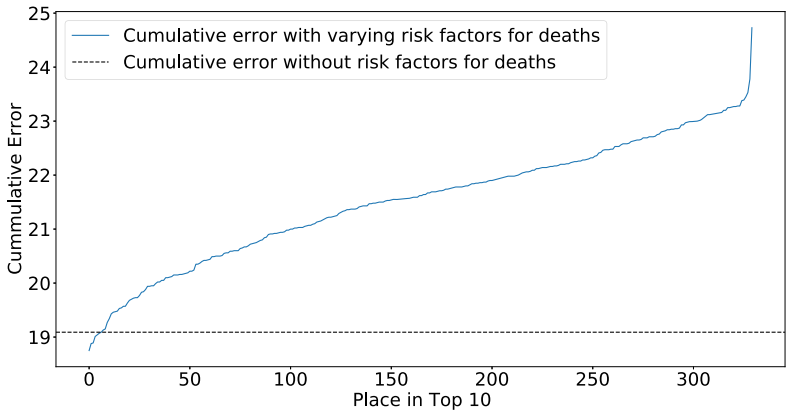
We conducted an additional analysis and plotted the top 10 best cumulative prediction model errors for each risk factor in Figure for cases and deaths. The horizontal black line shows the first occurrence of no risk factor being used.

Notably, we identified that a jump occurs when we sort the top 10 model predictions at around 320 models from a total of 3200 automatically generated models. This also underscores the need to run our deep learning model multiple times to obtain best parameter candidates. Furthermore, we see that the cumulative error for cases is about half the size as for deaths. This is caused by cases having a larger number of samples than those from deaths.

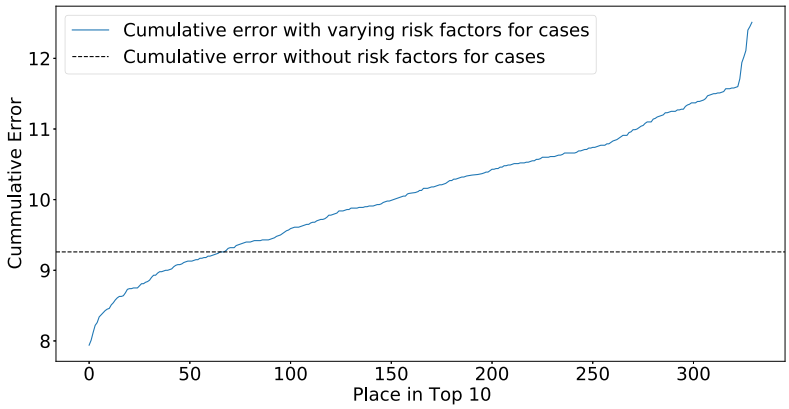
6 Conclusion

While this study focuses on if and which risk factors can improve the predictions, we do not focus on why. Such analysis could provide insights into causal relationships. For example, information such as health insurance and the number of hospitals and beds are often related to the quality of services that affect early detection and treatment. Furthermore, population parameters such as population density can determine the rate of spread and the number of cases. Similarly, distributions of comorbidities in a given population such as physical health, cholesterol screening, diabetes, etc., are found to be insightful in predicting outcomes (Maleki et al., 2020). Identifying such risk factors and correlating them to the prediction accuracy provides valuable candidates for future studies.

A variety of mathematical and computational models have been used for predicting COVID-19 outcomes (Jewell et al., 2020). It includes compartmental models such as the SIR (Susceptible-Infected-Recovered) and SEIR (Susceptible-Exposed-Infected-Recovered) which are relatively easy to compute, and thus, commonly used in epidemiology. However, their simplistic assump-



(a) Cumulative error for deaths.



(b) Cumulative error for cases.

Figure 7: Top 10 predictions from all different risk factors sorted by cumulative error for deaths (7a) and cases (7b). The horizontal line in each figure represents the model cumulative error with no risk factor as input for deaths and cases respectively.

tions, such as homogeneous mixing within populations, might make them less realistic for pandemics such as COVID-19. For instance, such models often assume a closed population that does not change. However, COVID-19 dynamics have prominently included massive migration, deaths due to comorbidities temporarily conferred immunity, and notably, major disparities in terms of socio-economic determinants of health.

