IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 21 April 2023, accepted 14 June 2023, date of publication 22 June 2023, date of current version 29 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3288487

RESEARCH ARTICLE

Static Seeding and Clustering of LSTM Embeddings to Learn From Loosely Time-Decoupled Events

CHRISTIAN G. MANASSEH[1], **RAZVAN VELICHE**[1], **JARED BENNETT**[1], **AND HAMILTON SCOTT CLOUSE**[2]

[1]Mobius Logic Inc., Tysons, VA 22102, USA
[2]Air Force Research Laboratory, Wright-Patterson Air Force Base, OH 45433, USA

Corresponding author: Christian G. Manasseh (cman@alum.mit.edu)

This work was supported by the Air Force Research Laboratory (AFRL) Autonomy Capability Team 3 (ACT3) under Contract FA8649-20-C-0130.

ABSTRACT Humans learn from the occurrence of events at different places and times to predict similar trajectories of events. We define loosely decoupled time (LDT) phenomena as two or more events that could occur in different places and across different timelines but share similarities in the nature of the event and the properties of the location. In this work, we improve the use of recurrent neural networks (RNN), particularly long short-term memory (LSTM) networks, to enable AI solutions that generate better time series predictions for LDT. We used similarity measures between the time series based on the time series properties detected by the LSTM and introduced embeddings representing these properties. The embeddings represent the properties of the event, which, coupled with the LSTM structure, can be clustered to identify similar temporally unaligned events. In this study, we explore methods of seeding a multivariate LSTM from time-invariant data related to the geophysical and demographic phenomena modeled by the LSTM. We applied these methods to time-series data derived from COVID-19 detected infection and death cases. We use publicly available socioeconomic data to seed the LSTM models, creating embeddings, to determine whether such seeding improves case predictions. The embeddings produced by these LSTMs are clustered to identify the best-matching candidates for forecasting evolving time series. Applying this method, we showed an improvement in the 10-day moving average predictions of disease propagation at the US County level.

INDEX TERMS Algorithm design and analysis, artificial intelligence, numerical algorithms and problems, statistical methods, time series analysis.

I. INTRODUCTION

In many real-life applications, a dataset consists of instances with features that are both static and dynamic. For example, consider patient health data, such as age and gender, which are relatively static features compared to high-frequency dynamic heartbeat data collected from electrode sensors connected to the patient. Sequence classification models such as recurrent neural networks (RNN) [1], long short-term memory (LSTM) [2], or hidden Markov models (HMM) [3] can be used to model the dynamic time-variant features of an event but are not suitable for addressing static features [4]. In the patient health data example, an LSTM structure can be used to model the heartbeat time-series data across multiple patients; however, it is not suitable for processing static and dynamic data simultaneously [4]. Ensemble methods, such as those provided by Dietterich in [5] and Bagnall et al. [6] provide another way to address this issue: predictions made by temporal models such as dynamic time warping (DTW) [7], rotation forests, [8] and COTE [9] on dynamic data are combined with the predictions of a discriminative classifier on static data by performing distance measures, as presented in [10]. Tzirakis et al. in [11] develop a methodology accomplishing simultaneously: (1) hierarchical clustering of raw

The associate editor coordinating the review of this manuscript and approving it for publication was Chun-Wei Tsai.

dynamic data, (2) learning of deep end-to-end representations, and (3) temporal segments boundaries identification. They computed the similarity between time-series segments using an extension of DTW. A global loss function was used to optimize all three objectives. Although this method results in representations learned from the clusters detected in this process, it does not intrinsically tie these representations to each time series.

In this study, we introduce the concept of a loosely decoupled time-series (LDT) phenomenon and improve LSTM networks to enable artificial intelligence (AI) solutions to offer better time-series predictions informed by static features or features that vary at a different frequency than the main event being modeled. A key feature of LSTMs is that they maintain a dual-purpose internal state (memory) that can aid in the learning and forecasting processes [2]. We used similarity measures between the time series based on the time series properties detected by the LSTM and introduced embeddings [12] representing these properties. The embeddings are constructed from features that are either static or change at a different frequency than the time series of the main event being modeled. We applied this method to improve the predictions of COVID-19 infections and deaths. We treat COVID-19 detected infections and death cases [13] as the main time-variant dynamic event and used socioeconomic data at the US-County level as static features to inform predictions among counties with similar socio-economic structures but differing time lags in COVID-19 disease propagation among their populations.

This study develops ideas from disparate sources, such as COVID-19 forecasting, signature verification, and useful-life estimation from sensor data. Li et al. [14] demonstrated improved signature verification by casting signatures as static representations of dynamic pseudo-processes, using a dynamic process to generate an attention mechanism for the static representation. This has obvious ties with the COVID-19 pandemic as a dynamic process. We chose the geospatial and demographic characteristics of communities as our static representation for two reasons: latent handling of mobility-impacted disease transmission and data augmentation. COVID-19 spread is heavily impacted by population mobility, and Panagopoulos et al. [15] attempted to capture directly using graph neural networks, with vertices as cities and edges as movement between cities, whereas Xiao et al. [16] used intra-city mobility patterns to train an adversarial encoder framework to predict next-at-risk communities. Both groups suffered from a lack of training data, and Panagopoulos et al. [15] attempted to alleviate this using transfer learning between graphs generated from different countries. Wang et al. [17] attempted to use augmented data for training using an ABM to generate synthetic data based on an SEIR epidemic model. However, in our experience (unpublished), the SEIR model is not a good description of COVID-19, and the efficacy of epidemic models is highly dependent on their internal social-interaction model and estimated parameter values. Therefore, we propose a clustering approach based on geospatial and demographic attributes to augment our training data with other US counties that are similar in latent space and known-pandemic trajectory (matching the COVID-19 spread based on where each county is in their respective pandemic trajectory).

