TDEFSI: Theory-guided Deep Learning-based Epidemic Forecasting with Synthetic Information

LIJING WANG, Network Systems Science and Advanced Computing Division, Biocomplexity Institute and Initiative & Department of Computer Science, University of Virginia
JIANGZHUO CHEN, Network Systems Science and Advanced Computing Division, Biocomplexity Institute and Initiative, University of Virginia
MADHAV MARATHE, Network Systems Science and Advanced Computing Division, Biocomplexity Institute and Initiative & Department of Computer Science, University of Virginia

Influenza-like illness (ILI) places a heavy social and economic burden on our society. Traditionally, ILI surveillance data are updated weekly and provided at a spatially coarse resolution. Producing timely and reliable high-resolution spatiotemporal forecasts for ILI is crucial for local preparedness and optimal interventions. We present *T*heory-guided *D*eep Learning-based *E*pidemic *F*orecasting with *S*ynthetic *I*nformation (TDEFSI),[1] an epidemic forecasting framework that integrates the strengths of deep neural networks and high-resolution simulations of epidemic processes over networks. TDEFSI yields accurate high-resolution spatiotemporal forecasts using low-resolution time-series data.

During the training phase, TDEFSI uses high-resolution simulations of epidemics that explicitly model spatial and social heterogeneity inherent in urban regions as one component of training data. We train a two-branch recurrent neural network model to take both within-season and between-season low-resolution observations as features and output high-resolution detailed forecasts. The resulting forecasts are not just driven by observed data but also capture the intricate social, demographic, and geographic attributes of specific urban regions and mathematical theories of disease propagation over networks.

We focus on forecasting the incidence of ILI and evaluate TDEFSI's performance using synthetic and real-world testing datasets at the state and county levels in the USA. The results show that, at the state level, our method achieves comparable/better performance than several state-of-the-art methods. At the county level, TDEFSI outperforms the other methods. The proposed method can be applied to other infectious diseases as well.

CCS Concepts: • **Computing methodologies** → *Causal reasoning and diagnostics*; *Spatial and physical reasoning*; *Reasoning about belief and knowledge*;

Additional Key Words and Phrases: Epidemic forecasting, deep neural network, LSTM, causal model, synthetic information, physical consistency

[1]The preliminary version of this work [87] was presented at the *31st Innovative Applications of Artificial Intelligence Conference (IAAI'19)*.

This work has been partially supported by Defense Threat Reduction Agency (DTRA) Grant HDTRA1-17-D-0023, National Institutes of Health (NIH) Grant 1R01GM109718, NSF BIG DATA Grant IIS-1633028, and NSF DIBBS Grant ACI-1443054.
Authors' addresses: L. Wang, M. Marathe, and J. Chen, Network Systems Science and Advanced Computing Division, Biocomplexity Institute and Initiative & Department of Computer Science, University of Virginia, Charlottesville, VA, 22904; emails: lw8bn@virginia.edu, marathe@virginia.edu, chenj@virginia.edu.

2374-0353/2020/04-ART15 $15.00
https://doi.org/10.1145/3380971

ACM Reference format:
Lijing Wang, Jiangzhuo Chen, and Madhav Marathe. 2020. TDEFSI: Theory-guided Deep Learning-based Epidemic Forecasting with Synthetic Information. *ACM Trans. Spatial Algorithms Syst.* 6, 3, Article 15 (April 2020), 39 pages.
https://doi.org/10.1145/3380971

1 INTRODUCTION

Influenza-like illness (ILI) poses a serious threat to global public health. Worldwide, annually, seasonal influenza causes three to five million cases of severe illness and 290,000 to 650,000 deaths [90]. Since 2010 in the USA, seasonal influenza has resulted in 10–50 million cases annually, 140,000 to 960,000 hospitalizations, and between 12,000 and 79,000 deaths and is responsible for approximately $87.1 billion in economic losses [19, 62]. Producing timely, well-informed, and reliable forecasts for ILI of an ongoing flu epidemic is crucial for preparedness and optimal intervention [27]. Traditionally, ILI surveillance data from the Centers for Disease Control and Prevention (CDC) has been used as reference data to predict future ILI incidence. The surveillance data are updated weekly but often delayed by 1 to 4 weeks and is provided at a HHS region (i.e., the 10 regions defined by the United States Department of Health & Human Services) level and recently at the state level. Considering the heterogeneity between different subregions, accurate predictions with a finer resolution, e.g., at county or city level in the USA, are crucial for local public health decision making, optimal mitigation resource allocation among subregions, and household or individual-level preventive actions informed by neighboring prevalence [100]. Given spatially coarse-grained surveillance data, it is challenging to forecast at a finer spatial level.

In this article, we use *flat-resolution* forecasting to denote the prediction of ILI incidence with the same resolution as the surveillance data and *high-resolution* forecasting to denote the prediction with a higher geographical resolution than provided in surveillance data. We focus on state-level ILI surveillance and state (flat-resolution) or county-level (high-resolution) ILI forecasts. We use the term *deep neural networks (DNN)* to denote multi-layer neural networks with multiple inputs and outputs.

1.1 Our Contributions

We propose a novel epidemic forecasting framework, called *T*heory-guided *D*eep Learning-based *E*pidemic *F*orecasting with *S*ynthetic *I*nformation (*TDEFSI*).

Overall approach. TDEFSI produces accurate weekly high-resolution ILI forecasts from flat-resolution observations. This is achieved by using a two-branch neural network model for ILI forecasting. It combines within-season observations (observed data points of the previous weeks that characterize the ongoing epidemic) and between-season historical observations (observed data points from similar weeks of the past seasons that characterize general trends around the current week). It can generate probabilisitic forecasts by using Monte Carlo Dropout technique [35].

A key contribution of the article is to use theory generated synthetic data to train the neural network. This is necessitated by the fact that disease surveillance data are sparse. Furthermore, the data are noisy and incomplete. We overcome the limitations by training TDEFSI using data generated by high-performance computing-based simulations of well accepted causal processes that capture epidemic dynamics. These simulations are based on decades of work and have been extensively validated. The simulations allow us to (i) use a realistic representation of the underlying social contact network that captures the multi-scale spatial, temporal, and social interactions, as well as the inherent heterogeneity of social networks (individual demographic attributes, heavy tailed nature of social contacts, etc.), leading to *forecasts that are context specific and capture the unique properties of a given urban region*; (ii) produce multi-resolution forecasts even though

observational data might only be available at an aggregate level, leading to *an ability to forecast disease incidence at a county or a city level as well as forecasts for desired demographic groups*; and (iii) capture the underlying causal processes and mathematical theories leading to *explainable and generalizable AI*—the combination of theory and data-driven machine learning is an important and emerging approach to scientific problems that are data sparse.

Key findings. Extensive experiments were carried out using both real-world as well as synthetic datasets for testing. (i) In experiments on synthetic testing data, we evaluate TDEFSI performance with different hyperparameter settings and find that the best look-back window size is 52 weeks, the same as the period of influenza seasons, for both state-level and county-level forcasting. (ii) In experiments on two states of the USA using their real ILI incidence data as ground truth, we compare TDEFSI and its variants with several state-of-the-art forecasting methods, among which four methods can only make state-level predictions directly and one method can make both state-level and county-level predictions directly. The results indicate that in most cases TDEFSI methods achieves comparable/better performance than the comparison methods at the state level. For high-resolution forecasting at the county level, TDEFSI significantly outperforms the comparison methods. Between the variants of TDEFSI, we find that the between-season branch of our neural network model improves the forecasting accuracy. (iii) We also find that the two physical constraints in our TDEFSI model, which address spatial consistency and non-negative consistency respectively, contribute to the improvement on the forecasting performance. (iv) Through a comparison between TDEFSI models trained with datasets generated by no-intervention simulations and those by intervention-aware simulations, we find that in our TDEFSI framework realistic settings in the causal model behind the neural network do improve the generalizability of the trained forecasting model. (v) In general, TDEFSI is able to capture the heterogeneity in epidemic dynamics among counties in a state and the spatial spread of the disease across the counties.

To the best of our knowledge, TDEFSI is the first to use a realistic causal high-resolution model to train a deep neural network for epidemic forecasting. The basic approach is general and points to the potential utility of the approach to study other problems in social and ecological sciences. Unlike physical systems, encoding system-level constraints is often possible only via simulations; the theories are largely local rules of interactions. In this sense, training the neural network using simulations provides a natural way to place constraints on the concept class that the neural network effectively learns.

A natural question that arises is as follows: *Why does one need to use a neural network when simulations are available?* There are multiple reasons to do this: (i) computational efficiency (ability to rapidly produce forecasts, (ii) generalizability (often simulation parameters might end up overfitting to the data), and (iii) ability to incorporate additional data sources. In this sense, *DL+simulations* appears to be a promising approach for forecasting rather than using either of them individually. See the next section for further discussion.

2 RELATED WORK

2.1 Epidemic Forecasting

Forecasting the spatial and temporal evolution of infectious disease epidemics has been an area of active research over the past couple of decades [11, 12, 37, 56, 57, 61, 66, 67, 71, 73, 76, 77, 83, 84]. We briefly review related work in epidemic forecasting and deep learning pertinent to our problem; see References [3, 23, 68] for more details. We discuss four ILI forecasting methods: causal methods, statistical methods, artificial neural network methods, and hybrid methods. See Figure 1 for a brief summary.

Causal methods. In epidemiology, within-host progression models for ILI include the following: susceptible-infectious-recovered (SIR), susceptible-exposed-infectious-recovered (SEIR),

- **Epidemic forecasting**
  - **Causal methods**
    Pros: employ mathematical models of disease transmission; make multi-fidelity predictions. The models can often capture human decision making and thus provide a path for counterfactual forecasts.
    Cons: generally computationally expensive as they require parameter estimation over a high dimensional space. For networked model, obtaining the needed data to build realistic social networks can be challenging.
  - **Statistical methods**
    Pros: learn patterns from the historical time series; easy to implement the model; fast to train and forecast.
    Cons: usually assume a simple relationship between the inputs and outputs; unable to make heterogeneous high-resolution forecasting.
  - **Artificial neural network (ANN) methods**
    Pros: learn patterns from the historical data; capture non-linear relationship between the inputs and outputs.
    Cons: model performance depends on the availability of large amount of training data; unable to make heterogeneous high-resolution forecasting; lack of explainability; overfitting is a concern due to the small size of the training dataset.
  - **Hybrid methods**
    Pros: combine data driven (statistical and ANN methods) and causal methods; integrate strengths of both methods.
    Cons: have not been explored until now in epidemic forecasting domain.
  - **TDEFSI method**
    Pros: combine deep neural networks and high-resolution epidemic simulations; avoid overfitting using large volume of training data from causal models; enable heterogeneous high-resolution forecasting; yield accurate high-resolution spatiotemporal forecasts using low-resolution time-series data.
- **Data augmentation for time series**
  - **Data augmentation for TSC**
    Pros: generate artificial time series data to reduce classification error using techniques, such as slicing window, warping window, permutating, scaling, cropping, VAEs, GANs, etc.
    Cons: difficult to apply to time series regression problems.
  - **Data augmentation for TSR**
    Pros: generate new time series using transformation or decomposition techniques.
    Cons: not well investigated in epidemic forecasting domain; difficult to generate high-resolution time series.
  - **TDEFSI method**
    Pros: synthesize large volume of high-resolution time series from simulations of causal processes based on mathematical epidemiology theory.
    Cons: challenging to minimize the difference between synthetic data and real data.

Fig. 1. Brief summary of existing ILI forecasting methods and data augmentation techniques.

susceptible-infectious-recovered-susceptible (SIRS), and their extensions [4, 52]. Forecasting methods employing these models are called causal methods (or mechanistic methods), because they are based on the causal mechanisms of infectious diseases. In these methods the underlying epidemic model can be either a compartmental model (CM) [33, 55, 58] or an agent-based model (ABM) [22, 70]. In a compartmental model, a population is divided into compartments (e.g., S, E, I, R). A differential equation system characterizes the change of the sizes of each compartment due to disease propagation and progression. To get county-level epidemics in a compartmental model, one needs to create compartments in each county, where county population sizes and between county travel data become crucial. In an agent-based model, disease spreads among heterogeneous agents through an unstructured network. Dynamics with individual behavior change exhibit significant impact on epidemic and dynamic forecast models [29], which can be implemented using a high-performance computing model [14]. The individual-level details in an agent-based model can be easily aggregated to obtain epidemic data of any resolution, e.g., number of newly infected people in a county in a specific week. Many forecasting methods have been developed based on either CM or ABM [42, 63, 67, 76, 79, 98, 99, 102]. Shaman et al. [76] developed a framework for initializing real-time forecasts of seasonal influenza outbreaks, using a data assimilation technique commonly applied in numerical weather prediction. Tuite et al. [79] used an SIR CM to estimate parameters and morbidity in pandemic H1N1. Yang et al. [98] applied various filter methods to model and forecast influenza activity using an SIRS CM. In Reference [67], the authors proposed a simulation optimization approach based on the SEIR ABM for epidemic forecasting. Hua et al. [42] and Zhao et al. [102] infer the parameters of the SEIR ABM from social media data for ILI forecasting. *Limitations: Causal methods are generally computationally expensive as they require the parameter estimation over a high-dimensional space. As a result the use of such methods for real-time forecasting is challenging.*

Statistical methods. Statistical methods employ statistical and time-series-based methodologies to learn patterns in historical epidemic data and leverage those patterns for forecasting [16, 44]. Popular statistical methods for ILI forecasting include, e.g., generalized linear models (GLM), autoregressive integrated moving average (ARIMA), and generalized autoregressive moving average (GARMA) [5, 9, 28]. Wang et al. [89] proposed a dynamic Poisson autoregressive model with exogenous input variables (DPARX) for flu forecasting. Yang et al. [97] proposed ARGO, an autoregressive-based influenza tracking model for nowcasting incorporating CDC ILI data and Google search data. The extensive work based on ARGO is discussed in Reference [96]. *Limitations: Statistical methods are fast. But they crucially depend on the availability of training data and as such can only produce flat-resolution forecasts. High-resolution forecasts must be calculated by multiplying the flat-resolution forecasts with high-resolution population proportions. The trained models could not capture the heterogeneous dynamics between high-resolution regions. Furthermore, since they are purely data driven, they do not capture the underlying causal mechanisms. As a result epidemic dynamics affected by behavioral adaptations are usually hard to capture.*

Artificial neural network methods. Artificial neural networks (ANN) have gained increased prominence in epidemic forecasting due to their self-learning ability without prior knowledge. Xu et al. [95] first introduced feed-forward neural network (FNN) into surveillance of infectious diseases and investigated its predictive utility using CDC ILI data, Google search data, and meteorological data. Recurrent neural network (RNN) has been demonstrated to be able to capture dynamic temporal behavior of a time sequence. In Reference [85] Volkova et al. built an LSTM model for short-term ILI forecasting using CDC ILI and Twitter data. Venna et al. [82] proposed an LSTM-based method that integrates the impacts of climatic factors and geographical proximity to achieve better forecasting performance. Wu et al. [93] constructed a deep learning structure combining RNN and convolutional neural network to fuse information from different sources. Deng

et al. [25] recently designed a cross-location attention-based graph neural network for learning time-series embeddings and location aware attentions. *Limitations: Just like statistical methods, ANN-based forecasting methods are data driven and have similar limitations. In addition, the model performance usually depends on the availability of a very large training dataset. Another well-known limitation of ANN methods is their ability to explain the resulting forecasts.*

Hybrid methods. Hybrid methods combine data-driven and causal methods. They are attractive as they can borrow the best from both worlds [45]. The authors in Reference [69] proposed a dynamic Bayesian model for influenza forecasting that combines the machine learning approach and a compartmental model to explicitly account for systematic deviations between mechanistic models and the observed data. Such methods have shown promise as evidenced in recent papers on the study of physical and biological systems [31, 32, 40, 46–49, 91, 94]—see Reference [46] for a discussion on this subject.

TDEFSI method. Our method combines the deep neural networks and high-resolution epidemic simulations to enable accurate weekly high-resolution ILI forecasts from flat-resolution observations. Compared with causal methods, TDEFSI avoids searching optimal disease model parameters over a high-dimensional space, because it does not need to identify any specific causal models for the forecasting. Compared with data-driven methods (statistical and neural network methods), TDEFSI explicitly models spatial and social heterogeneity in a region from the training data. It can capture the heterogeneous dynamics between high-resolution regions, as well as underlying causal processes and mathematical theories. In addition, the large volume of synthetic training data helps TDEFSI to overcome the risk of overfitting due to sparse observation data.

2.2 Data Augmentation for Time Series

Data augmentation in deep neural networks is the process of generating artificial data to reduce overfitting. It has been shown to improve deep neural network's generalization capabilities in many tasks especially in computer vision tasks such as image or video recognition [75]. Various augmentation techniques have been applied to specific problems, including affine transformation of the original images [74, 81, 92] and unsupervised generation of new data using Generative Adversarial Nets (GANs) [39, 60, 72, 103] or variational autoencoder (VAE) models [74], and so on. However, the techniques for image augmentation do not generalize well to time series. The main reason is that image augmentation is not expected to change the class of an image, while for time-series data, one cannot confirm the effect of such transformations on the nature of a time series. In what follows we introduce related work on time-series data augmentation.

Data augmentation for time-series classification. For time-series classification (TSC) problems, one of the most popular methods is the slicing window technique, originally introduced for deep CNNs in Reference [24]. The method was inspired by the image cropping technique for computer vision tasks [101]. In Reference [53], it was adopted to improve the CNNs' mortgage delinquency prediction using customer's historical transactional data. The authors in Reference [51] used it to improve the Support Vector Machines accuracy for classifying electroencephalographic time series. The authors in Reference [80] proposed a novel data augmentation method (including window slicing, permutating, rotating, time-warping, scaling, magnitude-wrapping, jiterring, cropping) specific to wearable sensor collected time-series data. Le Guennec et al. [54] extended the slicing window technique with a warping window that generates synthetic time series by warping the data through time. It extracts multiple small-size windows from a single window and lengthens/shortens a part of the window data, respectively. The methods are reported to reduce classification error on several types of time-series data. Forestier et al. [34] proposed to average a set of time series as a new synthetic series. It relies on an extension of Dynamic Time Warping (DTW) Barycentric Averaging (DBA).