In terms of implementation, SIR and SEIR models are fitted with point estimates, which may not adequately reflect the disease dynamics. For instance, it is difficult to account for behavioral changes (e.g., human mobility) or systemic inadequacies (e.g., availability of hospital beds) that may appear in specific subpopulations. Finally, the compartmental model estimates do not allow for quantification of uncertainty, which makes it harder to use them in policy-making. A comparison of modeling approaches concluded that while SEIR model performed better for a couple of states in the U.S., alternatives have performed better for the other states (Bertozzi et al., 2020). AICov has addressed most of the above modeling issues in its design, as per the requirements of its forecasting objectives.

We present AICov as an integrative deep learning framework for COVID-19 forecasting with the help of population covariates. Thus, its objectives are different from those of traditional compartmental models. One of the important features of the architecture for AICov is that it is by design targeting Cloud and High Performance Computing (HPC) resources to conduct parameter sweeps to leverage sophisticated deep learning toolkits. The architecture allows the integration of various data sources that can update the data from its sources on demand and update its model predictions based on newly introduced data. Parameters can easily be adjusted via Jupyter notebooks that, in turn, call the computational backends on HPC and cloud resources.

In addition to this architecture, we used data collected from multiple public sources and agencies, and integrated the same across spatially contiguous units such as cities or metropolitan areas. In our analysis, we have focused on 110 selected cities of the U.S., but the framework is general and can be used to analyze similar data on pandemics from anywhere in the world.

The comprehensive analysis showcases the feasibility of our approach. Based on the outcomes reported it can lead to an improved prediction once we integrate risk factors in addition to the time-dependent data such as cases and deaths resulting from COVID-19. While the improvements observed were modest, they do present a concrete way of improving the forecast models. Inclusion of further putative factors from the ongoing worldwide studies on Covid-19 will only strengthen the future applications of our integrative framework.

We have shown that deep learning can return very good results while using smooth data, which we used in our empirical fits. For real data as presented to us for the daily changes, it still produced good results and even was able to predict changes based on weekly fluctuations. This is typically not achieved by other non-data based model approaches. Also, we have used in our data only cases and deaths, but we intend to expand this by using data about recovery and, in the future, immunization.

We have experimented with different hyperparameters and included in this study a selection of hyperparameters that have worked well for this data set.

As we have set up the first version of our AICov software, we have used cloud and high-performance computers. All of our sophisticated analyses were run in a day on at most 16 computers. All deep learning algorithms that were run on GPUs were run on NVIDIA Tesla K80s. However, our framework is on purpose generalized so that other compute resources can be integrated and leveraged. This includes local, HPC, and cloud computing resources, as well as different GPUs. Naturally, we also need to take into consideration not only the compute time but also the time it takes for the researcher to derive the models. As our approach is self-learning based on automated data updates, the time to prepare an updated model can be automatized and minimal input is needed. Thus the overall effort is very competitive.

Acronyms

This section contains the list of acronyms in Table 5 as used in the paper for easy reference. The definitions of the risk factors are however given within the paper in Table 1 on page 301 and not repeated here.

Table 5: Non-risk factor acronyms used in this paper.

Abbreviation	Description
API	Application Programming Interfaces
BRFSS	Behavioral Risk Factor Surveillance System
CDC	U.S. Centers for Disease Control and Prevention
CI	Continuous Integration
COVID-19	COrona VIrus Disease 2019
CSSE	Center for Systems Science and Engineering
cum	cumulative
FIPS	Federal Information Processing Standards
GPUs	Graphics Processing Units
HPC	High Performance Computing
LSTM	Long Short-Term Memory
MSE	Mean Squared Error
NBDIF	NIST Big Data Working Group
NCHS	CDC National Center for Health Statistics
NIST	National Institute of Standards and Technology
NONE	No risk factors used
NORM_POP	Normalized Population
NSF	National Science Foundation
RELU	REctified Linear Unit
REST	REpresentational State Transfer
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
SEIR	Susceptible, Expose, Infectious, Recovered
SELU	Scaled Exponential Linear Unit
SIR	Susceptible, Infectious, Recovered
U.S.	United States of America
WHO	World Health Organization