Our novel contributions are as follows:

1) Definition and demonstration of loosely decoupled times using static and trajectory-matched dynamic features for improved spatiotemporal prediction.
2) Computationally simple (K-means or K-medoids) latent-space clustering of static geospatial and demographic features accounting for mobility patterns and socioeconomic behaviors.
3) Built-in data augmentation through clustering of data with trajectory-matched pandemic behavior, effectively increasing the training data by reducing the number of prediction classes, thus obviating the need for potentially problematic synthetic data augmentation.

II. LOOSELY DECOUPLED TIMESERIES

We define a loosely decoupled time series (LDT) phenomenon as the relationship between two or more events that could occur at the same place or at different places but across different timelines, sharing similarities in the nature of the event and the properties of the location. We contrast the LDT with event-coupled [18] and tightly coupled time series [19]. Event-coupled time series consist of phenomena starting at the same time, whereas LDT allows for a lag between event onsets. Tightly coupled time series start at the same time and are coupled in time throughout the event, such as the case of audio or speech and the corresponding video of lip gestures, whereas LDT events can occur at varying time frequencies, such as the loose coupling of birth rates measured annually, and unemployment measured monthly. Other examples of LDT include the time series associated with a news cycle (hourly) in relation to the time series associated with the spread of violence (daily or weekly victim counts) or the spread of disease (daily infection or death counts) being covered by the news cycle. The LDT can also span two or more events occurring at different locations.

We represent LDT as:

$$\sim [x\,(t, env)\,, x\,(a_1 * t + b_1, env_1)\,,\\ \times\, x\,(a_2 * t + b_2, env_2)\,, \ldots] \quad (1)$$

where $x\,(t, env_i)$ is a time sequenced event $[0, \ldots, T]$ conditioned on the environment env_i, while a_i provide for a varied frequency time series and b_i provide for a time lag between the two events.

III. OUR APPROACH

A key feature of LSTMs is the maintenance of a dual-purpose internal state (memory) that aids in learning and forecasting. This ameliorates the exploding or vanishing gradient problem experienced by RNNs [2] at the expense of a slightly higher memory and computational complexity. This internal state convolves more distant and recent information input, acting as a compression or embedding mechanism for the time series.

We use this internal memory state as an embedded representation of the time series after appropriately training an LSTM model on the subsequences of the time series.

We represent the LSTM model trained on timeseries $x\left(t, env_i\right)$, as [2]:

$$y^{out_j}\left(t\right) = f_{out_j}\left(net_{out_j}\left(t\right)\right),$$
$$y^{in_j}\left(t\right) = f_{in_j}\left(x_j\left(t, env_i\right)\right)$$

where

$$net_{out_j}\left(t\right) = \sum_u w_{out_j u} y^u \left(t-1\right)$$

And

$$net_{in_j}\left(t\right) = \sum_u w_{in_j u} y^u \left(t-1\right)$$

We also have

$$net_{c_j}\left(t\right) = \sum_u w_{c_j u} y^u \left(t-1\right)$$

Which produces a trained LSTM model represented as:

$$L(T, env_i) \tag{2}$$

The summation index u, based on [2], can represent input units, gate units, memory cells, or even conventional hidden units. These different types of units convey useful information regarding the current state of the LSTM. These may also be recurrent self-connections such as $w_{c_j c_j}$. At time t, c_j's output $y^{c_j}(t)$ of cj is computed as follows:

$$y^{c_j}\left(t\right) = y^{out_j}\left(t\right) h(s_{c_j}\left(t\right))$$

The "internal state" or embedding representation $s_{c_j}(t)$ is:

$$s_{c_j}\left(0\right) = 0, s_{c_j}\left(t\right) = s_{c_j}\left(t-1\right) + y^{in_j}\left(t\right) g\left(net_{c_j}\left(t\right)\right) fort > 0$$

The differentiable function g "squashes" net_{c_j}; the differentiable function h scales the memory cell outputs computed from the internal state s_{c_j}.

We represent the embedding representation of $L(T, env_i)$ as:

$$s_c\left(T, env_i\right) \tag{3}$$

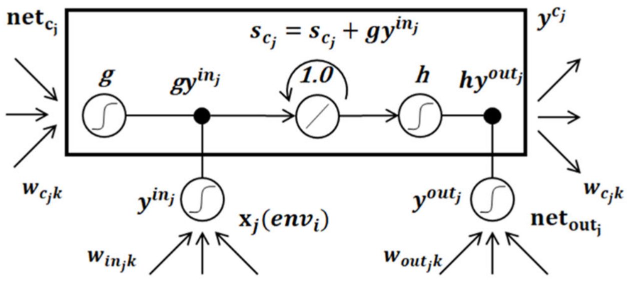


FIGURE 1. Architecture of memory cell c_j (the box), the j-th memory cell block, and its gate units in_j, and out_j. The self-recurrent connection (with weight 1.0) indicates feedback with a delay of 1 time step. The index k ranges over hidden units u [2].

We enforce the same cell size u for all LSTMs trained to ensure the equidimensional representation of the internal states (3) to facilitate comparison independent of their length. Our general focus is on using the equidimensional embedding representations of (3) along with the time invariant LSTM-based prediction: given an evolving timeseries $x\left(t, env_i\right)$, search for and select longer/more evolved timeseries $x\left(a_j * t + b_j, env_j\right)$ that can be associated to produce a better prediction for the next time step(s) of (2) for $x\left(t, env_i\right)$.

To locate the associated time series, we expand the internal state (3) of an LSTM to include the static properties of env_i:

$$p_i(env_i) \tag{4}$$

We combine (3) and (4) to form:

$$s_c\left(T, env_i\right), p_i(env_i)] \tag{5}$$

We call (5) the embedding representation of the phenomena at env_i. We then adopt a clustering method to cluster the embedding representations of several phenomena to identify LDT tuples. In this study, two clustering methods were explored. K-means:

$$C_1, C_2, \ldots C_k = \arg min \sum_{i=1}^{k} \sum_{x \in S_i} \|x - C_i\|^2 \tag{6}$$

With k centroid points $C_k$and minimizing the sum over each cluster of the sum of the square of the distance between the point and its centroid, the centroid is not necessarily a point from the data set.