Data augmentation for time-series regression. Unlike data augmentations for TSC, data augmentation for time-series regression (TSR) has not been well investigated yet to the best of our knowledge. Bergmeir et al. [10] presented a method using Box-Coxfor transformation followed by an STL decomposition to separate the time series into trend, seasonal part, and remainder. The remainder was then bootstrapped using a moving block bootstrap, and a new series was assembled using this bootstrapped remainder.

All above methods for TSC or TSR apply techniques directly on observed time sequences, which generate synthetic data at the same resolution as the original data. In our problem, we try to forecast at a higher resolution when there is no or very sparse high-resolution observations.

TDEFSI method. We generate synthetic high-resolution data using high performance computing-based simulations of well-accepted causal processes that capture epdemic dynamics. Different from data augmentation techniques introduced above, we synthesize high-resolution data that are not available or quite sparse in the real world.

3 PROBLEM SETUP

Given an observed time series of weekly ILI incidence for a specific region, we focus on predicting ILI incidence for both the region and its subregions in short-term. Without loss of generality, in this article we consider making predictions for a state of the USA and all counties in the state, using observations only from CDC state-level ILI incidence data [20]. In this setting, state-level forecasting is flat resolution, while county-level forecasting is high resolution. The proposed framework is not limited to this setting and can be generalized for subregion forecasting in any region, e.g., state-level forecasting in a country where only national-level surveillance data are available. Our proposed method is different from traditional ILI incidence forecasting methods in that the model is trained on synthetic ILI incidence data but forecasts by taking ILI surveillance data as inputs.

Let $\mathbf{y} = \langle y_1, y_2, \ldots, y_T, \ldots \rangle$ denote the sequence of weekly state-level ILI incidence, where $y_i \in \mathbb{R}$. Let $\mathbf{y}^C = \langle y_1^C, y_2^C, \ldots, y_T^C, \ldots \rangle$ denote the sequence of weekly ILI incidence for a particular county C within the state. Assume that there are K counties $\mathcal{D} = \{C_1, C_2, \ldots, C_K\}$ in the state. Let $\mathbf{y}_t^{\mathcal{D}} = \{y_t^C | C \in \mathcal{D}\}$ denote ILI incidence of all counties in the state at week t. Suppose we are given only state-level ILI incidence up to week T. The problem is defined as predicting both state-level and county-level incidence at week t, where $t = T + 1$, denoted as $\mathbf{z}_t = (y_t, \mathbf{y}_t^{\mathcal{D}}), \mathbf{z}_t \in \mathbb{R}^{K+1}$, given $\langle y_1, y_2, \ldots, y_T \rangle$.

In our problem, when training the deep neural network models, we consider three types of physical consistency requirements based on epidemiologic domain knowledge. They are *temporal consistency*, *spatial consistency*, and *non-negative consistency*. (i) Temporal consistency: The ILI diseases transmit via person to person contacts. The number of infected cases at the current time point depends on the number of infected cases at the previous time points. In addition, infected persons' incubation periods and infectious periods vary due to the heterogeneity among individuals. In our work, we use the long short term memory (LSTM) network [41] to capture the temporal dependencies among variables. (ii) Spatial consistency: The high-resolution ILI incidence should be consistent with the flat-resolution ILI incidence. In our problem, this consistency is represented as $y_t = \sum_{C \in \mathcal{D}} y_t^C$, i.e., the state incidence equals the sum of ILI incidence at the county level. (iii) Non-negative consistency: The number of infected cases at time t is either zero or a positive value, denoted as $y_t, y_t^C \geq 0$.

4 TDEFSI

4.1 Framework

The TDEFSI framework consists of three major components (shown in Figure 2): (i) *Disease model parameter space construction*: Given a state and an existing disease model, we estimate a marginal

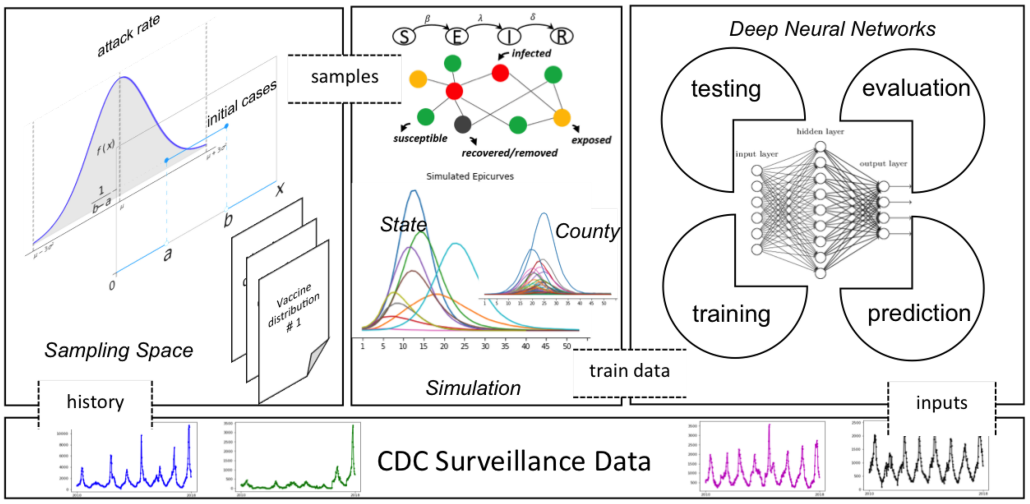


Fig. 2. TDEFSI framework. In this framework, a region-specific disease parameter space for a disease model is constructed based on historical surveillance data. Synthetic training data consisting of both state-level and county-level weekly ILI incidence curves is generated by simulations parameterized by samples from the parameter space. An LSTM-based deep neural network model is trained on the synthetic data. The trained model produces forecasts by taking surveillance data as the input.

distribution for each model parameter based on the surveillance data of the state and its neighbors; (ii) *Synthetic training data generation*: We generate a synthetic training dataset at both flat-resolution and high-resolution scales for that state by running simulations parameterized from the parameter space; and (iii) *Deep neural network training and forecasting*: We design a two-branch deep neural network model trained on the synthetic training dataset and use surveillance data as its inputs for forecasting. We will elaborate on the details in the following subsections.

4.2 SEIR-based Epidemic Simulation

We simulate the spread of the disease in a synthetic population via its social contact network. In this work, we use the synthetic social contact network of each state in the USA (a brief description of the methodology used for constructing the synthetic population and the social network can be found in Appendix A). The SEIR disease model is widely used for ILI diseases [52]. Each person is in one of the following four health states at any time: susceptible (S), exposed (E), infectious (I), recovered or removed (R). A person v is in the susceptible state until he becomes exposed. If v becomes exposed, then he remains so for $p_E(v)$ days, called the incubation period, during which he is not infectious. Then he becomes infectious and remains so for $p_I(v)$ days, called the infectious period. Both $p_E(v)$ and $p_I(v)$ are sampled from corresponding distributions, as shown in Algorithm 1, e.g., $p_E(v) \sim \{1 : 0.3, 2 : 0.5, 3 : 0.2\}$ means that an exposed person will remain so for 1 day with probability 0.3, 2 days with probability 0.5, and 3 days with probability 0.2, similar to $P_I(v)$. Finally, he becomes removed (or recovered) and remains so permanently. While the SEIR model characterizes within-host disease progression, between-host disease propagation is modeled by transmissions from person to person with a probability parameter τ, through either complete mixing or heterogeneous connections between people. With our contact network model, the disease spreads in a population in the following way. It can only be transmitted from an infectious node to a susceptible node. On any day, if node u is infectious and v is susceptible, then disease transmission from u to v occurs with probability $p(\tau, w(u, v))$, where $w(u, v)$ represents

the contact duration between node u and node v. The disease propagates probabilistically along the edges of the contact network.

Various simulators are developed to model human mobility, disease spread, and public health intervention. They include compartment-based patch models [33, 55, 58], as well as agent-based models such as EpiFast [14], GSAM [70], and FluTE [22]. Any of these simulators can be used in TDEFSI to generate synthetic training data. In this work, we adopt an agent-based simulator EpiFast [14]. The outputs are individual infections with their days of being infected in a simulated season. They can be aggregated to any temporal and spatial scale, such as daily (weekly) state-(county-) level ILI incidence. Vaccine intervention I_V can be implemented in EpiFast simulations, by specifying the quantity of vaccines applied to the population in each week. Next, we describe how to estimate a distribution on the parameter space $\mathcal{P}(p_E, p_I, \tau, N_I, I_V)$ from CDC historical data, where N_I denotes the initial number of infections. In our simulations, N_I of the population are infectious while all the rest are susceptible at the beginning of the simulation.

4.3 Disease Model Parameter Space

Of the parameters, (p_E, p_I) can be taken from literature [59]. We assume that each of (τ, N_I, I_V) follows a distribution that can be estimated from historical data. For clarity, we define an epidemiological week in a calendar year as **ew**, and a seasonal week in a flu season as **sw**, where $ew(40)$ is $sw(1)$. The historical time series of CDC surveillance data (refers to historical training data) used to construct parameter space is split into seasons at $ew(40)$ of each year. That is, each flu season starts from $ew(40)$ of a calendar year and ends in $ew(39)$ of the next year. Note that this applies to the USA, but **sw** may be specified differently for other countries.

We want to highlight that the number of clinically attended cases and the reported or tested cases are lower than the actual number of cases in the population. Additionally, reporting rates can vary between regions. To address the gap between ILINet case count and population case count, we scale the former with a scaling factor, called surveillance ratio. The ratio is different among different states. See more details of the surveillance ratio in Appendix A.2.

First, we collect observations of each parameter value as follows:

- **Initial Case Number (N_I)**: We collect the ILI incidence of $sw(1)$ of each season for the target state and its neighboring states (i.e., geographically contiguous states).
- **Vaccine Intervention (I_V)**: We collect vaccination schedules of the past influenza seasons in the USA [18]. Each schedule consists of timing and percentage coverage of vaccine application throughout the season. Vaccine efficacy (reduction of disease transmission probability) and compliance rate (probability that a person will take the vaccine) are set according to a survey used in Reference [86], which is conducted by Gfk.com, under the National Institutes of Health grant no. 1R01GM109718. This survey collects data on demographics of the respondents and their preventive health behaviors during a hypothetical influenza outbreak. We assume that each person follows a common compliance rate and the state-level vaccine schedule is the same as the nationwide schedule.
- **Transmissibility (τ)**: First, we compute the overall attack rate (i.e., the fraction of population getting infected in the season) of each historical season for the target state and its neighboring states. Then for each attack rate ar, say of season s and state r, we calibrate a transmissibility value as the solution to $\min_\tau |AR(EpiFast(\tau, P_E, P_I, N_I, I_V)) - ar|$, where p_E and p_I are sampled for each person from the distributions shown in Table 1; N_I is the initial case number of season s and state r; I_V is the vaccination schedule for season s; $EpiFast(\cdot)$ is a simulation run on the population of state j with the parameters $(\tau, P_E, P_I, N_I, I_V)$; and $AR(\cdot)$ computes attack rate from the output of $EpiFast(\cdot)$. Details of this process are shown in Algorithm 1.

ALGORITHM 1: Calibrating disease model parameter τ

Input: Simulator PS, CDC historical data *histCDC*, and synthetic social contact networks $Network$.
Output: Calibrated τ^*.
$p_E \sim \{1:0.3, 2:0.5, 3:0.2\}$ [59, 86];
$p_I \sim \{3:0.3, 4:0.4, 5:0.2, 6:0.1\}$ [59, 86];
$I_V = \emptyset$;
regions = {state and its adjacent neighbors};
seasons = {available seasons of *histCDC*};
$\tau^* = \emptyset$;
for *r in regions* **do**
  for *s in seasons* **do**
    $totalili_{(r,s)} = TOTAL(histCDC_{(r,s)})$;
    $ar_{(r,s)} = \frac{totalili_{(r,s)}}{population_{(r)}}$;
    $\tau^*_{(r,s)} = \min_\tau |AR(EpiFast(\tau, P_E, P_I, I_V, N_{I(r,s)}, Network_{(r,s)})) - ar_{(r,s)}|$;
    $\tau^* = \tau^* \cup \tau^*_{(r,s)}$
  end
end

Second, for τ and N_I, we fit the collected samples to several distributions including normal, uniform.

Then we run KS-test to choose a best well fit distribution (refer to Appendix A.3 for more details). For I_V, we assume the six vaccination schedules follow a discrete uniform distribution. In this way, a region-specific parameter space $\mathcal{P}$ is constructed.

We first implement our TDEFSI framework without considering interventions in the simulations. Then we add I_V to $\mathcal{P}$ to generate more realistic synthetic training data. This will improve the forecasting performance of TDEFSI. We will discuss the impact of including I_V on the forecasting performance of TDEFSI in Section 5.9.

4.4 Training Dataset from Simulations

For each simulation run, a specific parameter setting is sampled from $\mathcal{P}$, and the simulator is called to generate daily individual health states. These individual health states are aggregated to get state- and county-level weekly incidences, called *synthetic epicurves*. Week 1 in the synthetic epicurve corresponds to $sw(1)$ of a flu season. Large volumes of high-resolution synthetic data are generated by repeating the sampling and simulating process. Let us denote all simulated epicurves by $\Omega = \{(\mathbf{y}_{(i)}, \mathbf{y}^{\mathcal{D}}_{(i)}) \in \mathbb{R}^{\ell \times (K+1)} | i = 1, 2, \ldots, r\}$, where ℓ is the length of an epicurve (number of weeks), K is the number of counties in the state, and r is the total number of simulation runs. Algorithm 2 describes the generating process.

Compared with CDC surveillance data, the training dataset Ω is prominent in two aspects: (i) it includes high-resolution spatial dependencies between subregions and (ii) the large volume of synthetic training data reduces the possibility of overfitting when training a deep neural network model. Thus, the trained model has better generalization ability.

4.5 TDEFSI: A Deep Neural Network Model

The Long Short Term Memory (LSTM) network [41] is adopted in our neural network architecture to capture the inherent temporal dependency in the weekly incidence data. Figure 3 shows unrolled k-stacked LSTM layers. Each LSTM layer consists of a sequence of cells. The number

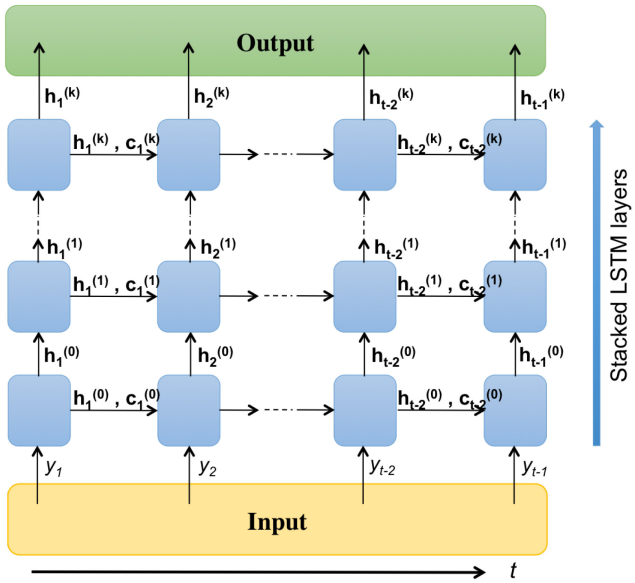


Fig. 3. Unrolled k-stacked LSTM layers. Each LSTM layer consists of a sequence of cells. The number of cells depends on the number of input time points. In this figure, the input is a time series of $y_1, \ldots, y_{t-1}$, the output comprises all the cell outputs $\mathbf{h}^{(k)}$ from the last layer k ("last" depthwise, not timewise). Each LSTM layer consists of $t-1$ cells. In the first LSTM layer, a cell will work as described in 1, e.g., cell 2 takes y_2, cell state $\mathbf{c}_1^{(0)}$ and cell output $\mathbf{h}_1^{(0)}$ from the previous cell 1 as inputs, then outputs $(\mathbf{c}_2^{(0)}, \mathbf{h}_2^{(0)})$ so you could feed them into next cell and feed $\mathbf{h}_2^{(0)}$ into next layer. The first LSTM layer take $y_1, \ldots, y_{t-1}$ as the input, the second layer take $\mathbf{h}_1^{(0)}, \ldots, \mathbf{h}_{t-1}^{(0)}$ as the input, and rest of the layers behave in the same manner.

ALGORITHM 2: Generating Training Dataset for TDEFSI

Input: Simulator PS, and Parameter space $\mathcal{P}$.
Output: Simulated epicurves $\Omega = \{(\mathbf{y}_{(i)}, \mathbf{y}_{(i)}^{\mathcal{D}}) | i = 1, 2, \ldots, r\}$.

$\Omega = \emptyset$;
for $i = 1$ *to* r **do**
  P = Sample($\mathcal{P}$);
  $(\mathbf{y}_{(i)}, \mathbf{y}_{(i)}^{\mathcal{D}})$ = PS(P);
  $\Omega = \Omega \cup (\mathbf{y}_{(i)}, \mathbf{y}_{(i)}^{\mathcal{D}})$
end

of cells depends on the number of input time points. In this figure, the input is a time series of $y_1, \ldots, y_{t-1}$, the output comprises all the cell outputs $\mathbf{h}^{(k)}$ from the last layer k ("last" depthwise, not timewise). Each LSTM layer consists of $t-1$ cells. In the first LSTM layer (layer 0), a cell will work as described in Equation (1), e.g., cell 2 takes y_2, cell state $\mathbf{c}_1^{(0)}$, and cell output $\mathbf{h}_1^{(0)}$ from the previous cell 1 as inputs, and then outputs $(\mathbf{c}_2^{(0)}, \mathbf{h}_2^{(0)})$ so you could feed them into the next cell and feed $\mathbf{h}_2^{(0)}$ into the next layer (layer 1). The first LSTM layer takes $y_1, \ldots, y_{t-1}$ as the input, the second layer takes $\mathbf{h}_1^{(0)}, \ldots, \mathbf{h}_{t-1}^{(0)}$ as the input, and the rest of the layers behave in the same manner.