Supplementary Material

The code and paper document represented to implement AICov are contained in several repositories:

1. The entire cloudmesh code on which the cloud based implementation of the AICov framework is based and contains over 70 contributors is available publicly at https://github.com/cloudmesh. Cloudmesh contains a number of modules that dependent on the users access to cloud resources can be customized. A detailed manual about the configuration is available at https://cloudmesh.github.io/cloudmesh-manual/.
2. The entire COVID-19 analysis leverages cloudmesh and uses Jupyter notebooks to coordinate its workflow as discussed in the architecture Figure 2. The code and data for the results presented in this paper are located in the repository at https://github.com/cloudmesh/cloudmesh-covid.

   The data was analysed on a variety of supercomputing resources including an allocation

of 20 compute nodes that were utilized to execute the repeated model creation to assure reproducible results.
However, the use of the data is copyrighted and must be authorized to be used for other publications without contacting the authors. The data gathering and analysis is a significant intellectual contribution and we like to avoid that the data is taken before we have not secured a publication.

3. The entire paper is located in LaTeXsource in the GitHub repository https://github.com/cyberaide/paper-covid. This repository will be open sourced after acceptance of publication to not violate any publisher restrictions. If desired the authors can grant access to this repository prior to publication. Please contact the corresponding author.

A zip file is provided for the publication for archival purposes. However, it will be much more convenient and easier to use our GitHub distribution as discussed in the supplementary section.

Acknowledgement

We thank J. Kadupitiya for several useful discussions on deep learning frameworks.

Funding

This work is partially supported by the National Science Foundation (NSF) through awards Cyberinfrastructure Framework for 21st Century Data Infrastructure Building Blocks (1443054), Network for Computational Nanotechnology Engineered nanoBIO Node (1720625), Cybertraining (1829704), CyberInfrastructure for Network Engineering and Science (1835598) and Global Pervasive Computational Epidemiology (1918626).

References

American Hospital Directory (2020). Information about hospitals from public and private data sources including medpar, opps, hospital cost reports, and other CMS files. Web Page. URL: https://www.ahd.com/.

Bertozzi A, Franco E, Mohler G, Short M, Sledge D (2020). The challenges of modeling and forecasting the spread of COVID-19. *Proceedings of the National Academy of Sciences*, 117(29): 16732–16738. https://www.pnas.org/content/117/29/16732, https://www.pnas.org/content/117/29/16732.full.pdf, doi: https://doi.org/10.1073/pnas.2006520117.

CDC (2020a). Behavioral risk factor surveillance system survey. Web Page. URL: https://www.cdc.gov/brfss/index.html.

CDC (2020b). Forecasts of total deaths. Web Page. URL: https://www.cdc.gov/coronavirus/2019-ncov/covid-data/forecasting-us.html#modeling-groups.

CDC (2020c). NCHS – National Center for Health Statistics. Web Page. URL: https://www.cdc.gov/nchs/index.htm.

Centers for Disease Control and Prevention (2020a). Open data for chronic disease and health promotion data and indicators. Web Page. URL: https://chronicdata.cdc.gov/.

Centers for Disease Control and Prevention (2020b). Social vulnerability index. Web Page. URL: https://svi.cdc.gov/.

Chang WL, von Laszewski G (2019). NIST Big Data Interoperability Framework: Volume 8, Reference Architecture Interfaces, *Technical report*, National Institute of Standards and Technology. URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1500-9r1.pdf.

Graves A, Schmidhuber J (2009). Offline handwriting recognition with multidimensional recurrent neural networks. In: *Advances in Neural Information Processing Systems* (D Koller, D Schuurmans, Y Bengio, L Bottou, eds.), volume 21, 545–552. Curran Associates, Inc.

Greff K, Srivastava RK, Koutník J, Steunebrink BR, Schmidhuber J (2017). LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10): 2222–2232.