And K-medoids:

$$M_1, M_2, \ldots M_k = \arg min \sum_{i=1}^{k} \sum_{x \in S_i} \|x - M_i\|^2 \tag{7}$$

which is similar to K-means but uses medoids M_k chosen from points in the dataset.

Once the LDT clustered tuples are identified, we seed the main LSTM model based on this static data prior to each training episode as well as prior to making a forecast. We refer to this seeding as local static embedding. This seeding takes the form of a dense embedding layer from the static data vector to the (initialization of) hidden layer in the LSTM model. This dense embedding was simultaneously trained by backpropagation with the LSTM training regimen. The embedding layer is reused at the estimation time to reseed the hidden layer of the LSTM at the initial step.

IV. FORECASTING COVID-19

Coronavirus disease (COVID-19) is a disease (SARS-CoV-2) which spread rapidly worldwide. Various metrics have been proposed for monitoring the spread and evolution. Although there is significant variation in the data collected, two main metrics of interest are publicly available. One is the number of people "infected" (those that tested positive on a standardized testing platform), and the other is the number of people who died due to a recorded affiliation with the virus.

However, both metrics were subject to discrepancies. While some districts only report PCR-positive tests as the gold standard, others report results from less-reliable methods or even antibody tests (antibody tests rarely show positivity until the end of an infection and then show positivity for months or more after viral clearance). Additionally, death counts often represent both deaths due directly to COVID-19 complications and deaths due to unrelated causes, but with positive test results (from preventative screening).

Simultaneously, there is a need from the public (and public health officials) to predict the evolution of these metrics days, potentially weeks ahead, for resource allocation and policy formulation. This has prompted numerous efforts to develop and apply predictive models [20], [21].

The standard models used in public health are the derivatives of the SIR model [22]. These models are based on the "evolution" of an individual through the stages of a disease, from Susceptible "S" (has potential to get infected) to Infected "I" (virus is present) to Recovered "R" (disease ran its course). Variations considering asymptomatic or unreported infections as well as death as an outcome were also used. While well understood, both from a theoretical and practical (estimation) perspective, these models are necessarily limited by the assumption of compartmentalization (disease evolving in isolation). Human movement patterns lead to the diffusion of infections across boundaries. Solving coupled SIR compartmental models subject to constraints and diffusion is significantly more difficult and potentially intractable without deeper (longer history) samples.

Another aspect of compartmental models is their focus on inference, rather than prediction. The primary focus of SIR-type models is to estimate disease characteristics (e.g., transmission rate) rather than prediction. In addition, predictions are only helpful up to a certain point. Just knowing what will happen is of limited usefulness in the absence of scenario-based alternatives. From a prescriptive (predictive + actionable) around COVID-19, it would be helpful to build upon similarities and "local tests" between US counties. By local tests, we mean different restrictions and implementation or adherence to these restrictions and their impact on the disease trajectory.

Working with researchers at the Air Force Research Lab/Autonomous Capabilities Team (ACT3), we applied the above-described methods of LDT-enhanced LSTM modeling to COVID-19 detected infection and death cases [13]. The time interval under study was from March to October 2020. During this time, the global events were such that there was a limited supply of COVID-19 testing resources, hesitation in applying and adopting non-pharmaceutical interventions (NPIs), and several US counties adopting lockdown procedures. During the same period, a single dominant variant of COVID-19 was identified. From this perspective, disease evolution was not influenced by mixed variants as it became more common in subsequent months. We limited the geographic span and resolution of our study to the US county level. County-level data are more likely to contain a systematic definition of a COVID-19 case following locally consistent testing approaches and capabilities, consistent NPI measures, and consistent lockdown directives (if any were applied). More importantly, counties in the same state may experience a lag in the disease spread. Therefore, known data about the spread of the disease in counties already affected can inform the future state of counties that are starting to experience their first cases. Similarly, the effect of NPIs (e.g., mask/lock-down mandates) observed in the infection and death time series of some counties can inform the expected effect from similar NPIs in counties that are considering such measures. Thus, the value proposition of modeling county-level data is significant from an operational and NPI implementation point of view.

A. DESCRIPTION OF DATA

The 2010 census demographic datasets [23] for each US county were used in the analysis. Table 1 presents the data fields used in this study.

TABLE 1. 2010 Census demographic [23] data fields used in analysis.

Data Field	Description
SUMLEV	Geographic Summary Level
STATE	State FIPS code
COUNTY	County FIPS code
STNAME	State Name
CTYNAME	County Name
YEAR	Year
AGEGRP	Age group
TOT_POP	Total population
TOT_MALE	Total male population
TOT_FEMALE	Total female population
WA_MALE	White alone male population
WA_FEMALE	White alone female population
BA_MALE	Black or African American alone male population
BA_FEMALE	Black or African American alone female population
AA_MALE	Asian alone male population
AA_FEMALE	Asian alone female population
TOM_MALE	Two or More Races male population
TOM_FEMALE	Two or More Races female population

TABLE 2. USDA economic research service [24] data fields used in analysis.

Data Field	Description
FIPS_Code	State-county FIPS code
State	State abbreviation
Med_HH_Income_Percent_of_State_Total_2019	County household median income as a percent of the State total median household income, 2019
Median_Household_Income_2019	Estimate of median household Income, 2019
Unemployment_rate_2019	Unemployment rate, 2019
Unemployment_rate_2020	Unemployment rate, 2020

The economic data for each county were sourced from the USDA Economic Research Service, [24] and Table 2 lists the data fields that were used in this analysis.

In total, 385 socioeconomic features from 3142 US counties were used in the analysis. These constituted the static feature set and were assumed to remain constant during the analysis period.

COVID-19 infection metrics are aggregated into various entities. Johns Hopkins University [25] is an early and continuing resource for such data. However, they only collate what is reported by local health authorities, which are subject to local delays and constraints in identifying and reporting disease spread.