Let $H^{(i)}, 0 \leq i \leq k$ be the dimension of the hidden state in layer i. For the first layer, assume the input of the current cell is y_{t-1}. Then the computation within the cell is described mathematically

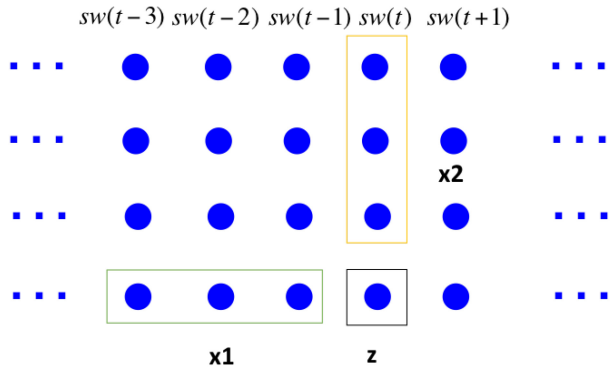


Fig. 4. Within-season and between-season observations as the input for the TDEFSI neural network model. In this graph, there are four flu seasons (rows). Nodes in each row denote weekly ILI incidence in each season, which are ordered by sw. For a target week $sw(t)$ (black square), the model observes two kinds of information: (i) within-season observations $\mathbf{x1}$—the ILI incidence from the previous weeks back from week $sw(t)$ (green rectangular); (ii) between-season observations $\mathbf{x2}$—the historical ILI incidence from similar weeks of the past seasons (yellow rectangular). $\mathbf{z}$ is the ILI forecasting of the target week. $\mathbf{x1}$ and $\mathbf{x2}$ are state-level ILI, while $\mathbf{z}$ includes state and county-level ILI.

as:

$$
\begin{aligned}
\mathbf{i}_{t-1}^{(0)} &= \sigma(\mathbf{W}_i^{(0)} \cdot y_{t-1} + \mathbf{U}_i^{(0)} \cdot \mathbf{h}_{t-2}^{(0)} + \mathbf{b}_i^{(0)}) \in \mathbb{R}^{H^{(0)}} \\
\mathbf{f}_{t-1}^{(0)} &= \sigma(\mathbf{W}_f^{(0)} \cdot y_{t-1} + \mathbf{U}_f^{(0)} \cdot \mathbf{h}_{t-2}^{(0)} + \mathbf{b}_f^{(0)}) \in \mathbb{R}^{H^{(0)}} \\
\mathbf{o}_{t-1}^{(0)} &= \sigma(\mathbf{W}_o^{(0)} \cdot y_{t-1} + \mathbf{U}_o^{(0)} \cdot \mathbf{h}_{t-2}^{(0)} + \mathbf{b}_o^{(0)}) \in \mathbb{R}^{H^{(0)}} \\
\widetilde{\mathbf{C}}_{t-1}^{(0)} &= tanh(\mathbf{W}_C^{(0)} \cdot y_{t-1} + \mathbf{U}_C^{(0)} \cdot \mathbf{h}_{t-2}^{(0)} + \mathbf{b}_C^{(0)}) \in \mathbb{R}^{H^{(0)}} \\
\mathbf{C}_{t-1}^{(0)} &= \mathbf{f}_{t-1}^{(0)} \circ \mathbf{C}_{t-2}^{(0)} + \mathbf{i}_{t-1}^{(0)} \circ \widetilde{\mathbf{C}}_{t-1}^{(0)} \in \mathbb{R}^{H^{(0)}} \\
\mathbf{h}_{t-1}^{(0)} &= \mathbf{o}_{t-1}^{(0)} \circ \mathbf{C}_{t-1}^{(0)} \in \mathbb{R}^{H^{(0)}},
\end{aligned}
\tag{1}
$$

where σ and $tanh$ are sigmoid and tanh activation functions. $\mathbf{W} \in \mathbb{R}^{H^{(0)}}$, $\mathbf{U} \in \mathbb{R}^{H^{(0)} \times H^{(0)}}$, and $\mathbf{b} \in \mathbb{R}^{H^{(0)}}$ are learned weights and bias. $\mathbf{C}_{t-2}^{(0)}, \mathbf{h}_{t-2}^{(0)}$ are the cell state and output of the previous cell. Operator $\circ$ denotes element-wise product (Hadamard product). The cell computation is similar in the layer i, but with y_{t-1} being replaced by $\mathbf{h}_{t-1}^{(i-1)} \in \mathbb{R}^{H^{(i-1)}}$ and $\mathbf{W} \in \mathbb{R}^{H^{(i)} \times H^{(i-1)}}$.

In traditional time-series models, ILI incidences of the previous few weeks are used as the observations for the prediction of the current week. In TDEFSI, we use two kinds of observations: (i) *Within-season observations*, denoted as $\mathbf{x1} = \langle y_{t-a}, \ldots, y_{t-1} \rangle$, are ILI incidence from previous a weeks that are back from time step t. (ii) *Between-season observations*, denoted as $\mathbf{x2} = \langle y_{t-\ell*b}, \ldots, y_{t-\ell*1} \rangle$, are ILI incidences of the same sw from the past b seasons. They are used as the surrogate information to improve forecasting performance. As shown in Figure 4, for example, there are four seasons ordered by sw. The within-season observations are ILI incidence of previous $a = 3$ weeks in current season. The between-season observations are ILI incidence of the same $sw(t)$ from the past $b = 3$ seasons.

In TDEFSI model, we design a two-branch LSTM-based deep neural network model to capture temporal dynamics of within-season and between-season observations. As shown in Figure 5, the left branch consists of stacked LSTM layers that encode within-season observations $\mathbf{x1} = \langle y_{t-a}, \ldots, y_{t-1} \rangle$. The right branch is also LSTM based and encodes between-season observations

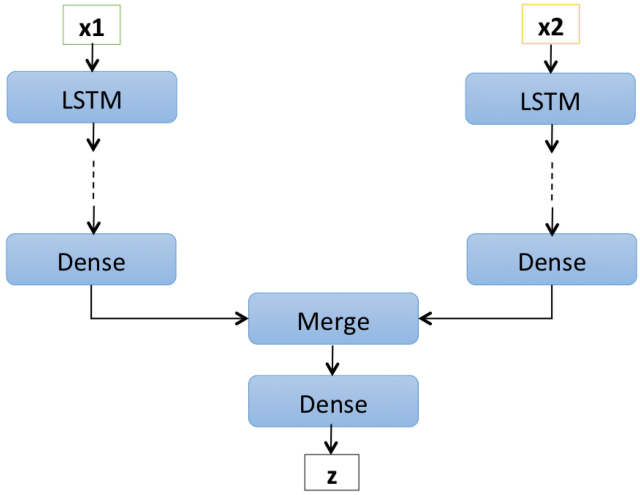


Fig. 5. TDEFSI neural network architecture. This architecture consists of two branches. The left branch consists of stacked LSTM layers that encodes state-level within-season observations **x1**, and the right branch consists of stacked LSTM layers that encodes state-level between-season observations **x2**. A merge layer is added to combine two branches and the output **z** is the state and county-level predictions.

$\mathbf{x2} = \langle y_{t-\ell*b}, \ldots, y_{t-\ell*1} \rangle$. A merge layer is added to combine the outputs of two branches. The final output is $\hat{\mathbf{z}}_t$ that consists of state-level and county-level predictions (as defined in Section 3).

In the left branch, the output of the Dense layer is as follows:

$$\mathbf{O}_l = \psi_l\left(\mathbf{w}_l \cdot \mathbf{h}_{t-1}^{(k_l)} + \mathbf{b}_l\right) \in \mathbb{R}^H, \tag{2}$$

where k_l is the number of LSTM layers in the left branch, H is the dimension of output of the left branch, $\mathbf{w}_l \in \mathbb{R}^{H \times H^{(k_l)}}$ and $\mathbf{b}_l \in \mathbb{R}^H$, and ψ_l is the activation function.

Similarly, the output of the Dense layer in the right branch is as follows:

$$\mathbf{O}_r = \psi_r\left(\mathbf{w}_r \cdot \mathbf{h}_{t-1}^{(k_r)} + \mathbf{b}_r\right) \in \mathbb{R}^H, \tag{3}$$

where k_r is the number of LSTM layers in the right branch, H is the dimension of output of the right branch, $\mathbf{w}_r \in \mathbb{R}^{H \times H^{(k_r)}}$ and $\mathbf{b}_r \in \mathbb{R}^H$, and ψ_r is the activation function.

The merge layer combines the output from two branches by addition, denoted as:

$$\hat{\mathbf{z}}_t = \psi(\mathbf{w}[\mathbf{O}_l \oplus \mathbf{O}_r] + \mathbf{b}) \in \mathbb{R}^{K+1}, \tag{4}$$

where $\mathbf{w} \in \mathbb{R}^{(K+1) \times H}$, $\mathbf{b} \in \mathbb{R}^{K+1}$, ψ is the activation function, and $\oplus$ denotes the element-wise addition.

This LSTM-based deep neural network model is able to connect historical ILI incidence information to the current prediction. It also allows long-term dependency learning without suffering the gradient vanishing problem. The number of LSTM layers is a hyperparameter that we tuned by grid searching.

We are interested in a predictor f, which predicts the current week's state-level and county-level incidence $\mathbf{z}_t$ based on the previous a weeks of within-season state-level ILI incidence **x1** and the previous b seasons of between-season state-level ILI incidence **x2**:

$$\hat{\mathbf{z}}_t = f([\mathbf{x1}, \mathbf{x2}]_t, \theta), \tag{5}$$

where θ denotes parameters of the predictor, $\hat{\mathbf{z}}_t$ denotes the prediction of $\mathbf{z}_t$. Note that *the output of f is always one week ahead forecast* in our model.

The optimization objective is as follows:

$$\min_{\theta} \mathcal{L}(\theta) = \sum_{t} \|\mathbf{z}_t - f([\mathbf{x1}, \mathbf{x2}]_t, \theta)\|_2^2 + \mu\phi(\hat{\mathbf{z}}_t) + \lambda\delta(\hat{\mathbf{z}}_t), \tag{6}$$

where $\phi(\hat{\mathbf{z}}_t)$ is an activity regularizer added to the outputs for spatial consistency constraint $\hat{y}_t = \sum_{C \in \mathcal{D}} \hat{y}_t^C$:

$$\phi(\hat{\mathbf{z}}_t) = \left| \hat{y}_t - \sum_{C \in \mathcal{D}} \hat{y}_t^C \right|, \tag{7}$$

and $\delta(\hat{\mathbf{z}}_t)$ is an activity regularizer added to the outputs for non-negative consistency constraint $\hat{y}_t, \hat{y}_t^C \geq 0$:

$$\delta(\hat{\mathbf{z}}_t) = \left| \frac{1}{K+1} \sum \max(-\hat{\mathbf{z}}_t, \mathbf{0}) \right|, \tag{8}$$

μ, λ are two pre-specified hyperparameters, $\min(\hat{\mathbf{z}}_t, \mathbf{0})$ returns element-wise minimum value, K is the number of counties in the state, and $\delta(\hat{\mathbf{z}}_t)$ returns the absolute mean of element-wise minimum values. The Adam optimization algorithm [50] is used to learn θ. How the activity regularizers affect the model performance will be discussed in Section 5.8.

Variants of TDEFSI. The two-branch neural network architecture has multiple variants: (i) *TDEFSI*: Two-branch neural network as shown in Figure 5. (ii) *TDEFSI-LONLY*: Only the left branch is used to take within-season observations. (iii) *TDEFSI-RDENSE*: The left branch comprises of stacked LSTM layers, while the right branch only uses Dense layers, which means that the model does not care about the temporal relationship between between-season data points. We will discuss the results of different variants in Section 5.

Training and forecasting. In the training process, we use synthetic training data Ω to train the TDEFSI models. The historical surveillance data are only used for constructing the disease model parameter space $\mathcal{P}$. In the predicting step, the trained model takes state-level surveillance as input and makes one week ahead forecasts at both state and county levels. TDEFSI models are trained once before the target flu season starts, then can be used for forecasting throughout the season.

Multi-step forecasting. In practical situations, we are interested in making predictions for several weeks ahead using iterative method. In TDEFSI, the left branch of the model appends the most recent state-level prediction to the input for predicting the target of the next week, and the right branch uses the state-level ILI incidences from the past seasons with *sw* equal to the next week number.

5 EXPERIMENTS

In this section, we will describe datasets, comparison methods, experiment setup, and evaluation metrics. A brief summary of TDEFSI settings is shown in Figure 6. And we present results of performance analysis on both simulated testing data and real ILI testing data and conduct sensitivity analysis on physical consistency constraints and vaccination-based interventions. We also use a case study to demonstrate the capability of TDEFSI model to provide uncertainty in predictions. A brief summary of the experiment results is shown in Figure 7. In all experiments the models are trained and tested for each state independent of other states.

5.1 Datasets

5.1.1 Real Dataset. CDC ILI incidence [20]: The CDC surveillance data used in the experiments are the weekly ILI incidence at state level from 2010 $ew(40)$ to 2018 $ew(18)$. Note that they may be revised continuously until the end of a flu season. We use the finalized data in this article. *ILI Lab*

- **Real Dataset**
  Weekly CDC state level ILINet-reported case counts for all states in the USA (2010-2018) (total 397 data points per state)
  - *real-training:* the beginning 80% of season 2010-2011 to 2015-2016 (251 data points per state)
  - *real-validating:* the last 20% of season 2010-2011 to 2015-2016 (63 data points per state)
  - *real-testing:* season 2016-2017 to 2017-2018 (83 data points per state)

  Weekly county level ILI Lab tested flu positive counts for NJ (2016-2018)
  - *County level real-evaluating:* 64 data points per county of NJ
- **Simulated Dataset**
  VA: 1000 epicurves in vaccine-case and 1000 epicurves in base-case.
  NJ: 1000 epicurves in vaccine-case and 1000 epicurves in base-case.
  - *sim-training:* 80% of 1000 epicurves
  - *sim-validating:* 15% of 1000 epicurves
  - *sim-testing:* 5% of 1000 epicurves
- **Disease Model Simulator**
  SEIR agent-based model – EpiFast
- **Disease Model Parameter Space**
  $\mathcal{P}(p_E, p_I, \tau, N_I, I_V)$, the learned distribution for $\mathcal{P}$ is shown in Table 5.
- **TDEFSI Neural Network Models**
  The architecture (e.g. the number of layers or hidden units) for TDEFSI and its variants is described in Section 5.3. The input dimension $a = 52$ and $b = 5$, spatial and non-negative coefficients are set with $(\mu, \lambda)_{VA} = (0.1, 0.1)$, $(\mu, \lambda)_{NJ} = (1, 0.01)$. TDEFSI models are trained with vaccine-case sim-training dataset. We choose the final model by grid searching using sim-validating dataset. Adam optimizer with all default values are used. In the training process, the best models are selected by early stopping when the validation accuracy does not increase for 50 consecutive epochs, and the maximum epoch number is 300.
- **Prediction Target**
  ILINet-report case counts forecasting with $horizon = \{1, 2, 3, 4, 5\}$.

Fig. 6. Brief summary of TDEFSI settings.

tested flu positive counts of New Jersey [26]: To evaluate the county-level forecasting performance, we collect state-level and county-level ILI Lab tested flu positive counts of season 2016–2017 and 2017–2018 in NJ. The data are available from $ew(40)$ to the next year's $ew(20)$. We use it as the ground truth when evaluating county-level forecasting. *Google data* [36, 38]: The Google correlate terms (keyword: influenza) of each state are queried; we choose the top 100 terms. Then the Google Health Trends of each correlated term for each state is collected and aggregated weekly from 2010 $ew(40)$ to 2018 $ew(18)$. *Weather data* [21]: We download daily weather data (including max temperature, min temperature, precipitation) from Climate Data Online (CDO) for each state and compute weekly data as the average of daily data from 2010 $ew(40)$ to 2018 $ew(18)$. Google data and weather data are used as surrogate information in comparison methods (described in Section 5.2).

We divide the data into *real-training:* the beginning 80% of season 2010–2011 to season 2015–2016 (251 data points per state); *real-validating:* the last 20% of season 2010–2011 to season 2015–2016 (63 data points per state); *real-testing:* season 2016–2017 to season 2017–2018 (83 data points

- **Exploratory Analysis of Spatial Dynamics of NJ County Level Dataset**
  The spreading process shows spatial heterogeneity over the counties, which is related with the population size and the commute flow.
- **TDEFSI Performance on a Simulated Testing Dataset**
  The best models at the state and county levels are the models with $a = 52$.
- **TDEFSI Performance on a Real Seasonal ILI Testing Dataset**
  - ***Performance of Flat-resolution Forecasting.***
    TDEFSI and its variants achieve comparable or better performance than baselines evaluted on real-testing dataset.
  - ***Performance of High-resolution Forecasting.***
    TDEFSI and its variants outperform baselines evaluated on county level real-evaluating dataset.
  - ***Overall performance.***
    TDEFSI and its variants achieve better forecasting performance. Among the three proposed models, TDEFSI and TDEFSI-RDENSE outperform TDEFSI-LONLY.
- **Physical Consistency Constraints Analysis**
  The spatial and non-negative consistency constraints with proper λ and μ help improve the forecasting performance. However, the optimal λ and μ vary between different regions.
- **Vaccination-based Interventions**
  The model learned using simulations that incorporate vaccinations yields better generalizability to unseen surveillance data.
- **Prediction Uncertainty Estimation**
  The proposed model provides a natural way to estimate the prediction uncertainty using MC Dropout technique.