Hochreiter S (1991). Untersuchungen zu dynamischen neuronalen netzen, *Technical Report Diploma thesis*, Technische Univ. Munich, Institut f. Informatik.

Hochreiter S, Schmidhuber J (1997). Long short-term memory. *Neural Computation*, 9(8): 1735–1780. doi: https://doi.org/10.1162/neco.1997.9.8.1735.

Jewell N, Lewnard J, Jewell B (2020). Predictive mathematical models of the COVID-19 pandemic. *JAMA*, 323(19): 1893–1894.

Johns Hopkins Coronavirus Resource Center (2020). COVID-19 map. Web Page. URL: https://coronavirus.jhu.edu/map.html.

Kadupitiya J, Fox GC, Jadhao V (2020). Simulating molecular dynamics with large timesteps using recurrent neural networks. arXiv preprint: https://arxiv.org/abs/2004.06493.

Keras (2015). Working with RNNs. URL: https://keras.io/guides/working_with_rnns/.

Maleki M, McLachlan G, Gurewitsch R, Aruru M, Pyne S (2020). A mixture of regressions model of COVID-19 death rates and population comorbidities. *Statistics and Applications*, 18(1): 295–306.

Marsland R, Mehta1 P (2020). Data-driven modeling reveals a universal dynamic underlying the COVID-19 pandemic under social distancing. arXiv preprint: https://arxiv.org/abs/2004.10666.

New York Times (2020a). Coronavirus in the U.S.: Latest map and case count – The New York Times. Web Page. URL: https://www.nytimes.com/interactive/2020/us/coronavirus-us-cases.html.

New York Times (2020b). An ongoing repository of data on coronavirus cases and deaths in the U.S. GitHub. URL: https://github.com/nytimes/covid-19-data.

Petropoulos F, Makridakis S (2020). Forecasting the novel coronavirus COVID-19. *PLoS One*, 15(3): e0231236. doi: https://doi.org/10.1371/journal.pone.0231236.

Pyne S, Vullikanti AKS, Marathe MV (2015). Chapter 8 – Big data applications in health sciences and epidemiology. In: *Handbook of Statistics* (V Govindaraju, VV Raghavan, CR Rao, eds.), volume 33, 171–202. Elsevier. doi: https://doi.org/10.1016/B978-0-444-63492-4.00008-3.

Rumelhart DE, Hinton GE, Williams RJ (1986). Learning representations by back-propagating errors. *Nature*, 323(6088): 533–536.

Schmidhuber J, Wierstra D, Gomez FJ (2005). Evolino: Hybrid neuroevolution/optimal linear search for sequence learning. In: *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence* (LP Kaelbling, A Saffiotti, eds.), Edinburgh, Scotland, UK, July 30–August 5, 853–858. Professional Book Center. URL: http://ijcai.org/Proceedings/05/Papers/1452.pdf.

Ting DSW, Carin L, Dzau V, Wong TY (2020). Digital technology and COVID-19. *Nature Medicine*, 26(4): 459–461. doi: https://doi.org/10.1038/s41591-020-0824-5.

US Census Bureau (2020a). Census.gov. URL: https://www.census.gov/.

US Census Bureau (2020b). QuickFacts: United States. Web Page. URL: https://www.census.gov/quickfacts/fact/table/US/PST045219.

von Laszewski G (2020). Cloudmesh manual. Web Page. URL: https://cloudmesh.github.io/cloudmesh-manual/.

von Laszewski G, Orlowski A, Otten RH, Markowitz R, Gandhi S, Chai A, et al. (2020a). Using gas for speedy generation of hybrid multi-cloud auto generated AI services, *Technical report*, Indiana University. Submitted for publication. URL: https://github.com/laszewski/laszewski.github.io/raw/master/papers/vonLaszewski-openapi.pdf.

von Laszewski G, et al. (2020b). Cloudmesh OpenAPI installation instructions. Web Page. URL: https://github.com/cloudmesh/cloudmesh-openapi/blob/main/README.md.

Welt Health Organization (2020). WHO coronavirus disease (COVID-19) dashboard. Web Page. URL: https://covid19.who.int/.