For example, it has been observed that the reported counts exhibit a periodic dip around weekends. This is simply due to the limitations of scheduled activities for labs running these tests. Correspondingly, there is a ''bump'' in counts at the beginning of the week, usually on Mondays.

The COVID-19 datasets [13], [25] were used to analyze the daily number of infections and deaths. Table 3 lists the data fields used in the analysis.

The number of cumulative infections and deaths was normalized against the county population data. Raw data from [13] were used as is, with the exception of days in which a drop in cumulative infections or cumulative deaths were reported, and the last reported value before the drop was used for all subsequent days until the cumulative values reached that level again.

TABLE 3. **COVID-19 [13], [25] data fields used in the analysis.**

Data Field	Description
FIPS	State-county FIPS code
Date	(Implicit from file name)
Confirmed	Infections confirmed in area
Deaths	Deaths in area

B. LSTM TRAINING REGIMEN

Focusing on prediction rather than inference potentially increases the utility of models from other domains. Time series analysis is one such domain; however, the structural constraints on these models are not easily aligned with the expectations of disease evolution. An ideal model would ''remember'' trends and changes over varying time horizons (e.g., the convexity of the infected cases' trend changed N days ago, where N could vary with the region under consideration). LSTM models have an established history in natural language processing, where learning the relationships between potentially distant words helps predict the next word. This is predicated on the underlying structure of the language from which the samples are drawn (e.g., English), with long-term memory keeping track of words earlier in the input. It is this long-term memory we had in mind when testing LSTM as a solution for predicting COVID-19 ''trajectories''. Given sufficient data, the model should learn to distinguish the accelerating spread regions of the timeline from more linear or saturated growth regions.

In our setup, the LSTM layer is followed by a dense layer with an output that is dependent on the predicted variables. The variables are infected, and dead counts which are normalized by the population of the county.

LSTM training modules from PyTorch [26] were used, and ray tune [27] was used to perform hyperparameter tuning using a grid search. LSTM models were constructed to allow for a grid search across 64, 128, 256, and 512 hidden memory cells and across one, two, or three network layers. Several loss functions were tested for training: mean square error (MSE), relative mean square error (RMSE), indexed or scaled versions to account for the changing variability in the inputs across time, and versions penalizing for non-monotonic output. Input tensors covering a period of 7–30 days were used. The output was compared to actual values on 1-, 3- and 5-day sliding intervals (day offset). The models with the highest prediction accuracy (lowest loss vs. desired output) were successively retained by Ray Tune within the allocated time/computing resources; these models learned the most accurate representation for that county and point in time (PIT). More than six hundred models were trained to extract county level embeddings over time.

For the purpose of consolidating our results, we consider two loss functions:

- MSE (abs): absolute mean square error between the output and expected values
- RMSE (rel): the relative difference between the output and expected values, with a large penalty imposed for producing nonmonotonic sequences. A small quantity (10^{-8}) was added to the denominator to avoid division by zero.

The expectation was that RMSE-based models would more closely match the disease trends, especially in the earlier stages when their population-normalized values are exceedingly small.

The following setup was used for all experiments:

- data for all counties cover the interval from the first recorded case (for each US county) through 09/18/2020
- there is always a ''buffer'' of the last 30 days which are not ''seen'' by the trained models (test_days = 30); this means 09/19 through 10/18 is reserved for testing / evaluation of the model
- the models are trained for a certain time/computation ''budget'' using Ray Tune's ASHA Scheduler
- individual counties' training epochs, consisting of one pass through all (chunked) historical data.
- Mini-batch training was used for all models (three batches for individual county models)

C. LSTM HIDDEN STATES AS EMBEDDINGS

For each US county, the LSTM training regimen produces an optimum characteristic LSTM model for predicting the number of infections and deaths for 1, 3, or 5 days in advance. The optimum characteristic LSTM consisted of three layers with a hidden state made of 256 cells. The hidden state of each optimally trained LSTM model, represented as $L(T, env_i)$, with the hidden state represented as $s_c\,(T, env_i)$ for County, 'i', was used to represent the US County. The hidden state was a vector of 256 dimensions representing COVID-19 embedding for each US county, $[s_c\,(T, env_i)\,, p_i(env_i)]$.

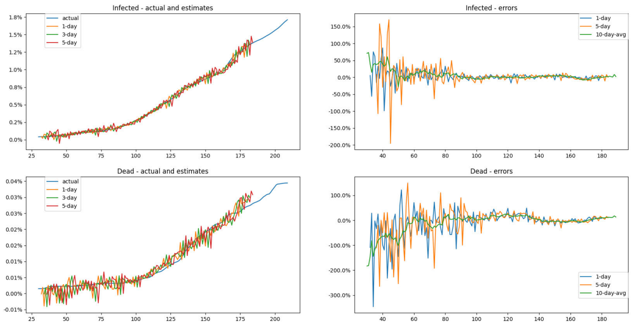


FIGURE 2. COVID-19 infection and dead 10-day-moving-average percent of population predictions contrasted against actual numbers for one US County. Predictions for 1, 3, and 5-forward looking days are provided on the left, and corresponding error rates (RMSE) are provided on the right. The x-axis in all plots represents days. The plots on the right show that, for this particular county, 90 and 120 days or more need to have passed from the onset of the disease in that county for the infection and death prediction error rates respectively to stabilize.