Fig. 7. Brief summary of our experimental analysis.

per state); and *county-level real-evaluating:* county-level ILI lab tested flu positive counts for NJ (64 data points per county of NJ). For TDEFSI models, we use the training dataset to learn disease parameter space, while for baselines, we use training dataset to train the model directly and use validating dataset to validate and choose the final models. Testing and county-level evaluating datasets are used for all methods to evaluate their performance. And the final result of each method is the average value of 10 trials.

5.1.2 Simulated Dataset. For each state, we generate 1,000 simulated curves of weekly ILI incidence at both state level and county level. Of each curve, the first week $sw(1)$ corresponds to epi-week 40 $ew(40)$ of real seasonal curves. We divide the data into *sim-training:* 80% of 1,000 simulated curves; *sim-validating:* 15% of 1,000 simulated curves; and *sim-testing:* 5% of 1,000 simulated curves. The synthetic data are only used for training and validating of TDEFSI models. No baselines are applied for synthetic data.

5.2 Methods Used for Our Comparative Analysis

Our method is compared with five state-of-the-art ANN methods, statistical methods, and causal methods. They are as follows:

- *LSTM* (CDC data) [41] and *AdapLSTM* (CDC + weather data) [82] representing artificial neural network methods;

Table 1. Marginal Distributions of the Parameter Spaces for VA and NJ

Parameter	State	Name	Distribution	*P*-value
p_E	VA	Discrete distribution	(1:0.3, 2:0.5, 3:0.2) [59, 86]	—
	NJ	Discrete distribution	(1:0.3, 2:0.5, 3:0.2) [59, 86]	—
p_I	VA	Discrete distribution	(3:0.3, 4:0.4, 5:0.2, 6:0.1) [59, 86]	—
	NJ	Discrete distribution	(3:0.3, 4:0.4, 5:0.2, 6:0.1) [59, 86]	—
τ	VA	Normal	$\mathcal{N}(\mu = 4.88\text{e}{-5}, \delta = 9.33\text{e}{-7})$	0.74
	NJ	Normal	$\mathcal{N}(\mu = 4.63\text{e}{-5}, \delta = 1.05\text{e}{-6})$	0.85
N_I	VA	Uniform	$\mathcal{U}(7355, 16278)$	0.85
	NJ	Uniform	$\mathcal{U}(567, 7647)$	0.40
I_V	VA	Discrete uniform	6 vaccination schedules [18]	—
	NJ	Discrete uniform	6 vaccination schedules [18]	—

The null hypothesis for the two-sample KS test is that both groups were sampled from populations with identical distributions. If the p-value returned by the KS test is less than a significance level, then we reject the null hypothesis. In our experiments, we do not specify a significance level but instead choose the distribution with the largest p-value among multiple assumed distributions.

- *SARIMA* (CDC Data) [9] and *ARGO* (CDC + Google data) [97] representing statistical methods; and
- *EpiFast* [7] representing causal models.

AdapLSTM, LSTM, ARGO, and SARIMA can make flat-resolution forecasting directly from the model, then flat-resolution forecasts can be turned into high-resolution forecasts by multiplying by county-level population proportions. EpiFast is applied for both flat-resolution and high-resolution forecasting directly.

5.3 Experiment Setup

In this section, we describe the experiment settings, including simulation setting and TDEFSI model setting. Note that we conduct the experiments on two states of the USA, i.e., VA and NJ. State-level forecasting performance will be evaluated on both VA and NJ, while county-level forecasting performance is evaluated on NJ only due to the limitation on the availability of high-resolution observations.

Disease model settings for generating simulated training data. The simulation parameter settings are listed in Table 1. The length of a simulated epicurve is set to $\ell = 52$, and the total runs of simulations is $r = 1{,}000$. We adopt EpiFast as the simulator, PS='EpiFast'. More details on parameter space learning are described in Section A.3.

TDEFSI model settings. We set up the architectures for TDEFSI and its variants as follows:

- ***TDEFSI***: The left branch consists of two stacked LSTM layers, one dense layer; the right branch consists of one LSTM layer, one dense layer. $k_l = 2, k_r = 1, H^{(k_l)} = H^{(k_r)} = 128, H = 256, \psi_l, \psi_r, \psi$ are linear functions.
- ***TDEFSI-LONLY***: The left branch consists of two stacked LSTM layers, one dense layer and no right branch. $k_l = 2, H^{(k_l)} = 128, H = 256, \psi_l, \psi$ are linear functions.
- ***TDEFSI-RDENSE***: The left branch consists of two stacked LSTM layers, one dense layer; the right branch consists of one dense layer. $k_l = 2, k_r = 0, H^{(k_l)} = H^{(k_r)} = 128, H = 256, \psi_l, \psi_r, \psi$ are linear functions.

For all TDEFSI models, we set $a = 52$, $b = 5$, $(\mu, \lambda)_{VA} = (0.1, 0.1)$, $(\mu, \lambda)_{NJ} = (1, 0.01)$. We use Adam optimizer with all default values. We choose the final model using grid searching with sim-validating dataset. The grid searching space is about 500 models, including $a(10, 20, 30, 40, 50)$,

$b(5)$, $\mu(0, 0.001, 0.01, 0.1, 1)$, $\lambda(0, 0.001, 0.01, 0.1, 1)$, $k_l(1, 2)$, $H(128, 256)$. In the training process, the best models are selected by early stopping when the validation accuracy does not increase for 50 consecutive epochs, and the maximum epoch number is 300. Unless explicitly noted, in our experiments, these hyperparameters are set with the values described above. The settings of comparison methods are elaborated in Appendix A.4.

Our experiments are conducted on two testing datasets: (i) synthetic testing dataset and (ii) real seasonal ILI dataset.

Experimental setup for testing on simulated dataset. We make predictions for 10 weeks ahead, i.e., $horizon = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. Only TDEFSI is tested and analyzed using sim-testing dataset. No comparison methods are applied, since there is no surrogate information corresponding to the simulated seasons.

Experimental setup for testing on real seasonal ILI dataset. In these experiments, we evaluate TDEFSI models and all comparison methods. The experiments are performed on two states: Virginia (VA) and New Jersey (NJ). The county-level evaluation is conducted on NJ counties. For TDEFSI and its variants, the real-training dataset is used to estimate disease parameter space, while for all baselines, real-training and real-validating are used for training directly. The county-level real-evaluating dataset is only used for evaluation of the performance of county-level predictions. At each time step in the testing season, each model makes predictions up to five weeks ahead, i.e., $horizon = \{1, 2, 3, 4, 5\}$.

5.4 Performance Metrics

The metrics used to evaluate the forecasting performance are *root mean squared error (RMSE)*, *mean absolute percentage error (MAPE)*, and *Pearson correlation (PCORR)*.

- **Root mean squared error (RMSE)**:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2} \tag{9}$$

- **Mean absolute percentage error (MAPE)**:

$$MAPE = \left(\frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y_i}}{y_i + 1}\right|\right) * 100, \tag{10}$$

  where the denominator is smoothed by 1 to avoid zero values.
- **Pearson correlation (PCORR)**:

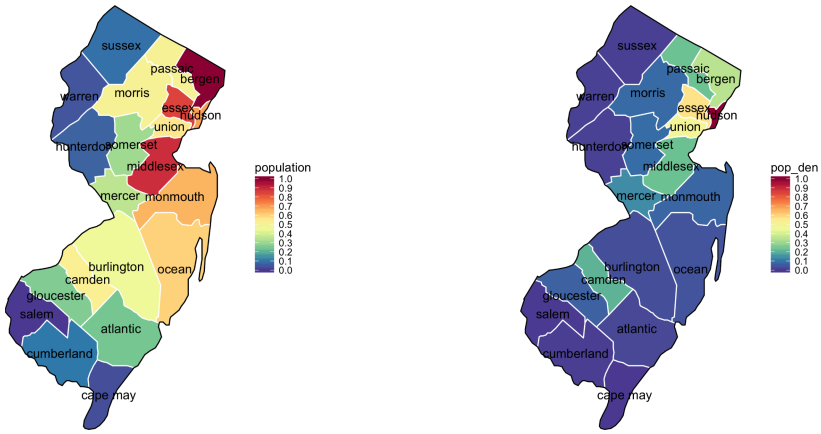
$$PCORR = \frac{cov(\mathbf{y}, \hat{\mathbf{y}})}{\sigma_{\mathbf{y}}\sigma_{\hat{\mathbf{y}}}}, \tag{11}$$

  where $cov(\mathbf{y}, \hat{\mathbf{y}})$ is the covariance of $\mathbf{y}$ and $\hat{\mathbf{y}}$, and σ is the standard deviation.

Among these metrics, RMSE and MAPE evaluate ILI incidence prediction accuracy, PCORR evaluates linear correlation between the true curve and the predicted curve.

5.5 Exploratory Analysis of Spatial Dynamics of NJ County Level Dataset

The spatiotemporal spread of influenza in a state depends on the social demographic attributes (e.g., population density) of the counties as well as the individual behavior and movement between counties. In this subsection we explore the county demographic data and between county commute data using visualization, and discuss their association with the disease spread spatially over time.



(a) Population of NJ counties, 2010 (b) Population density of NJ counties, 2010

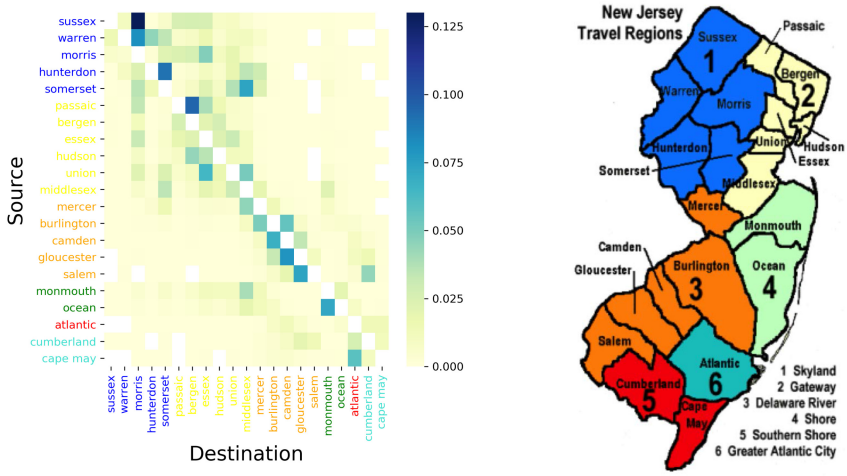
Fig. 8. Statistics of NJ counties [43]. (a) Population of NJ counties, 2020. (b) Population density of NJ counties, 2010. The population density is the population per square mile. The values shown in the map of both statistics are normalized by $(\frac{x-min}{max-min})$ so that the range is $[0, 1]$. The counties located in the eastern NJ have large population size, and the counties around the northeastern area are of especially high population density.

In Figure 8, we show the statistics for NJ counties including population and population density (i.e., population per square mile). Values are normalized by using $(\frac{x-min}{max-min})$ so that the range is $[0, 1]$. In general the counties located in northwestern NJ and southwestern NJ have small population and population density, while the counties concentrated in northeastern NJ have large ones.

From county-to-county commute counts data from the American Community Survey (ACS) 2009–2013 [1], we extract commute counts of which both source and destination are NJ counties. In Figure 9, we show the adjacency matrix of commute flows with the counties of NJ arranged according to spatial neighborhood. The flow in the figure is the normalized commute counts by the population size of the source county. A larger value means a larger commute flow between the two counties. The figure shows larger commute flows between counties that are physically close to each other. Nevertheless, there is substantial flow between counties that are far away from each other—this small-world-like flow is a hall-mark of human mobility patterns. During an epidemic, counties with large populations and high connectivity serve as hubs—these counties often start the epidemic early and also aid the spread to other counties.

In Figure 10 we visualize the correlation between county demographic attributes (population size and density) and county epidemic features (peak timing and peak intensity) in the ground-truth data. While counties with larger populations or higher population density seem to peak later in the season, this is not always true: There are small, low density counties that peak late. But there is no high-density county that peaks early. This suggests that the spatial features, e.g., the conventional geographic distance or *effective distance* (defined based on the commute flow matrix) [15] to the source county (where the epidemic starts), may play an important role in determining the disease spread trajectory among the counties of the state.

In Figure 11 we show the change of the ILI case numbers of New Jersey counties through the weeks **sw** of the 2017–2018 influenza season, at weeks 10, 13, 18, 21, and 25. For this season, one can note that the flu starts to spread rapidly in the east part of the state where the counties have large populations. Interested readers can find a week-by-week animation in Reference [88]. The spreading process shows spatial heterogeneity over the counties and is correlated to the population size and commute flow.



(a) Adjacency matrix of commute flows

(b) New Jersey county regions map

Fig. 9. (a) Adjacency matrix of commute flows with the counties of NJ arranged according to spatial neighborhood. We use the Regions of New Jersey as defined by the New Jersey State Department of Tourism (color of the labels match the map). Note that the visualization shows the normalized commute counts by county population size. A larger value means a larger commute flow between the two counties. The self-loop flows are not shown, since they're an order of magnitude larger than the rest. (b) New Jersey county regions map (downloaded from Reference [78]).

5.6 Performance on Simulated Testing Dataset

In this experiment, we tested TDEFSI on sim-testing dataset for VA. We set $\lambda = \mu = 0$, and set $b = 5$, and then conduct sensitivity analysis on the length of within-season observations, i.e., $a = \{10, 20, 30, 40, 52\}$. Figure 12 shows the state-level forecasting curves (partial curves of sim-testing) in horizon 1, 5, 10 using various a. The black curve is the ground truth, while the other colors correspond to different a values. By comparing across (a), (b), (c), we find that the predictive power of the model weakens as the horizon increases. In addition, the model with a seasonal length of $a = 52$ performs the best.

To verify our observations, we evaluate the model performances with metrics RMSE, MAPE, and PCORR at both the state level (shown in Figure 13) and county level (shown in Figure 14). The best model is always the one with $a = 52$. This is not random. It is the manifestation of a flu season normally consisting of 52/53 weeks, i.e., the seasonality of the time-series data. Thus, we suggest setting a to a multiple of 52 in practice. Unless explicitly stated, we fix $a = 52, b = 5$ in the rest of our experiments.

5.7 Performance on Real Seasonal ILI Testing Dataset

5.7.1 Performance of Flat-resolution Forecasting. We forecast state-level ILI incidence on real-testing dataset for VA and NJ. Table 2 shows the performance on RMSE, MAPE, and PCORR for (a) VA and (b) NJ with horizon=$\{1, 2, 3, 4, 5\}$. Figure 15 presents the overall performance across two states, two seasons, and five horizons. (i) *Performance on RMSE*: In VA, TDEFSI, TDEFSI-LONLY, TDEFSI-RDENSE, SARIMA, ARGO, and LSTM achieve similar performance that is better than EpiFast and AdapLSTM. Compared with other methods, AdapLSTM does not perform well with small horizons while EpiFast has poor performance with large horizons. In NJ, TDEFSI, TDEFSI-LONLY,

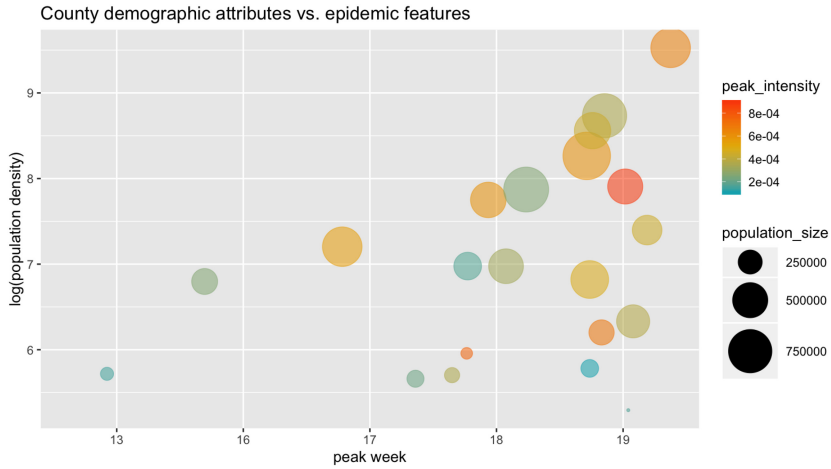
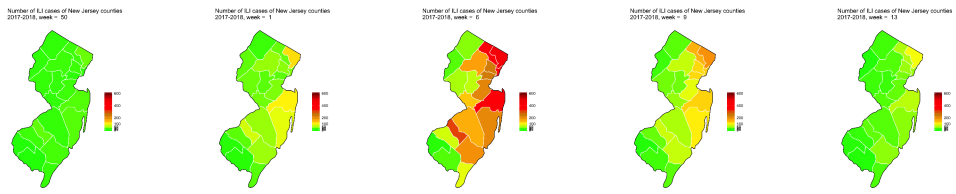
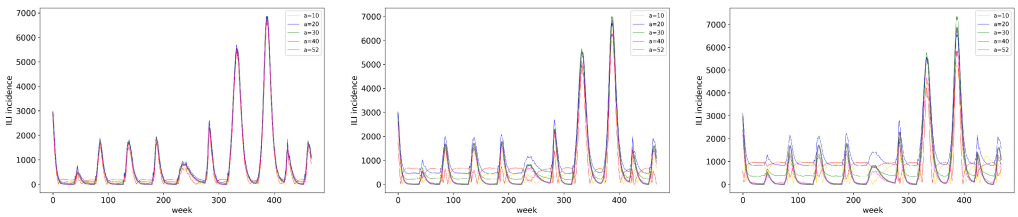


Fig. 10. The correlation between the peak (including peak week and peak intensity) and the population density of NJ counties. The x-axis denotes peak week ordered by sw, and the y-axis represents log value of the population density. The bubble color and size denote peak intensity and population size. The peak week and peak intensity are only partially correlated with county population size/density.