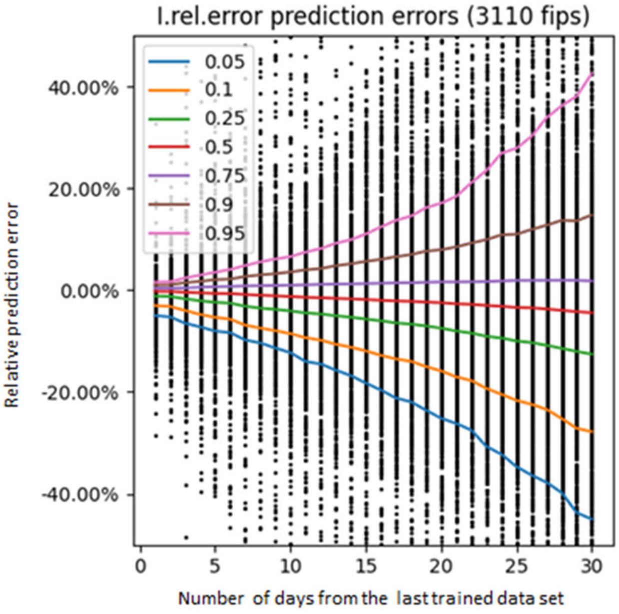


FIGURE 3. RMSE error plots for one US county showing COVID-19 infection predictions for 1-30 days into the future. 5-day forward looking predictions are +/-3% accurate, and 10-day forward looking prediction errors are around +/-10% accurate.

The US County embeddings were clustered using the two clustering methods in (6) and (7) with k = 3 clusters at 30, 60, and 90 days in time. The choice of these cluster reflects the practical decision-making time horizons that were being applied during the COVID pandemic and allow decision makers ample time to make informed decisions based on how the disease spread in similar counties. Clusters identified counties with similarity based on a shorter history, matched other counties with a cluster, and then analyzed the evolution of the groups' timeline to inform the new county's evolution.

V. ANALYSIS OF RESULTS

The results presented in this section focus on analyzing 17 US counties based on the state of Ohio, with over six hundred embeddings based on various points in time (PIT) of the trained LSTM models for every county. The following notation is used in this section to represent the data analyzed:

- Data are a vector of observed values over time and can be obtained with or without socioeconomic data.
- The clustering method is either (6) or (7), with (7) represented with the label "k-means" and (7) as "k-medoids" in the plots.
- Plots were taken at a point-in-time (PIT) reflected in the label of a plot and for a predefined set of clusters identified as cl:n, where n is the number of clusters in the plot.
- Clustering involves all hidden layer states ("all") or only the last one ("last").

Figure 4 provides an example of k-means clustering into three clusters of 17 trained LSTM models with 60 days of training data using actual COVID-19 infection counts, relative to the total county population.

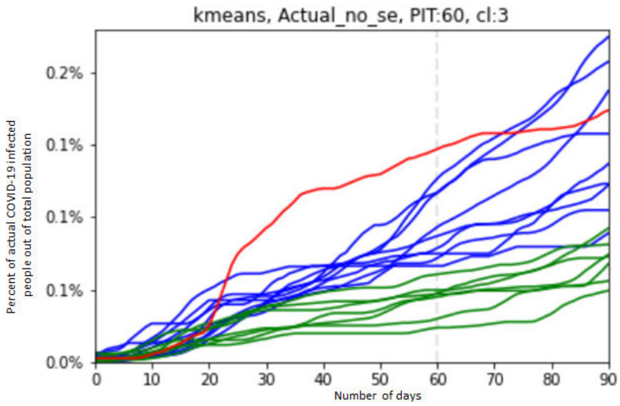


FIGURE 4. Example plot for 17 counties in OH clustered using k-means and the actual COVID-19 infections out of total population count. Socio-economic data was not factored in the clustering. Clustering was done using the hidden neural network layers (state) of an LSTM model trained with 60 days of data.

We analyzed the alignment (concordance) of the two clustering methods (6) and (7) using two metrics:

1) Accuracy based on a confusion matrix
   a) Calculated as the percent of the diagonal values present in overall confusion matrix
   b) Dependent on the cluster order (based on a specific permutation of clusters)
   c) Has values ranging from 0 to 1, with one being more accurate
   d) Represented as "Acc" in plots
2) Adjusted Rand Index
   a) Considers the random chance "alignment" of clustering methods
   b) Independent of cluster order
   c) Has values ranging from -1 to 1, with one being more accurate.
   d) Represented as "ARI" in plots

The following figures show the results of the analysis.

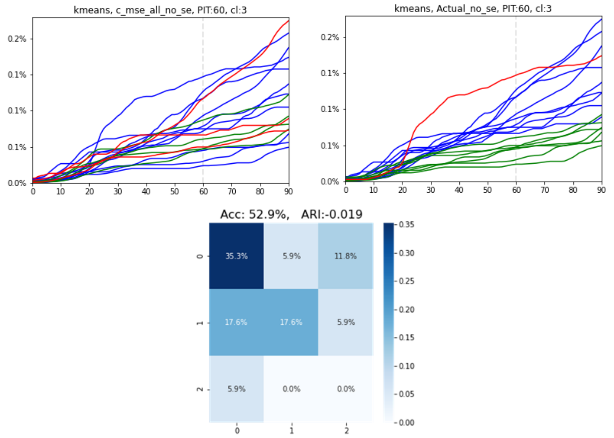


FIGURE 5. K-means with 3 clusters used to analyze PIT=60 with no socio-economic data. Top left shows clusters based on LSTM embeddings using all layers. Top right shows clusters based on actual values only. The bottom center shows the confusion matrix and ARI between the top two graphs.

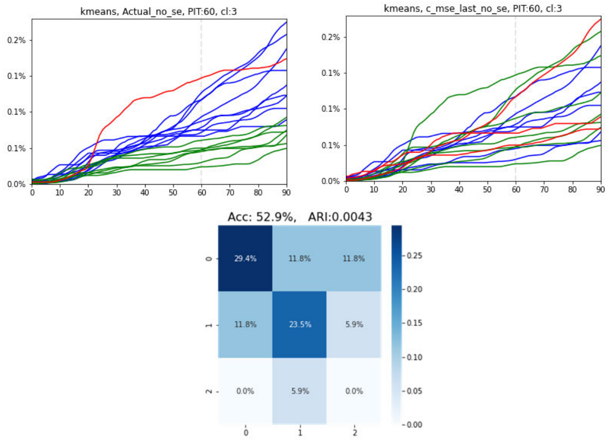


FIGURE 6. K-means with 3 clusters used to analyze PIT=60 with no socio-economic data. Top left shows clusters based on actual values only. Top right shows clusters based on LSTM embeddings using the last layer only. The bottom center shows the confusion matrix and ARI between the top two graphs.

From Figure 5, Figure 6, Figure 7, and Figure 8 it can be concluded that including socio-economic factors increases the consistency (overlap) between actual, observation-based, and embedding-based clustering. It is also noticeable that K-means (6) has an advantage over K-medoids and that using the last hidden layer of the LSTM as the embedding vector yields better results than using all layers. The optimum clustering approach for COVID-19 data is to include socio-economic data and use the last layer of the LSTM as the embedding in a k-means clustering algorithm.

We then analyzed the change in clustering over time. We define a "cluster stability" metric as follows (using Figure 9 as the example):

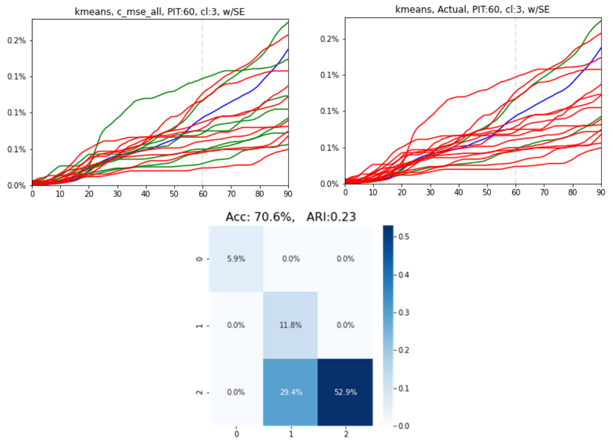


FIGURE 7. K-means with 3 clusters used to analyze PIT=60 with socio-economic data. Top left shows clusters based on LSTM embeddings using all layers. Top right shows clusters based on actual values only. The bottom center shows the confusion matrix and ARI between the top two graphs.

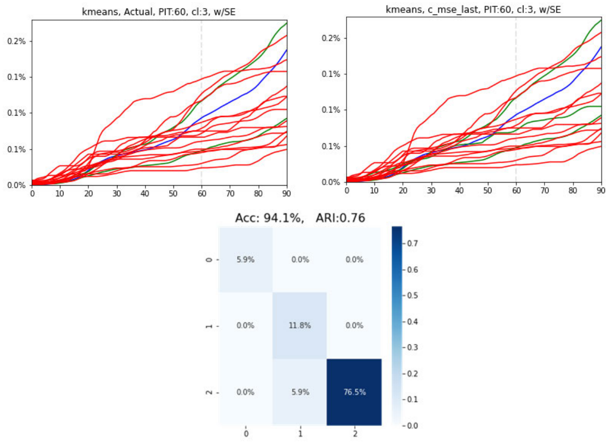


FIGURE 8. K-means with 3 clusters used to analyze PIT=60 with socio-economic data. Top left shows clusters based on actual values only. Top right shows clusters based on LSTM embeddings using the last layer only. The bottom center shows the confusion matrix and ARI between the top two graphs.

- Cluster 0 (green dots) had initially ten members
  - Of these, 3 shifted to cluster 1 and 3 shifted to cluster 2
- Cluster 1 (orange dots) had initially 4 members
  - Of these, 2 shifted to cluster 0 and 1 shifted to cluster 2
- Cluster 2 (blue dots) had initially 3 members
  - Of these, 1 shifted to cluster 0 and 1 shifted to cluster 1
- The overall cluster stability metric is defined as the maximum accuracy (for optimal cluster reordering, yielding the maximum diagonal).
  - In this case: $(4+1+1)/17 = 35.3\%$

Table 4 shows the cluster stability calculations (last two columns) across two types of embeddings: without socioeconomic data and with socioeconomic data. The two columns

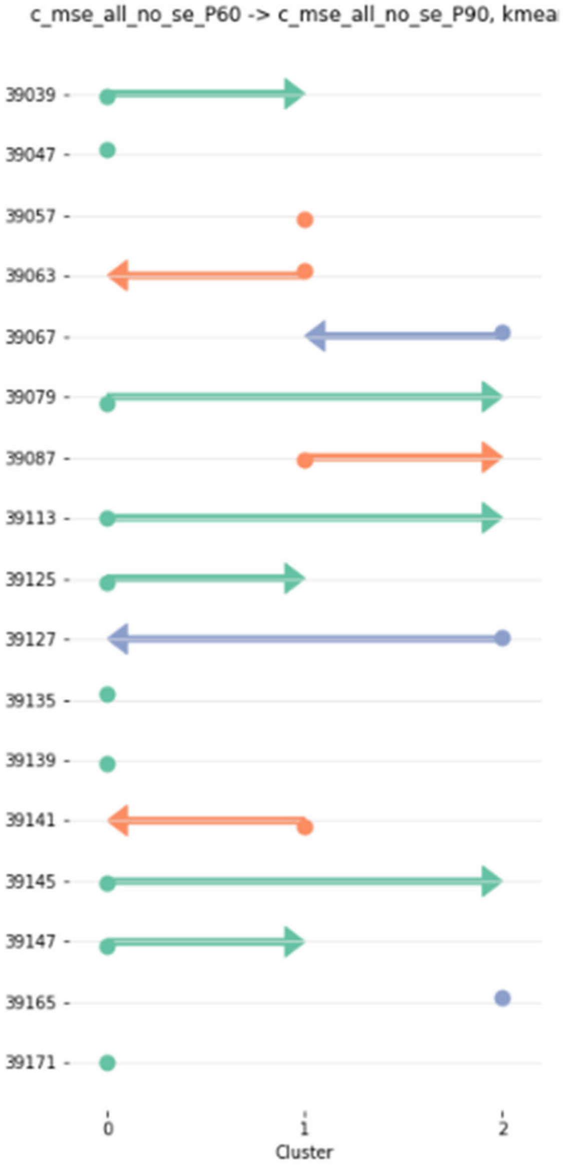
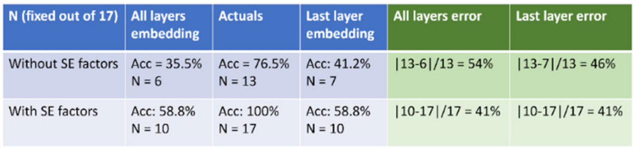


FIGURE 9. Using K-means and 3 clusters for the 17 counties in OH with no socio-economic data, this plot shows how the 17 counties changed cluster assignment between PIT=60 and PIT=90.