(a) Week 10 (b) Week 13 (c) Week 18 (d) Week 21 (e) Week 25

Fig. 11. The number of ILI cases of New Jersey counties, season 2017–2018. (a) Week 10. (b) Week 13. (c) Week 18. (d) Week 21. (e) Week 25. Note that the flu starts to spread rapidly in regions 2, 3, 4, and 6 that have counties with large population.



(a) horizon 1 (b) horizon 5 (c) horizon 10

Fig. 12. State-level forecasting curves on sim-testing dataset with (a) horizon 1, (b) horizon 5, and (c) horizon 10. The x-axis is the week number of 10 simulated curves. Various settings of a are compared. The black curve is the ground truth, while the other colors correspond to models with different values of a. It is observable that the predictive power of the model weakens as the horizon increases. The model (magenta curve) with $a = 52$ performs the best.

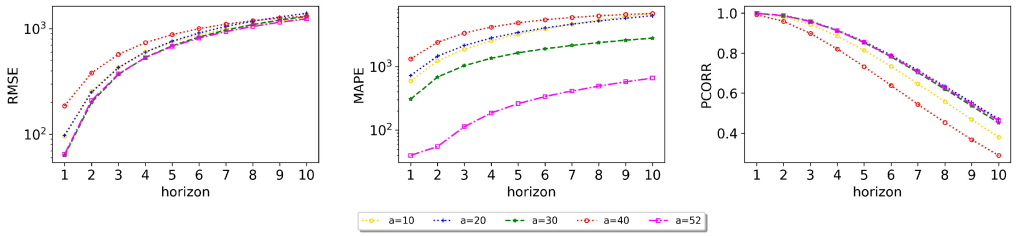


Fig. 13. State-level forecasting performance on sim-testing dataset of VA with various length of within-season observations $a = \{10, 20, 30, 40, 52\}$, which is evaluated by RMSE (left), MAPE (middle), and PCORR (right). The x-axis represents horizons from 1 to 10. The value is averaged on all weeks of testing curves. A log y-scale is used in RMSE and MAPE. Across different horizons and metrics, the best model is always the model with $a = 52$.

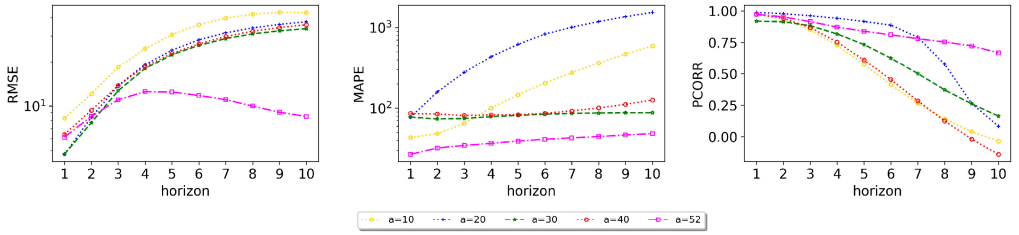


Fig. 14. County-level forecasting performance on sim-testing dataset of VA with various length of within-season observations $a = \{10, 20, 30, 40, 52\}$, which is evaluated by RMSE (left), MAPE (middle), and PCORR (right). The x-axis represents horizons from 1 to 10. The value is averaged on all weeks of testing curves. A log y-scale is used in RMSE and MAPE. Across different horizons and metrics, the best model is always the model with $a = 52$, especially with larger horizons.

and TDEFSI-RDENSE consistently outperform others across the horizon. Overall, TDEFSI and its variants slightly outperform comparison methods in RMSE. (ii) *Performance on MAPE*: In VA, SARIMA performs the best overall among all methods. In NJ, TDEFSI-RDENSE achieves the best performance closely followed by SARIMA. Overall, SARIMA outperforms others, and TDEFSI and its variants achieve similar performance with ARGO, which are better than LSTM, AdapLSTM, and EpiFast. (iii) *Performance on PCORR*: In VA, ARGO performs the best with horizon 1,2,3 and TDEFSI achieves better performance with horizon 4,5. In NJ, TDEFSI performs the best and TDEFSI-LONLY, TDEFSI-RDENSE achieve similar performance. Overall, TDEFSI and its variants slightly outperform SARIMA, ARGO, and LSTM, while they are much better than AdapLSTM and EpiFast.

Figure 16 shows the *weekly* state-level model performance measured on season 2017–2018 using RMSE: The x-axis denotes *ew* number, the value is averaged over five horizons. A log y-scale is used. The black vertical line marks the peak week of the season. We observe that these models perform with great variance around the beginning and the end of a season than in weeks near the peak.

The above discussion can be summarized as follows:

- Our TDEFSI and its variants achieve comparable/better performance than the other methods on the state-level ILI forecasting.
- EpiFast and AdapLSTM perform relatively worse than other methods in our experiments.

5.7.2 Performance of High-resolution Forecasting. The performance of county-level forecasts is evaluated on NJ counties. Note that EpiFast, TDEFSI, TDEFSI-LONLY, and TDEFSI-RDENSE make

Table 2. State Level Performance across Season 2016–2017 and 2017–2018 for VA and NJ with Horizon = 1, 2, 3, 4, 5

	VA					NJ				
RMSE	1	2	3	4	5	1	2	3	4	5
SARIMA	**824**	1463	2059	2440	2682	218	464	690	891	1050
ARGO	1073	1592	2072	2444	2580	313	512	717	760	874
LSTM	1083	1629	**2013**	2273	**2438**	240	470	699	902	1070
AdapLSTM	2012	2038	2264	2382	2449	586	729	640	871	1006
EpiFast	1300	2087	2989	3674	4284	238	382	567	725	871
TDEFSI	1000	**1447**	2014	**2358**	2544	**174**	**344**	511	665	757
TDEFSI-LONLY	900	1572	2119	2582	2742	197	373	531	696	801
TDEFSI-RDENSE	1109	1686	2136	2421	2540	193	358	**506**	**630**	**711**
MAPE	1	2	3	4	5	1	2	3	4	5
SARIMA	**15.96**	**32.57**	**50.62**	**65.60**	77.94	**13.28**	24.32	35.62	48.32	59.99
ARGO	31.06	54.00	73.69	78.97	77.85	24.96	33.14	44.52	50.05	54.60
LSTM	38.40	49.29	58.80	67.98	71.00	39.44	78.53	131.19	189.79	243.40
AdapLSTM	42.67	51.22	61.02	67.33	**70.60**	64.30	64.77	65.56	74.14	76.50
EpiFast	31.14	53.45	84.32	124.05	167.44	30.32	32.40	50.75	64.61	76.27
TDEFSI	25.75	40.69	58.61	74.06	88.95	18.16	29.74	43.49	55.12	66.09
TDEFSI-LONLY	22.40	35.18	59.27	89.95	123.70	15.56	32.21	45.74	60.46	72.13
TDEFSI-RDENSE	31.89	51.69	76.94	101.38	125.23	15.17	**21.74**	**29.19**	**37.95**	**44.14**
PCORR	1	2	3	4	5	1	2	3	4	5
SARIMA	0.9461	0.8271	0.6468	0.4925	0.3788	0.9541	0.8173	0.6421	0.4611	0.3195
ARGO	**0.9590**	0.8728	**0.7219**	0.4518	0.3218	0.9444	0.8005	0.6043	0.4530	0.2921
LSTM	0.9223	0.7890	0.6350	0.5050	0.4101	0.9603	0.8542	0.6995	0.5340	0.3939
AdapLSTM	0.7048	0.6397	0.5174	0.4307	0.3818	0.8113	0.5912	**0.7686**	0.4477	0.2753
EpiFast	0.8876	0.7665	0.5616	0.3906	0.2340	0.9573	0.8535	0.7044	0.3835	0.2841
TDEFSI	0.9358	0.8487	0.6892	**0.5555**	**0.4647**	**0.9683**	**0.8773**	0.7348	**0.5639**	0.4247
TDEFSI-LONLY	0.9460	**0.8776**	0.7037	0.5074	0.3266	0.9659	0.8697	0.7288	0.4946	0.3245
TDEFSI-RDENSE	0.9043	0.7824	0.6182	0.4409	0.2826	0.9654	0.8692	0.7280	0.5630	**0.4248**

The best value is marked in bold, and the second best value is marked with underline.

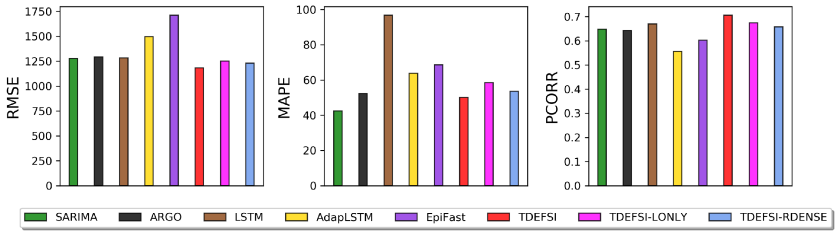
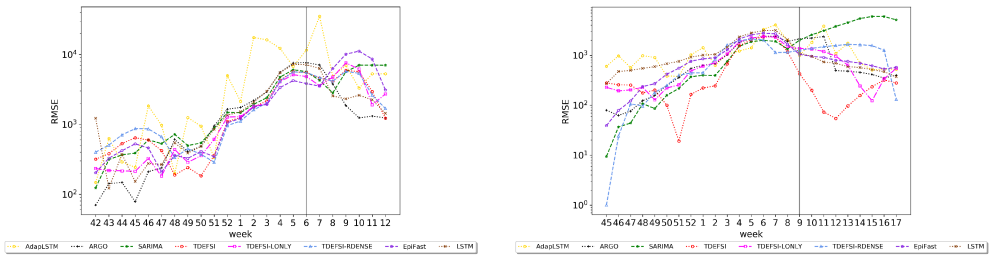


Fig. 15. State-level performance (RMSE, MAPE, PCORR). The value is averaged across two states, two seasons, and five horizons.

county-level predictions directly from models, while the other baselines obtain county-level predictions by multiplying state-level prediction with county population proportions. Table 3 shows the forecasting performance on RMSE, MAPE, PCORR with horizon={1, 2, 3, 4, 5}. The value is the average across weeks and counties. Figure 17 presents the overall performance across all counties, weeks, horizons. From the table we observe that SARIMA performs well with horizon = 1. TDEFSI



(a) VA, 2017-2018 (b) NJ, 2017-2018

Fig. 16. State-level performance by weeks (RMSE). (a) VA, 2017–2018; (b) NJ, 2017–2018. TDEFSI and its variants, and all comparison methods are evaluated and compared. The x-axis denotes ew number, the value is averaged on five horizons. A log y-scale is used. The black vertical line marks the peak week of the season in the state.

Table 3. County Level Performance for Counties of NJ with Horizon = 1, 2, 3, 4, 5

	NJ-Counties				
RMSE	1	2	3	4	5
SARIMA	**30.58**	38.02	48.60	58.92	67.68
ARGO	33.69	39.89	49.61	51.46	57.35
LSTM	33.80	41.95	52.25	61.56	68.30
AdapLSTM	36.67	45.30	39.46	51.70	59.60
EpiFast	34.34	36.74	40.51	47.40	54.09
TDEFSI	35.17	**31.40**	**34.70**	**40.44**	**45.95**
TDEFSI-LONLY	33.13	36.45	42.41	50.63	56.22
TDEFSI-RDENSE	34.79	31.59	35.22	40.98	46.35
MAPE	1	2	3	4	5
SARIMA	575.19	550.74	540.04	525.20	525.57
ARGO	649.32	552.18	498.42	430.74	366.89
LSTM	745.52	876.56	1066.80	1264.64	1417.91
AdapLSTM	584.18	489.51	417.72	599.53	717.61
EpiFast	712.97	632.96	577.74	519.37	487.54
TDEFSI	**260.95**	**247.70**	**209.69**	**270.58**	**308.95**
TDEFSI-LONLY	603.33	528.62	478.08	454.52	435.50
TDEFSI-RDENSE	614.95	499.13	412.68	360.99	315.78
PCORR	1	2	3	4	5
SARIMA	**0.8645**	0.7474	0.5678	0.3806	0.2211
ARGO	0.8606	0.7388	0.5455	0.3922	0.2211
LSTM	0.8611	0.7699	0.6132	0.4234	0.2597
AdapLSTM	0.7260	0.5150	0.6717	0.3710	0.2205
EpiFast	0.8555	0.7762	0.6450	0.3530	0.2133
TDEFSI	0.7877	**0.8500**	**0.7835**	**0.6425**	**0.4710**
TDEFSI-LONLY	0.8499	0.7669	0.6184	0.4146	0.2176
TDEFSI-RDENSE	0.7860	0.8063	0.7056	0.5467	0.3774

The value is the average of 21 counties of NJ across season 2016–2017 and 2017–2018. The best value is marked in bold, and the second best value is marked with underline.

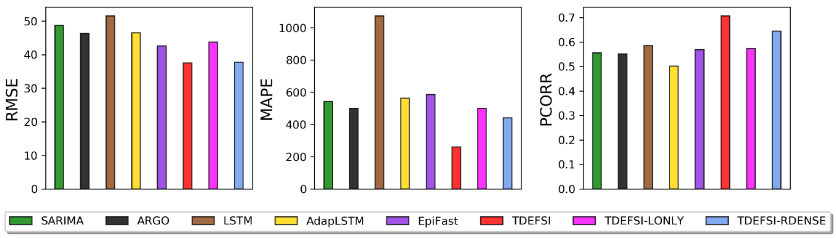
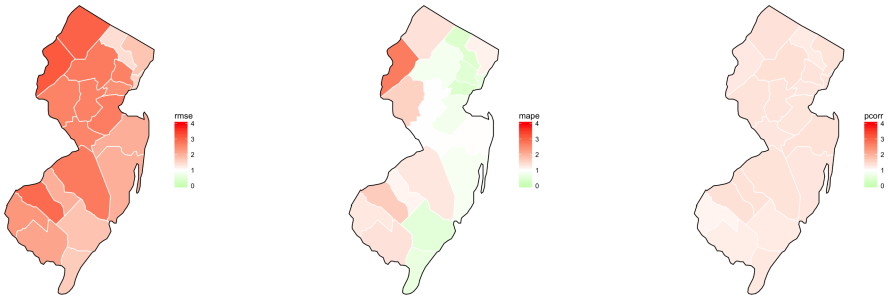


Fig. 17. County-level performance (RMSE, MAPE, PCORR). The value is averaged on two seasons, five horizons, and 21 counties of NJ.



(a) NJ counties, RMSE-ratio (b) NJ counties, MAPE-ratio (c) NJ counties, PCORR-ratio

Fig. 18. Comparison of the county-level spatial forecasting performance between TDEFSI and EpiFast for NJ, season 2017–2018. (a) RMSE-ratio; (b) MAPE-ratio; (c) PCORR-ratio. For each county in NJ, the ratio value of the county is computed using Equation (12), which is the average value across horizons. A value larger than 1 (red) means TDEFSI outperforms EpiFast, a value equal to 1 (white) means they both perform equally, and a value smaller than 1 (green) means EpiFast performs better than TDEFSI. The absolute magnitude of the value denotes the significance of the difference of the two models' performance. The comparison results exhibit that TDEFSI performs better than EpiFast in the counties located in western NJ.

consistently outperforms others across horizons, followed by TDEFSI-RDENSE. Among TDEFSI variants, TDEFSI and TDEFSI-RDENSE perform better than TDEFSI-LONLY, which indicates that the between-season observations are helpful for improving forecasting accuracy. The figure shows consistent results with the table. Overall, our method outperforms the comparison methods on the county-level forecasting.

Heterogeneous high-resolution forecasting. To better understand the results from a spatial perspective, we compare results between TDEFSI and EpiFast in Figure 18. The reason we choose to compare these two methods is that they both can make high-resolution predictions directly from the models. For each county in NJ, we compare TDEFSI and EpiFast using a ratio value for each of three metrics defined as:

$$
\begin{aligned}
RMSE-ratio &= \frac{\frac{1}{m}\sum_{i=1}^{m} RMSE_i(EpiFast)}{\frac{1}{m}\sum_{i=1}^{m} RMSE_i(TDEFSI)} \\
MAPE-ratio &= \frac{\frac{1}{m}\sum_{i=1}^{m} MAPE_i(EpiFast)}{\frac{1}{m}\sum_{i=1}^{m} MAPE_i(TDEFSI)} \\
PCORR-ratio &= \frac{\frac{1}{m}\sum_{i=1}^{m} (PCORR_i(TDEFSI)+1)}{\frac{1}{m}\sum_{i=1}^{m} (PCORR_i(EpiFast)+1)},
\end{aligned}
\tag{12}
$$

where m is the number of horizons. The ratio is averaged across all horizons. For any of these ratios, a value larger than 1 means TDEFSI outperforms EpiFast; a value close to 1 means they have similar performance; and a value smaller than 1 means EpiFast performs better than TDEFSI.

From Figure 18(a) RMSE-ratio, we observe that TDEFSI significantly outperforms EpiFast in all counties (all counties show red colors) especially in the western counties of NJ. In (b) MAPE-ratio, TDEFSI performs better than EpiFast in 11 of 21 counties, most of which are located in the west side of NJ. And (c) PCORR-ratio shows that TDEFSI constantly outperforms EpiFast in all counties (all in red colors). The comparison results exhibit that TDEFSI performs better than EpiFast in the counties located in western NJ. EpiFast tries to find a model that best matches the state-level observations, and use it to make predictions. However, the identified model is usually locally optimal due to the limitation of the searching algorithm and the computational efficiency. In our experiments, we run the searching algorithm once and then find a locally optimal model that performs fairly well in eastern NJ counties but not in western NJ counties. If we run the searching algorithm again, then we will find another locally optimal model that might perform well in western NJ counties instead. In TDEFSI model, the deep neural network model allows TDEFSI to learn from many models. What is learned is an ensemble of all models. Thus, TDEFSI is more robust than EpiFast in different runs of the flu forecasting experiment.