TABLE 4. Cluster stability over time (SE: Socio-Economic).



N (fixed out of 17)	All layers embedding	Actuals	Last layer embedding	All layers error	Last layer error
Without SE factors	Acc = 35.5% N = 6	Acc = 76.5% N = 13	Acc: 41.2% N = 7	|13-6|/13 = 54%	|13-7|/13 = 46%
With SE factors	Acc: 58.8% N = 10	Acc: 100% N = 17	Acc: 58.8% N = 10	|10-17|/17 = 41%	|10-17|/17 = 41%

reflect the calculations performed using LSTM embeddings derived from the last layer or all layers of the neural network.

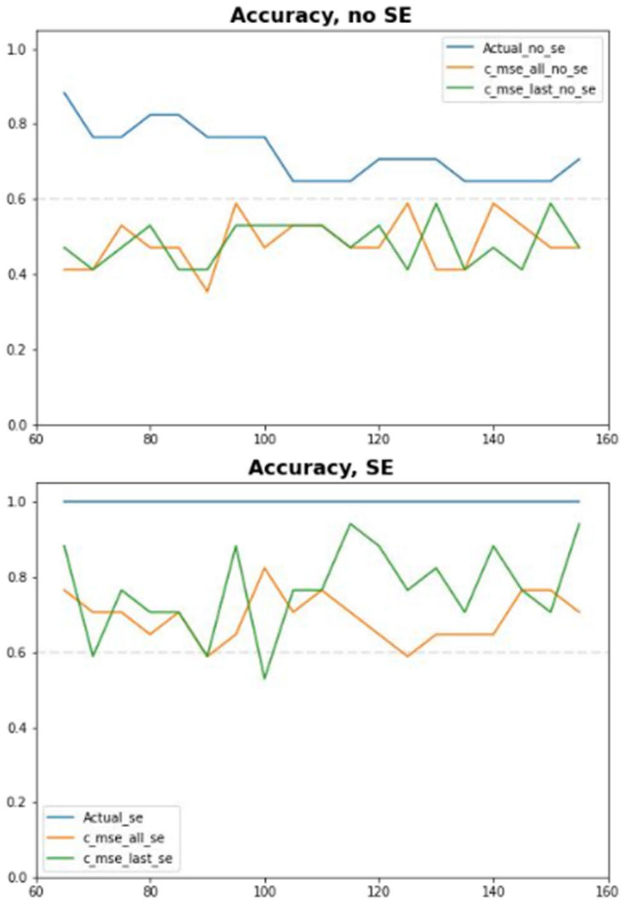


FIGURE 10. Cluster Stability plotted over time in comparison with PIT=60 for the cases of (top) without socio-economic data and (bottom) with socio-economic data.

Figure 10 plots the cluster stability metrics over a wide range of PIT variations and compares the effect of socio-economic data on the stability of the clusters. Adding the socio-economic data to the LSTM embeddings stabilizes the cluster formation over time.

VI. CONCLUSION

The work presented here demonstrates improved LSTM forecasting through embeddings derived from a loosely decoupled time series. We applied this methodology to COVID-19 infections and deaths at the US county level. Our socioeconomic embedding approach demonstrates enhanced 10-day moving average predictions compared to traditional LSTM modeling, especially in conjunction with K-means clustering of the final layer embeddings. Additionally, we demonstrate stability in the clustering of LDTs when combined with socioeconomic data, providing increased consistency in predictions.

With this approach, US counties that lag behind in catching the virus benefit from counties similar in socioeconomic demographics, but with an earlier start to their disease propagation, improving the predictive outcome.

REFERENCES

[1] M. I. Jordan, "Serial order: A parallel distributed processing approach," *Adv. Psychol.*, vol. 121, pp. 471–495, May 1997.

[2] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[3] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[4] A. Leontjeva and I. Kuzovkin, "Combining static and dynamic features for multivariate sequence classification," in *Proc. DSAA*, Oct. 2016, pp. 21–30.

[5] T. G. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Systems*. Berlin, Germany: Springer, 2000.

[6] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, "The great time series classification bake off: A review and experimental evaluation of recent algorithmic advances," *Data Mining Knowl. Discovery*, vol. 31, no. 3, pp. 606–660, May 2017.

[7] M. Brown and L. Rabiner, "Dynamic time warping for isolated word recognition based on ordered graph searching techniques," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Paris, France, May 1982, pp. 1255–1258.

[8] J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso, "Rotation forest: A new classifier ensemble method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1619–1630, Oct. 2006.

[9] A. Bagnall, J. Lines, J. Hills, and A. Bostrom, "Time-series classification with COTE: The collective of transformation-based ensembles," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 9, pp. 2522–2535, Sep. 2015.

[10] J. Paparrizos, C. Liu, A. J. Elmore, and M. J. Franklin, "Debunking four long-standing misconceptions of time-series distance measures," in *Proc. SIGMOD*, Jun. 2020, pp. 14–19.

[11] P. Tzirakis, M. A. Nicolaou, B. Schuller, and S. Zafeiriou, "Time-series clustering with jointly learning deep representations, clusters and temporal boundaries," in *Proc. 14th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, May 2019, pp. 1–5.

[12] Google. *Machine Learning Crash Course*. Accessed: Dec. 30, 2021. [Online]. Available: https://developers.google.com/machine-learning/crash-course/embeddings/video-lecture

[13] E. Dong, H. Du, and L. Gardner, "An interactive web-based dashboard to track COVID-19 in real time," *Lancet Infectious Diseases*, vol. 20, no. 5, pp. 533–534, doi: 10.1016/S1473-3099(20)30120-1.

[14] H. Li, P. Wei, and P. Hu, "Static-dynamic interaction networks for offline signature verification," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 1–9.

[15] G. Panagopoulos, G. Nikolentzos, and M. Vazirgiannis, "Transfer graph neural networks for pandemic forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 1–8.

[16] C. Xiao, J. Zhou, J. Huang, A. Zhuo, J. Liu, H. Xiong, and D. Dou, "C-Watcher: A framework for early detection of high-risk neighborhoods ahead of COVID-19 outbreak," in *Proc. 35th AAAI Conf. Artif. Intell. (AAAI)*, 2021, pp. 1–9.

[17] L. Wang, J. Chen, and M. Marathe, "DEFSI: Deep learning based epidemic forecasting with synthetic information," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 1–10.

[18] T. T. Kristjansson, B. J. Frey, and T. S. Huang, "Event-coupled hidden Markov models," in *Proc. IEEE Int. Conf. Multimedia Expo. ICME Latest Adv. Fast Changing World Multimedia*, Aug. 2000, pp. 385–388.

[19] M. I. Jordan, Z. Ghahramani, and L. K. Saul, "Hidden Markov decision trees," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 9, 1997, pp. 1–7.

[20] S. M. Shakeel, N. S. Kumar, P. P. Madalli, R. Srinivasaiah, and D. R. Swamy, "COVID-19 prediction models: A systematic literature review," *Osong Public Health Res. Perspect.*, vol. 12, no. 4, pp. 215–229, Dec. 2021.

[21] J. C. Clement, V. Ponnusamy, K. C. Sriharipriya, and R. Nandakumar, "A survey on mathematical, machine learning and deep learning models for COVID-19 transmission and diagnosis," *IEEE Rev. Biomed. Eng.*, vol. 15, pp. 325–340, 2022.

[22] W. Kermack and A. McKendrick, "A contribution to the mathematical theory of epidemics," *Proc. Royal Soc. Math., Phhysical Engineeering Sci.*, vol. 115, no. 772, pp. 1–22, Aug. 1927.

[23] UG Census. (2010). *US Census Datasets*. Accessed: Nov. 15, 2020. [Online]. Available: https://www2.census.gov/programs-surveys/popest/datasets/2010-2019/counties/asrh/cc-est2019-alldata.csv

[24] USDA. (Jun. 2, 2021). *USDA Economic Research Service*. [Online]. Available: https://www.ers.usda.gov/data-products/county-level-data-sets/download-data/

[25] E. Dong, H. Du, and L. Gardner, "An interactive web-based dashboard to track COVID-19 in real time," *Lancet Infectious Diseases*, vol. 20, no. 5, pp. 533–534, 2020.

[26] PyTorch. *PyTorch Machine Learning Framework*. Accessed: Jun. 26, 2023. [Online]. Available: https://pytorch.org/

[27] Ray. *Ray Tune: Scalable Hyperparameter Tuning*. Accessed: Jun. 26, 2023. [Online]. Available: https://docs.ray.io/en/latest/tune/index.html

CHRISTIAN G. MANASSEH received the M.Eng. degree in information technology from the Massachusetts Institute of Technology, Boston, MA, USA, in 1999, and the Ph.D. degree in systems engineering from the University of California at Berkeley, Berkeley, CA, USA, in 2010. He is currently the Founder of Mobius Logic Inc., a company that specializes in artificial intelligence and data science research and development with several grants from state and federal agencies. He has coauthored multiple research papers on the digitization of human behavior and artificial intelligence to assist human decision-making. His research interests include artificial general intelligence algorithms and systems and their use in human behavior-focused agent-based modeling simulations.

RAZVAN VELICHE received the B.S. in mathematics from Bucharest University, and the Ph.D degree in mathematics from Purdue University, with a focus on Algebraic Geometry. His publications touch upon graph theory, healthcare, and artificial intelligence. At the time of this research work, he was the Director of Data Science at Mobius Logic, working on Data Science projects and applications in Synthetic Populations, Agent Based Modeling, and Reinforcement Learning. He is particularly interested in data modeling, knowledge graphs, and knowledge representation, but maintains a strong interest in reinforcement learning and agent-based modeling.

JARED BENNETT received the B.S. degree in physics from The Ohio State University, in 2014, with a focus on single-molecule fluorescent microscopy, and the Ph.D. degree in biophysics from the University of California at Berkeley, Berkeley, CA, USA, in 2021, with a focus on mathematical modeling of gene drives. He is currently an Associate Engineer with Mobius Logic Inc., expanding the boundary between AGI and open-ended learning. His research interests include large-scale global optimization, disease transmission, and agent-based modeling in the context of social dynamics.

HAMILTON SCOTT CLOUSE received the Ph.D. degree in machine learning and data analysis for developing the theory for and implementing an adaptive, distributed, vision-augmented behavior characterization system. He is currently the Chief AI Officer and the Co-Founder of the Autonomy Capability Team 3 (ACT3), a team of more than 200 members dedicated to developing and fielding novel AI technologies. For decades, he was with partners in industry, academia, and both public and private research to provide innovative and state-of-the-art automation and data science solutions for a host of complex challenges. Fervor for research in artificial intelligence and data science drives him to stay at the cutting edge of these fields through frequent presentations, publications, and participation in the global research community. Consequently, he has led the research team to win several international academic competitions in natural language processing, emergent multi-agent cooperation, and scene understanding. Such performance coupled with continued grant awards and an ever-growing cadre of students has earned him a fellowship for young investigators (the Air Force Early Career Award) and a nomination for the same from the President of the United States (PECASE).