5.7.3 Discussion. In general, for state level, AdapLSTM and EpiFast do not perform very well in our experiments compared with other methods. For AdapLSTM, weather features are considered for post adjustment of LSTM outputs. As stated in Reference [82], the weather factors are estimated using time delays computed by *a priori* associations and selected by the largest confidence. However, in our experiment, they all show very low confidences (less than 0.3). This may cause arbitrary adjustment for predictions and consequently poor performance. For EpiFast, one possible reason is that we did not find a good estimate of the underlying disease model for a specific region and season due to the noisy CDC observations. If we rank the performance of all methods, then ARGO performs slightly better on VA than on NJ. The possible reason is that about 80% of the top 100 Google correlated terms for NJ are irrelevant to flu and most of them have zero frequencies, while the top 100 correlated terms for VA are of good quality. This will give ARGO a better performance on VA than on NJ. Similarly, LSTM performs relatively better on VA than on NJ. One possible reason is that LSTM cannot learn a pattern that has never occurred in the historical observations. So its performance depends on whether a similar epicurve occurred in previous seasons. As shown in Figure 19, the epicurve of VA 2017–2018 is similar to that of VA 2014–2015, and 2016–2017 is similar to 2012–2013. However, the epicurve of NJ 2017–2018 seems to be much higher than all previous ones, as well as 2016–2017. Actually, this is the limitation of all data-driven models. On the contrary, TDEFSI models have stable performance on both VA and NJ. They manage to avoid overfitting through training on a large volume of synthetic training data. In addition, the simulated training dataset includes many realistic simulated patterns that are unseen in the real world, thus provides a better generalizability to our models.

As seen through the results, TDEFSI enables high-resolution forecasting that outperforms baselines. Meanwhile, it achieves comparable/better performance than the comparison methods at state-level forecasting. And in our framework, the large volume of realistic simulated data allows us to train a more complex DNN model and reduces the risk of overfitting. Our experiments demonstrate that TDEFSI integrates the strengths of ANN methods and causal methods to improve epidemic forecasting.

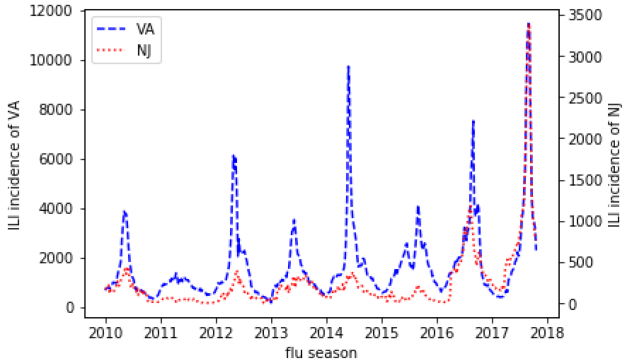


Fig. 19. CDC surveillance ILI incidence of VA (blue dash line) and NJ (red dot line). It is observable that, for testing season 2017–2018, a similar epi-curve (i.e., similar curve shape and the peak size) occurs at season 2014–2015 in VA, while no similar seasons could be found in NJ.

Table 4. Hyperparameters of TDEFSI Model and Their Values for Sensitivity Analysis

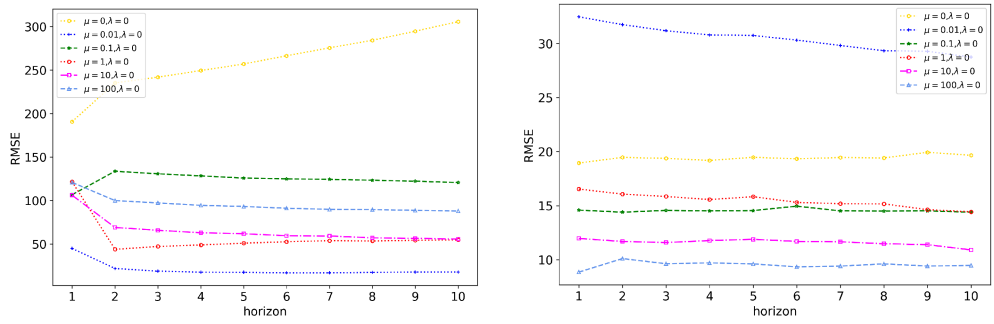
Parameters	Description	Values
a	length of within-season observations	10, 20, 30, 40, 52
b	length of between-season observations	5
λ	coefficient of spatial regularizer	0, 0.001, 0.01, 0.1, 1, 10, 100
μ	coefficient of non-negative regularizer	0, 0.001, 0.01, 0.1, 1, 10, 100

There are many hyperparameters in TDEFSI models, such as input dimension a, b, consistency coefficiency μ, λ, number of hidden layers k_r, k_l, number of hidden units $H^{(k_l)}$, $H^{(k_r)}$, H, learning rate, training epoch, and so on. In our experiments, we choose the final model by using grid searching on the hyperparameters using sim-validating dataset. In the training process, the best models are selected by early stopping when the validation accuracy does not increase for 50 consecutive epochs, and the maximum epoch number is 300.

5.8 Physical Consistency Constraints

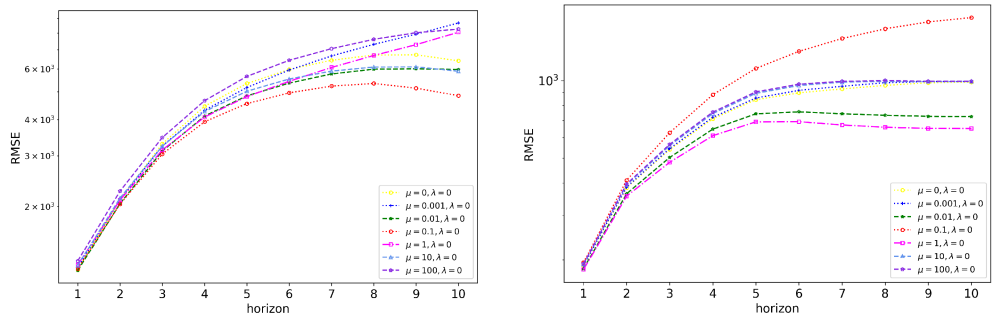
In this section, we conduct sensitivity analysis on two regularizer coefficients μ and λ in Equation (6), which control the weights of the spatial constraint ϕ and non-negative constraint δ in the loss function. $\mu = 0$ means no spatial constraint and $\lambda = 0$ means no non-negative constraint. We train TDEFSI by setting $a = 52, b = 5$ with various μ, λ values shown in Table 4. We then use the trained models to make predictions for Season 2017–2018 of VA and NJ. The performance is evaluated using RMSE.

Spatial consistency. The experiments are conducted using $\lambda = 0$ and $\mu = \{0, 0.001, 0.01, 0.1, 1, 10, 100\}$. We evaluate the spatial consistency by computing RMSE of the predicted state-level ILI incidence and the summation of the predicted county-level ILI incidence, i.e., $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - \sum_{C\in\mathcal{D}}\hat{y}_i^C)^2}$. Figure 20 shows the spatial consistency error measured by RMSE on (a) VA, 2017-2018 and (b) NJ, 2017–2018. The results show that the spatial consistency error does not vary much with horizon, but significantly depends on μ. The possible reason is that, in the TDEFSI model, the input is only state-level data, so the LSTM layers learn the temporal pattern on state-level time sequence that closely relates to model performance with horizons. However, spatial information is not propagated along the cells during training, but only compounds in the last step of outputs, thus is not impacted by horizons. The optimal μ differs between states. The results indicate that TDEFSI enables the spatial consistency with a proper μ value. However, a better



(a) VA, 2017-2018 (b) NJ, 2017-2018

Fig. 20. Spatial consistency error (computed as $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - \sum_{C\in\mathcal{D}}\hat{y}_i^C)^2}$) on (a) VA, 2017–2018; (b) NJ, 2017–2018. The coefficient of the spatial consistency regularizer is set to $\mu = \{0, 0.001, 0.01, 0.1, 1, 10, 100\}$. The results show that the spatial consistency error does not vary much with horizon but significantly depends on μ. The optimal μ differs between states.



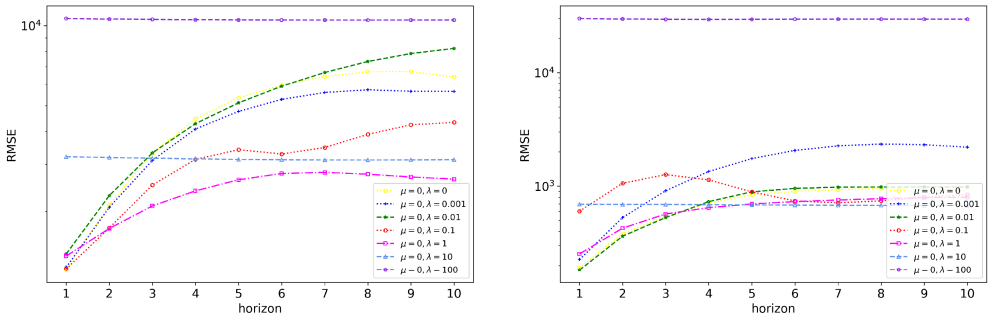
(a) VA, 2017-2018 (b) NJ, 2017-2018

Fig. 21. TDEFSI performance with spatial consistency constraints of different coefficients $\mu = \{0, 0.001, 0.01, 0.1, 1, 10, 100\}$. The performance is evaluated on (a) VA 2017–2018 season and (b) NJ 2017–2018 season. The results show that the coefficient μ has significant influence on the model forecasting performance especially with large horizons. The optimal value of μ should be chosen independently in different regions. A log y-scale is used in RMSE and MAPE.

spatial consistency does not mean a better model forecasting performance. In practice, we need to keep balance between keeping good spatial consistency and maintaining good model performance.

To evaluate the significance of the spatial consistency constraint for model forecasting power, we compare the forecasting performance of models on real seasonal data with various μ using RMSE (shown in Figure 21). For VA, the best performance is the model with $\mu = 0.1$. For NJ, the best performance is the model with $\mu = 1$. Overall, the spatial consistency constraint with a proper coefficient, which may vary between different regions, helps improve the forecasting performance.

Non-negative consistency. The experiments are conducted using $\mu = 0$ and $\lambda = \{0, 0.001, 0.01, 0.1, 1, 10, 100\}$. Similarly to the spatial consistency evaluation, we compare the performance of models with various λ using RMSE (shown in Figure 22). For VA, the best performance is the model with $\lambda = 1$, and the models with the non-negative consistency constraint ($\lambda \leq 1$) outperform the model without the constraint. For NJ, the best performance is the model with $\lambda = 1$. For both VA



(a) VA, 2017-2018 (b) NJ, 2017-2018

Fig. 22. TDEFSI performance with non-negative consistency constraints of different coefficients $\lambda = \{0, 0.001, 0.01, 0.1, 1, 10, 100\}$. The performance is evaluated on (a) VA 2017–2018 season; (b) NJ 2017–2018 season. The results show that the coefficient λ has significant influence on the model forecasting performance. The optimal value of λ should be chosen independently in different regions. A log y-scale is used in RMSE.

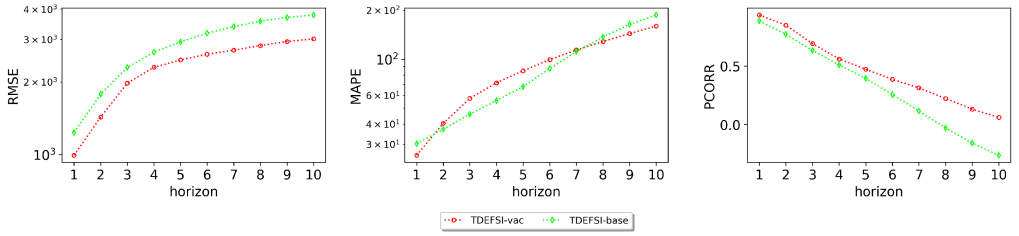
and NJ, from the figures we observe that the models with λ equal or larger than 10 will have no predicting power (i.e., they are almost horizontal lines with high RMSE). The possible reason is that a strong penalty (large λ) may cause the weights of the hidden units to shrink toward zero. When $\mathbf{W}, \mathbf{U}$ in Equation (1) become zero the LSTM layer gives a constant output. This will make the network stop learning and output constant predictions. Overall, the non-negative consistency constraint with a proper coefficient, which may vary between different regions, helps improve the forecasting performance.

Implications. Three types of physical consistency were incorporated in our TDEFSI models. Computational experiments show that these constraints can lead to a better domain consistency as well as improve the forecasting performance. By incorporating physical consistency, TDEFSI enables theory guided deep learning for epidemic forecasting. Spatial and non-negative consistency constraints also positively influence the overall performance. However we note that no single parameter setting works across all scenarios thus context specific tuning is needed.

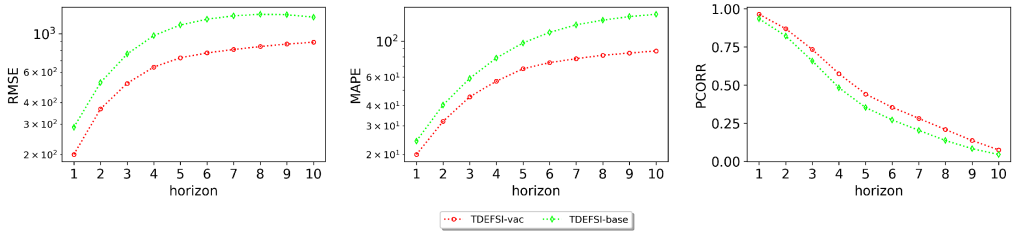
5.9 Vaccination-based Interventions

When TDEFSI framework uses an agent-based SEIR model to generate a simulated training dataset, it is straightforward to implement various interventions in the simulations. For example, in our parameter space $\mathcal{P}(p_E, p_I, \tau, N_I, I_V)$, I_V represents the vaccination-based intervention. We investigate how I_V affects the performance of TDEFSI by generating two synthetic training datasets: (i) *vaccine-case*: Generated by simulations with I_V (TDEFSI and its variants in previous experiments of Section 5 are trained on vaccine-case simulated training dataset); and (ii) *base-case*: Generated by simulations that share the common settings of p_E, p_I, τ, N_I with vaccine-case except $I_V = \emptyset$. We train TDEFSI on the vaccine-case and base-case with the same settings described in Section 5.3, and denote the trained models as *TDEFSI-vac* and *TDEFSI-base*, respectively. Note that here TDEFSI-vac is the same as TDEFSI in the previous experiments.

Figure 23(a) and Figure 23(b) show the state-level forecasting performance of VA and NJ on RMSE, MAPE, and PCORR using real-testing dataset. We observe that TDEFSI-vac significantly outperforms TDEFSI-base for all metrics on both states except that for the MAPE result of VA, TDEFSI-vac is compatible with TDEFSI-base. In Figure 24, we present the comparison ratio between two models from the spatial dimension of NJ counties. It is observable that TDEFSI-vac

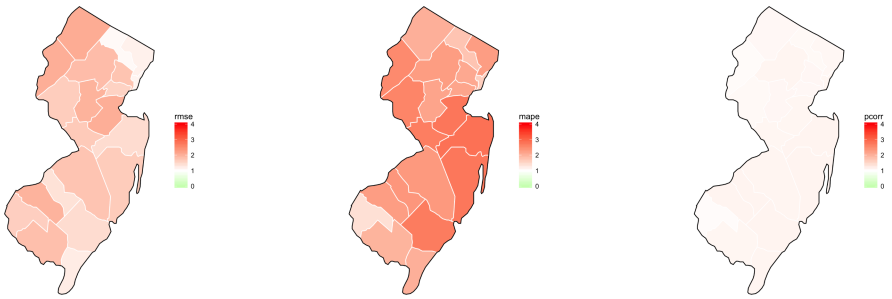


(a) VA, 2017-2018



(b) NJ, 2017-2018

Fig. 23. State-level forecasting performance comparison between TDEFSI models trained on the base-case simulated training dataset (TDEFSI-base) and the vaccine-case simulated training dataset (TDEFSI-vac). They test on VA, 2017–2018 with a horizon up to 10 weeks ahead. TDEFSI-vac outperforms TDEFSI-base across three metrics. A log y-scale is used in RMSE and MAPE.



(a) NJ counties, RMSE-ratio (b) NJ counties, MAPE-ratio (c) NJ counties, PCORR-ratio

Fig. 24. NJ, 2017–2018 county-level spatial forecasting performance comparison between TDEFSI-vac and TDEFSI-base for NJ, season 2017–2018. (a) RMSE-ratio; (b) MAPE-ratio; (c) PCORR-ratio. For each county in NJ, the ratio value of the county is computed using Equations (12), which is the average value across horizons. A value larger than 1 (red) means TDEFSI-vac outperforms TDEFSI-base, a value equal to 1 (white) means they both perform equally, and a value smaller than 1 (green) means TDEFSI-base performs better than TDEFSI-vac. The absolute magnitude of the value denotes the significance of the difference of the two models' performance. It is observable that TDEFSI-vac performs better than TDEFSI-base in all counties of NJ.

performs better than TDEFSI-base in all counties of NJ. The results indicate that vaccination-based interventions applied in the simulations to generate training datasets can significantly improve the forecasting performance.

The models learned from the vaccine-case datasets are more generalizable to unseen surveillance data. Our experiments show the significance of vaccination-based interventions applied in

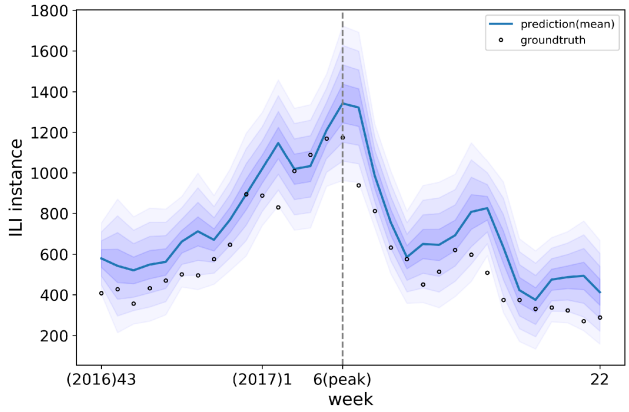


Fig. 25. NJ state-level mean predicted curve with predictive intervals of $(mean \pm k * std)$, where $k = \{0.5, 1, 1.5, 2\}$. The black circles are ground truths. We can observe that all ground truths are within 2 standard deviations.

the simulations on the forecasting performance. The proposed framework is extensible for other realistic interventions, such as school closure or antivirals, to further improve the forecasting performance.

5.10 Prediction Uncertainty Estimation

In the epidemic forecasting domain, probabilistic forecasting is important for capturing the uncertainty of the disease dynamics and to better support public health decision making. Probabilistic forecasting with deep learning models is challenging due to the lack of interpretability of such models. Most works on this are based on Bayesian Neural Networks. Gal et al. [35] in 2016 proved that using dropout technique is equivalent to Bayesian NN's and proposed Monte Carlo Dropout (MC Dropout) to estimate uncertainty in deep learning. The proposed method is computationally efficient. We implement MC Dropout in TDEFSI and demonstrate estimation of prediction uncertainty with a case study of state-level forecasting for NJ season 2016–2017. The model setting is the same as that described in Section 5.3, and the MC number is 20. Figure 25 shows the curve of mean predictions with predictive intervals of $(mean \pm k * std)$, where $k = \{0.5, 1, 1.5, 2\}$. We can observe that all ground truths are within 2 standard deviations.

6 CONCLUDING REMARKS AND DIRECTIONS FOR FUTURE WORK

We described TDEFSI—a novel epidemic forecasting framework that combines deep learning methods with high performance computing oriented simulations of epidemic processes over realistic social contact networks. TDEFSI and its variants use a two-branch LSTM-based neural network model and are designed to combine within-season and between-season observations. TDEFSI incorporates domain knowledge into deep neural network models by considering temporal, spatial, and non-negative consistency constraints as well as natural constraints imposed by the use of epidemic simulations.

The models are trained on a region-specific simulated dataset constructed at multiple spatially fine-grained scales. The trained models can provide high-resolution forecasts using flat-resolution surveillance data. We carried out extensive computational experiments on NJ and VA, using synthetic as well as state-level real surveillance data. The results show that TDEFSI combined with epidemic simulations achieve comparable/better performance than the state-of-the-art methods

for ILI forecasting at the state level. For high-resolution forecasting at the county level, TDEFSI significantly outperforms the comparison methods. Through sensitivity analysis on spatial and non-negative consistency constraints, we discuss the influence of these constraints on model performance. A case study of probabilistic forecasting on NJ state shows the model's ability to provide prediction uncertainty using MC Dropout technique. Experiments involving more states and more seasons are desirable to show that the performance comparison of TDEFSI against other methods is robust, but due to the limitation on the availability of high-resolution data and historical data of flu seasons we only tested the framework on two states and two seasons. In future work, we plan to look for more datasets so that the robustness of our observations can be tested.

Future work. A direction for future work is to investigate the use of synthetic data generated by social, epidemiological, and behavioral models in conjunction with observed data to improve epidemic forecasts. (i) In this work, we try to reduce the gap between simulated and real world data distributions by simulating with parameter settings learned from observations so that the generated epi-curves are realistic. In future work, we plan to further reduce the gap by using synthetic data based on real-time observations to train the neural networks. (ii) We also plan to explore the capability of TDEFSI on *what-if* forecasts. What-if forecasts capture various what-if scenarios due to expected or unexpected public health interventions or individual-level behavioral reactions as the epidemic evolves. They provide insights on possible trajectories of the ongoing epidemic under different assumptions. They can help public health decision making with risk/benefit predictions. The data-driven methods can only provide passive forecasts, while what-if forecasts are natural in TDEFSI thanks to the causal model behind it. A possible way to make what-if forecasts with TDEFSI works as follows: based on the current status of the epidemic, make a few assumptions about what may happen in the future that will change the epidemic dynamics; implement each assumption as a set of interventions (e.g., school closure from $ew(51)$ to $ew(52)$) in the simulations and generate synthetic epi-curves; re-train the deep neural network with the updated synthetic curves; and make predictions that describe future dynamics with this particular assumption. Note that one what-if scenario can be associated with multiple interventions.

A APPENDIX

A.1 Synthetic Social Contact Network

A synthetic population and the corresponding social contact network are used to simulate the spread of the disease. In our work, we use the synthetic social contact network of Virginia and New Jersey. Below we briefly describe the methodology used for constructing the synthetic population and the social network.[2] Interested readers can find more details about this methodology in References [6, 8, 14, 30, 64].

To construct the social network, first a statistical representation of each individual in the population is built using US Census data. This synthetic population is statistically equivalent to the real population when aggregated to the census block group level. Individuals in the synthetic population are assigned a complete range of demographic attributes as available in the Census [8, 13], including age, gender, household location, and household income.

Next, a set of activity templates are extracted from American time-use surveys [17] and the National Household Travel Survey. Each of these activity templates provides a daily sequence of activities for individuals and the time of day they are performed. Each synthetic household is matched with one of the survey households, using a decision tree based on demographics such as the size of the household, number of workers in the household, number of children, and so on.

[2]The description is similar as the one we described in our previous work [86], since they use the same synthetic dataset.

Table 5. Surveillance Ratios for Each State in the US

Alabama: 0.0759	Kansas: 0.1093	New York: 0.1204
Alaska: 0.1143	Kentucky: 0.1114	North Carolina: 0.0875
Arizona: 0.0723	Louisiana: 0.0931	North Dakota: 0.1960
Arkansas: 0.0894	Maine: 0.1931	Ohio: 0.1339
California: 0.0628	Maryland: 0.0755	Oklahoma: 0.1039
Colorado: 0.0764	Massachusetts: 0.1380	Oregon: 0.1050
Connecticut: 0.1047	Michigan: 0.1356	Pennsylvania: 0.1299
Delaware: 0.1030	Minnesota: 0.0898	Rhode Island: 0.0932
District of Columbia: 0.1852	Mississippi: 0.0874	South Carolina: 0.0663
Florida: 0.0582	Missouri: 0.1492	South Dakota: 0.1882
Georgia: 0.0701	Montana: 0.1739	Tennessee: 0.0811
Hawaii: 0.0705	Nebraska: 0.1329	Texas: 0.0738
Idaho: 0.1190	Nevada: 0.0643	Utah: 0.0913
Illinois: 0.1066	New Hampshire: 0.1566	Vermont: 0.2111
Indiana: 0.1215	New Jersey: 0.0692	Virginia: 0.0914
Iowa: 0.1420	New Mexico: 0.1258	Washington: 0.0885
Kansas: 0.1093	New York: 0.1204	West Virginia: 0.1684

The synthetic household members are then assigned the activity templates of the matching survey household members, giving each synthetic individual a daily sequence of activities. For each activity of each individual, a geographic location is identified based on land-use patterns, transportation network, and data from commercially available databases such as Dun and BradStreet.

A social network is constructed by connecting individuals simultaneously present at the same location. The co-location-based social network is dynamic and changes as people visit different locations and come in contact with individuals at these locations.

A.2 Surveillance Ratio

In our experiments, we scale the ILINet case count to the population case count using a surveillance ratio. We assume that the ratio between ILI cases captured by CDC ILINet (denoted ILITOTAL) and ILI cases in the population (ILIPOP) is the same as that between patients of all diseases captured by CDC ILINet (TOTALPATIENT) and patients of all diseases in the population (PATIENTPOP). We approximate PATIENTPOP with all doctor visit data from AHRQ [2]. The doctor visit data provides county-level counts for total hospital visits in a year that is aggregated to state-level counts later. Note that it is an underestimate. From surveillance ratio $= \frac{\text{ILITOTAL}}{\text{ILIPOP}} = \frac{\text{TOTALPATIENT}}{\text{PATIENTPOP}}$, we can derive the only unknown ILIPOP. Table 5 presents the surveillance ratios for all the states.

A.3 Disease Parameter Space

Among $\mathcal{P}$, p_I, p_E are from literature, I_V is derived from historical data, and we assume I_V follows a discrete uniform distribution. The distributions of τ and N_I are fitted distributions using KS-test on collected samples. The samples used to fit a distribution are collected from historical training seasons. For example, given a state New Jersey, the training data includes 6 seasons from 2010–2011 to 2015–2016, its neighbors are Delaware, New York, and Pennsylvania. Then we can collect $6 * 4 = 24$ samples of ar or N_I for NJ. We calibrate τ using Nelder-Mead [65] algorithm based on each collected pair of (ar, N_I). For each of ar, N_I, τ, we obtained 36 data points for VA and 24 for NJ. At the fitting step, normal and uniform distributions are included. We run KS-test (the null

hypothesis being that the sample is drawn from the reference distribution) to choose a distribution with the highest significance (p-value). The learned parameter space is shown in Table 1. Note that each parameter in $\mathcal{P}$ follows a marginal distribution.

A.4 Baseline Model Settings

In this section, we elaborate the details of model setting of the baselines. Note that, in the experiments, we choose the final model with the best validation accuracy by grid searching. Unless explicitly noted, the hyperparameters are set with default values from python libraries.

- *Single layer LSTM model (LSTM)*: It consists of one LSTM layer and one dense layer. The input is the sequence of state-level ILI incidence and the output is the state-level prediction of the current week. By grid searching, we set the look back window size to 52 and LSTM hidden units to 128. The Adam optimizer is used.
- *AdapLSTM* [82]: This method makes predictions using a simple LSTM model, then adjusts the predictions by applying impacts of weather factors and spatiotemporal factors. The LSTM model has the same setting with single layer LSTM model described above. In Reference [82], the weather features include maximum temperature, minimum temperature, humidity, and precipitation. However, humidity is not used in our experiments, since it is not publicly available in the collected weather dataset. The confidences of symbol pairs (the climatic variable time series and the flu count time series) in our experiment are less than 0.3, which will lead to arbitrary adjustment for predictions. The neighbors of each state used for spatiotemporal adjustment factor are geographical adjacent states that are the same with those used in constructing disease parameter space. For more details please refer to the original paper [82].
- *Simple SARIMA model (SARIMA)*: We use the Seasonal ARIMA model, denoted as $SARIMA(p, d, q)\times(P, D, Q)_m$, where p is the order (number of time lags) of the autoregressive model; d is the degree of differencing (the number of times the data have had past values subtracted); q is the order of the moving-average model; m refers to the number of periods in each season; and the uppercase P, D, Q refer to the autoregressive, differencing, and moving average terms for the seasonal part of the SARIMA model. By grid searching, the selected model is $SARIMA(8, 1, 0) \times (5, 0, 0)_{52}$. No exogenous variables are used in this model.
- *AutoRegression with Google search data (ARGO)* [97]: The method uses an autoregression model utilizing Google search data. We use the publicly available tool from Reference [97]. In our experiment, we set the look back window size to 52 and the training window to 104. In the Google data we collected, all of the top 100 Google correlate terms of VA are flu related, while only one out of the top 100 Google correlated terms of NJ are flu related. This may cause ARGO to perform better on VA than on NJ as discussed in Section 5.7.3.
- *EpiFast* [7]: This method takes the same setting of p_E and p_I as shown in Table 1 and searches for N_I, τ by minimizing the dissimilarity between the predicted and the actual ILI incidence using the Nelder-Mead algorithm [65].

ACKNOWLEDGMENTS

The authors thank members of the Network Systems Science and Advanced Computing (NSSAC) Division for interesting discussion and suggestions related to epidemic science and machine learning.

REFERENCES

[1] ACS. 2009-2013. 2009-2013 5-Year American Community Survey Commuting Flows. Retrieved from https://www.census.gov/data/tables/time-series/demo/commuting/commuting-flows.html.

[2] AHRQ. 2017. Hospital Visits for a Population. Retrieved June 1, 2017 from https://www.ahrq.gov/data/resources/index.html.
[3] Ali Alessa and Miad Faezipour. 2018. A review of influenza detection and prediction through social networking sites. *Theor. Biol. Med. Model.* 15, 1 (2018), 2.
[4] Norman T. J. Bailey et al. 1975. *The Mathematical Theory of Infectious Diseases and Its Applications* (2nd ed.). Charles Griffin & Company Ltd 5a, High Wycombe, Bucks, UK.
[5] Batuhan Bardak and Mehmet Tan. 2015. Prediction of influenza outbreaks by integrating Wikipedia article access logs and Google flu trend data. In *Proceedings of the 2015 IEEE 15th International Conference on Bioinformatics and Bioengineering (BIBE'15)*. IEEE, 1–6.
[6] Christopher L. Barrett, Richard J. Beckman, Maleq Khan, V. S. Anil Kumar, Madhav V. Marathe, Paula E. Stretz, Tridib Dutta, and Bryan Lewis. 2009. Generation and analysis of large synthetic social contact networks. In *Proceedings of the Winter Simulation Conference*. 1003–1014.
[7] Richard Beckman, Keith R. Bisset, Jiangzhuo Chen, Bryan Lewis, Madhav Marathe, and Paula Stretz. 2014. Isis: A networked-epidemiology based pervasive web app for infectious disease pandemic planning and response. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1847–1856.
[8] Richard J. Beckman, Keith A. Baggerly, and Michael D. McKay. 1996. Creating synthetic baseline populations. *Transport. Res. A* 30, 6 (1996), 415–429.
[9] Michael A. Benjamin, Robert A. Rigby, and D. Mikis Stasinopoulos. 2003. Generalized autoregressive moving average models. *J. Am. Stat. Assoc.* 98, 461 (2003), 214–223.
[10] Christoph Bergmeir, Rob J. Hyndman, and José M. Benítez. 2016. Bagging exponential smoothing methods using STL decomposition and Box-Cox transformation. *Int. J. Forecast.* 32, 2 (2016), 303–312.
[11] Matthew Biggerstaff, David Alper, Mark Dredze, Spencer Fox, Isaac Chun-Hai Fung, Kyle S Hickmann, Bryan Lewis, Roni Rosenfeld, Jeffrey Shaman, Ming-Hsiang Tsou, et al. 2016. Results from the centers for disease control and prevention's predict the 2013-2014 Influenza Season Challenge. *BMC Infect. Dis.* 16, 1 (2016), 357.
[12] Matthew Biggerstaff, Michael Johansson, David Alper, Logan C. Brooks, Prithwish Chakraborty, David C. Farrow, Sangwon Hyun, Sasikiran Kandula, Craig McGowan, Naren Ramakrishnan, et al. 2018. Results from the second year of a collaborative effort to forecast influenza seasons in the United States. *Epidemics* 24, 2018 (2018), 26–33.
[13] Keith Bisset and Madhav Marathe. 2009. A cyber-environment to support pandemic planning and response. *DOE SciDAC Mag.* 13 (2009), 36–47.
[14] Keith R. Bisset, Jiangzhuo Chen, Xizhou Feng, V. S. Anil Kumar, and Madhav V. Marathe. 2009. EpiFast: A fast algorithm for large scale realistic epidemic simulations on distributed memory systems. In *Proceedings of the 23rd International Conference on Supercomputing*. ACM, 430–439.
[15] Dirk Brockmann and Dirk Helbing. 2013. The hidden geometry of complex, network-driven contagion phenomena. *Science* 342, 6164 (2013), 1337–1342.
[16] Logan C. Brooks, David C. Farrow, Sangwon Hyun, Ryan J. Tibshirani, and Roni Rosenfeld. 2018. Nonmechanistic forecasts of seasonal influenza with iterative one-week-ahead distributions. *PLoS Comput. Biol.* 14, 6 (2018), e1006134.
[17] Bureau of Labor Statistics. 2017. American Time Use Survey. Retrieved from https://www.bls.gov/tus/.
[18] CDC. 2018. Historical Seasonal Influenza Vaccine Schedule. Retrieved June 01, 2018 from https://www.cdc.gov/flu/professionals/vaccination/vaccinesupply.htm.
[19] CDC. 2019. Disease Burden of Influenza. Retrieved April 01, 2019 from https://www.cdc.gov/flu/about/disease/burden.htm.
[20] CDC. 2019. Fluview Interactive. Retrieved April 20, 2019 from https://www.cdc.gov/flu/weekly/fluviewinteractive.htm.
[21] CDO. 2018. Climate Data Online. Retrived August 28, 2018 from https://www.ncdc.noaa.gov/cdo-web/datasets.
[22] Dennis L. Chao, M. Elizabeth Halloran, Valerie J. Obenchain, and Ira M. Longini Jr. 2010. FluTE, a publicly available stochastic influenza epidemic simulation model. *PLoS Comput. Biol.* 6, 1 (2010), e1000656.
[23] Jean-Paul Chretien, Dylan George, Jeffrey Shaman, Rohit A. Chitale, and F. Ellis McKenzie. 2014. Influenza forecasting in human populations: A scoping review. *PLoS ONE* 9, 4 (2014), e94130.
[24] Zhicheng Cui, Wenlin Chen, and Yixin Chen. 2016. Multi-scale convolutional neural networks for time series classification. *arXiv preprint arXiv:1603.06995* (2016).
[25] Songgaojun Deng, Shusen Wang, Huzefa Rangwala, Lijing Wang, and Yue Ning. 2019. Graph message passing with cross-location attentions for long-term ILI prediction. *arXiv preprint arXiv:1912.10202* (2019).
[26] DOH. 2019. ILI Weekly Reports. Retrieved April 20, 2019 from http://www.nj.gov/health/cd/statistics/flu-stats/.
[27] Colin Doms, Sarah C. Kramer, and Jeffrey Shaman. 2018. Assessing the use of influenza forecasts and epidemiological modeling in public health decision making in the United States. *Sci. Rep.* 8, 1 (2018), 12406.

[28] Andrea Freyer Dugas, Mehdi Jalalpour, Yulia Gel, Scott Levin, Fred Torcaso, Takeru Igusa, and Richard E. Rothman. 2013. Influenza forecasting with Google flu trends. *PLoS ONE* 8, 2 (2013), e56176.
[29] Ceyhun Eksin, Keith Paarporn, and Joshua S. Weitz. 2019. Systematic biases in disease forecasting-the role of behavior change. *Epidemics* 27, (2019), 96–105.
[30] Stephen Eubank, Hasan Guclu, V. S. Anil Kumar, Madhav V. Marathe, Aravind Srinivasan, Zoltan Toroczkai, and Nan Wang. 2004. Modelling disease outbreaks in realistic urban social networks. *Nature* 429, 6988 (2004), 180–184.
[31] James Faghmous, Hung Nguyen, Matthew Le, and Vipin Kumar. 2014. Spatio-temporal consistency as a means to identify unlabeled objects in a continuous data field. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
[32] Christopher C. Fischer, Kevin J. Tibbetts, Dane Morgan, and Gerbrand Ceder. 2006. Predicting crystal structure by merging data mining with quantum mechanics. *Nature Mater.* 5, 8 (Jul. 2006), 641.
[33] Antoine Flahault, Elisabeta Vergu, Laurent Coudeville, and Rebecca F. Grais. 2006. Strategies for containing a global influenza pandemic. *Vaccine* 24, 44 (2006), 6751–6755.
[34] Germain Forestier, François Petitjean, Hoang Anh Dau, Geoffrey I. Webb, and Eamonn Keogh. 2017. Generating synthetic time series to augment sparse datasets. In *Proceedings of the 2017 IEEE International Conference on Data Mining (ICDM'17)*. IEEE, 865–870.
[35] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the International Conference on Machine Learning*. 1050–1059.
[36] GHT. 2018. Google Health Trends. Retrieved August 28, 2018 from https://trends.google.com/trends.
[37] Edward Goldstein, Sarah Cobey, Saki Takahashi, Joel C. Miller, and Marc Lipsitch. 2011. Predicting the epidemic sizes of influenza A/H1N1, A/H3N2, and B: A statistical method. *PLoS Med.* 8, 7 (2011), e1001051.
[38] Google. 2018. Google Correlate Data. Retrieved August 28, 2018 from https://www.google.com/trends/correlate.
[39] Swaminathan Gurumurthy, Ravi Kiran Sarvadevabhatla, and R Venkatesh Babu. 2017. Deligan: Generative adversarial networks for diverse and limited data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 166–174.
[40] Geoffroy Hautier, Christopher C. Fischer, Anubhav Jain, Tim Mueller, and Gerbrand Ceder. 2010. Finding nature's missing ternary oxide compounds using machine learning and density functional theory. *Chem. Mater.* 22, 12 (2010), 3762–3767.
[41] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.* 9, 8 (1997), 1735–1780.
[42] Ting Hua, Chandan K. Reddy, Lei Zhang, Lijing Wang, Liang Zhao, Chang-Tien Lu, and Naren Ramakrishnan. 2018. Social media based simulation models for understanding disease dynamics. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*. International Joint Conferences on Artificial Intelligence Organization, 3797–3804.
[43] IndexMundi. 2010. New Jersey Facts. Retrieved March 1, 2019 from https://www.indexmundi.com/facts/united-states/quick-facts/new-jersey/.
[44] Sasikiran Kandula and Jeffrey Shaman. 2019. Near-term forecasts of influenza-like illness: An evaluation of autoregressive time series approaches. *Epidemics* 27 (2019), 41–51.
[45] Sasikiran Kandula, Teresa Yamana, Sen Pei, Wan Yang, Haruka Morita, and Jeffrey Shaman. 2018. Evaluation of mechanistic and statistical methods in forecasting influenza-like illness. *J. Roy. Soc. Interface* 15, 144 (2018), 20180174.
[46] Anuj Karpatne, Gowtham Atluri, James H. Faghmous, Michael Steinbach, Arindam Banerjee, Auroop Ganguly, Shashi Shekhar, Nagiza Samatova, and Vipin Kumar. 2017. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Trans. Knowl. Data Eng.* 29, 10 (2017), 2318–2331.
[47] Jaya Kawale, Stefan Liess, Arjun Kumar, Michael Steinbach, Peter Snyder, Vipin Kumar, Auroop R. Ganguly, Nagiza F. Samatova, and Fredrick Semazzi. 2013. A graph-based approach to find teleconnections in climate data. *Stat. Anal. Data Min.* 6, 3 (2013), 158–179.
[48] Ankush Khandelwal, Anuj Karpatne, Miriam E. Marlier, Jongyoun Kim, Dennis P. Lettenmaier, and Vipin Kumar. 2017. An approach for global monitoring of surface water extent variations in reservoirs using MODIS data. *Remote Sens. Environ.* 202 (2017), 113–128.
[49] Ankush Khandelwal, Varun Mithal, and Vipin Kumar. 2015. Post classification label refinement using implicit ordering constraint among data instances. In *Proceedings of the 2015 IEEE International Conference on Data Mining*. IEEE, 799–804.
[50] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[51] Mario Michael Krell, Anett Seeland, and Su Kyoung Kim. 2018. Data augmentation for brain-computer interfaces: Analysis on event-related potentials data. *arXiv preprint arXiv:1801.02730* (2018).
[52] Yu A. Kuznetsov and Carlo Piccardi. 1994. Bifurcation analysis of periodic SEIR and SIR epidemic models. *J. Math. Biol.* 32, 2 (1994), 109–121.

[53] Håvard Kvamme, Nikolai Sellereite, Kjersti Aas, and Steffen Sjursen. 2018. Predicting mortgage default using convolutional neural networks. *Expert Syst. Appl.* 102 (2018), 207–217.
[54] Arthur Le Guennec, Simon Malinowski, and Romain Tavenard. 2016. Data augmentation for time series classification using convolutional neural networks. In *ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data, Sep 2016, Riva Del Garda, Italy.*
[55] Jung Min Lee, Donghoon Choi, Giphil Cho, and Yongkuk Kim. 2012. The effect of public health interventions on the spread of influenza among cities. *J. Theor. Biol.* 293 (2012), 131–142.
[56] Ira M. Longini Jr., Paul E. M. Fine, and Stephen B. Thacker. 1986. Predicting the global spread of new infectious agents. *Am. J. Epidemiol.* 123, 3 (1986), 383–391.
[57] Markku Löytönen and Sonia I. Arbona. 1996. Forecasting the AIDS epidemic in Puerto Rico. *Soc. Sci. Med.* 42, 7 (1996), 997–1010.
[58] Antonella Lunelli, Andrea Pugliese, and Caterina Rizzo. 2009. Epidemic patch models applied to pandemic influenza: Contact matrix, stochasticity, robustness of predictions. *Math. Biosci.* 220, 1 (2009), 24–33.
[59] Achla Marathe, Bryan Lewis, Jiangzhuo Chen, and Stephen Eubank. 2011. Sensitivity of household transmission to household contact structure and size. *PLoS ONE* 6, 8 (Aug. 2011).
[60] Marco Marchesi. 2017. Megapixel size image creation using generative adversarial networks. *arXiv preprint arXiv:1706.00082* (2017).
[61] Craig J. McGowan, Matthew Biggerstaff, Michael Johansson, Karyn M. Apfeldorf, Michal Ben-Nun, Logan Brooks, Matteo Convertino, Madhav Erraguntla, David C. Farrow, John Freeze, et al. 2019. Collaborative efforts to forecast seasonal influenza in the United States, 2015–2016. *Sci. Rep.* 9, 1 (2019), 683.
[62] Noelle-Angelique M. Molinari, Ismael R. Ortega-Sanchez, Mark L. Messonnier, William W. Thompson, Pascale M. Wortley, Eric Weintraub, and Carolyn B. Bridges. 2007. The annual impact of seasonal influenza in the US: Measuring disease burden and costs. *Vaccine* 25, 27 (2007), 5086–5096.
[63] Haruka Morita, Sarah Kramer, Alexandra Heaney, Harold Gil, and Jeffrey Shaman. 2018. Influenza forecast optimization when using different surveillance data types and geographic scale. *Influenza Other Respir. Virus.* 12, 6 (2018), 755–764.
[64] NDSSL. 2014. Synthetic Data of Montgomery County, Virginia. Retrieved from http://ndssl.vbi.vt.edu/synthetic-data/.
[65] John A. Nelder and Roger Mead. 1965. A simplex method for function minimization. *Comput. J.* 7, 4 (1965), 308–313.
[66] Elaine Nsoesie, Madhav Mararthe, and John Brownstein. 2013. Forecasting peaks of seasonal influenza epidemics. *PLoS Curr.* 5 (2013).
[67] Elaine O. Nsoesie, Richard J. Beckman, Sara Shashaani, Kalyani S. Nagaraj, and Madhav V. Marathe. 2013. A simulation optimization approach to epidemic forecasting. *PLoS ONE* 8, 6 (2013), e67164.
[68] Elaine O. Nsoesie, John S. Brownstein, Naren Ramakrishnan, and Madhav V. Marathe. 2014. A systematic review of studies on forecasting the dynamics of influenza outbreaks. *Influenza Other Respir. Virus.* 8, 3 (2014), 309–316.
[69] Dave Osthus, James Gattiker, Reid Priedhorsky, Sara Y. Del Valle, et al. 2019. Dynamic Bayesian influenza forecasting in the United States with hierarchical discrepancy (with discussion). *Bayes. Anal.* 14, 1 (2019), 261–312.
[70] Jon Parker and Joshua M. Epstein. 2011. A distributed platform for global-scale agent-based models of disease transmission. *ACM Trans. Model. Comput. Simul.* 22, 1, Article 2 (Dec. 2011), 25 pages.
[71] Sen Pei, Sasikiran Kandula, Wan Yang, and Jeffrey Shaman. 2018. Forecasting the spatial transmission of influenza in the United States. *Proc. Natl. Acad. Sci. U.S.A.* 115, 11 (2018), 2752–2757.
[72] Luis Perez and Jason Wang. 2017. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621* (2017).
[73] Nicholas G. Reich, Logan C. Brooks, Spencer J. Fox, Sasikiran Kandula, Craig J. McGowan, Evan Moore, Dave Osthus, Evan L. Ray, Abhinav Tushar, Teresa K. Yamana, Matthew Biggerstaff, Michael A. Johansson, Roni Rosenfeld, and Jeffrey Shaman. 2019. A collaborative multiyear, multimodel assessment of seasonal influenza forecasting in the United States. *Proc. Natl. Acad. Sci. U.S.A.* 116, 8 (2019), 3146–3154. DOI: https://doi.org/10.1073/pnas.1812594116 arXiv:https://www.pnas.org/content/116/8/3146.full.pdf
[74] Hamada Rizk, Ahmed Shokry, and Moustafa Youssef. 2019. Effectiveness of data augmentation in cellular-based localization using deep learning. *arXiv preprint arXiv:1906.08171* (2019).
[75] Jan Schlüter and Thomas Grill. 2015. Exploring data augmentation for improved singing voice detection with neural networks. In *Proceedings of the Annual Conference of the International Society for Music Information Retrieval (ISMIR'15).* 121–126.
[76] Jeffrey Shaman and Alicia Karspeck. 2012. Forecasting seasonal outbreaks of influenza. *Proceedings of the National Academy of Sciences* 109, 50 (2012), 20425–20430.
[77] Jeffrey Shaman, Alicia Karspeck, Wan Yang, James Tamerius, and Marc Lipsitch. 2013. Real-time influenza forecasts during the 2012-2013 season. *Nature Commun.* 4, 2837 (2013).

[78] Thinglink. 2019. New Jersey Regions Map. Retrieved from https://www.thinglink.com/scene/788830737167024130.
[79] Ashleigh R. Tuite, Amy L. Greer, Michael Whelan, Anne-Luise Winter, Brenda Lee, Ping Yan, Jianhong Wu, Seyed Moghadas, David Buckeridge, Babak Pourbohloul, et al. 2010. Estimated epidemiologic parameters and morbidity associated with pandemic H1N1 influenza. *Can. Med. Assoc. J.* 182, 2 (2010), 131–136.
[80] Terry Taewoong Um, Franz Michael Josef Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana Kulić. 2017. Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. *arXiv preprint arXiv:1706.00527* (2017).
[81] Cristina Nader Vasconcelos and Bárbara Nader Vasconcelos. 2017. Increasing deep learning melanoma classification by classical and expert knowledge based image transforms. *CoRR, abs/1702.07025* 1 (2017).
[82] Siva R. Venna, Amirhossein Tavanaei, Raju N. Gottumukkala, Vijay V. Raghavan, Anthony S. Maida, and Stephen Nichols. 2019. A novel data-driven model for real-time influenza forecasting. *IEEE Access* 7 (2019), 7691–7701.
[83] Cécile Viboud, Pierre-Yves Boëlle, Fabrice Carrat, Alain-Jacques Valleron, and Antoine Flahault. 2003. Prediction of the spread of influenza epidemics by the method of analogues. *Am. J. Epidemiol.* 158, 10 (2003), 996–1006.
[84] Cécile Viboud, Kaiyuan Sun, Robert Gaffey, Marco Ajelli, Laura Fumanelli, Stefano Merler, Qian Zhang, Gerardo Chowell, Lone Simonsen, Alessandro Vespignani, et al. 2018. The RAPIDD ebola forecasting challenge: Synthesis and lessons learnt. *Epidemics* 22 (2018), 13–21.
[85] Svitlana Volkova, Ellyn Ayton, Katherine Porterfield, and Courtney D. Corley. 2017. Forecasting influenza-like illness dynamics for military populations using neural networks and social media. *PLoS ONE* 12, 12 (2017), e0188941.
[86] Lijing Wang, Jiangzhuo Chen, and Achla Marathe. 2018. A framework for discovering health disparities among cohorts in an influenza epidemic. *World Wide Web* 22, 6 (2018), 2997–3020.
[87] Lijing Wang, Jiangzhuo Chen, and Madhav Marathe. 2019. DEFSI: Deep learning based epidemic forecasting with synthetic information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 9607–9612.
[88] Lijing Wang, Jiangzhuo Chen, and Madhav Marathe. 2019. TDEFSI: Theory Guided Deep Learning Based Epidemic Forecasting with Synthetic Information (Supplement). Retrieved from https://github.com/christa60/defsi/blob/master/animation.gif.
[89] Zheng Wang, Prithwish Chakraborty, Sumiko R. Mekaru, John S. Brownstein, Jieping Ye, and Naren Ramakrishnan. 2015. Dynamic poisson autoregression for influenza-like-illness case count prediction. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1285–1294.
[90] WHO. 2019. Seasonal Influenza. Retrieved April 1, 2019 from http://www.who.int/news-room/fact-sheets/detail/influenza-(seasonal).
[91] Ken C. L. Wong, Linwei Wang, and Pengcheng Shi. 2009. Active model with orthotropic hyperelastic material for cardiac image analysis. In *Proceedings of the International Conference on Functional Imaging and Modeling of the Heart*. Springer, 229–238.
[92] Sebastien C. Wong, Adam Gatt, Victor Stamatescu, and Mark D. McDonnell. 2016. Understanding data augmentation for classification: When to warp? In *Proceedings of the 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA'16)*. IEEE, 1–6.
[93] Yuexin Wu, Yiming Yang, Hiroshi Nishiura, and Masaya Saitoh. 2018. Deep learning for epidemiological predictions. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 1085–1088.
[94] Jingjia Xu, John L. Sapp, Azar Rahimi Dehaghani, Fei Gao, Milan Horacek, and Linwei Wang. 2015. Robust transmural electrophysiological imaging: Integrating sparse and dynamic physiological models into ECG-based inference. In *Proceedings of the Medical Image Computing and Computer-Assisted Intervention (MICCAI'15)*, Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro Frangi (Eds.). Springer International Publishing, Cham, 519–527.
[95] Qinneng Xu, Yulia R. Gel, L. Leticia Ramirez Ramirez, Kusha Nezafati, Qingpeng Zhang, and Kwok-Leung Tsui. 2017. Forecasting influenza in Hong Kong with Google search queries and statistical model fusion. *PloS ONE* 12, 5 (2017), e0176690.
[96] Shihao Yang, Mauricio Santillana, John S. Brownstein, Josh Gray, Stewart Richardson, and S. C. Kou. 2017. Using electronic health records and Internet search information for accurate influenza forecasting. *BMC Infect. Dis.* 17, 1 (2017), 332.
[97] Shihao Yang, Mauricio Santillana, and Samuel C. Kou. 2015. Accurate estimation of influenza epidemics using Google search data via ARGO. *Proc. Natl. Acad. Sci. U.S.A.* 112, 47 (2015), 14473–14478.
[98] Wan Yang, Alicia Karspeck, and Jeffrey Shaman. 2014. Comparison of filtering methods for the modeling and retrospective forecasting of influenza epidemics. *PLoS Comput. Biol.* 10, 4 (2014), e1003583.
[99] Wan Yang, Marc Lipsitch, and Jeffrey Shaman. 2015. Inference of seasonal and pandemic influenza transmission dynamics. *Proc. Natl. Acad. Sci. U.S.A.* 112, 9 (2015), 2723–2728.
[100] Wan Yang, Donald R. Olson, and Jeffrey Shaman. 2016. Forecasting influenza outbreaks in boroughs and neighborhoods of New York City. *PLoS Comput. Biol.* 12, 11 (2016), e1005201.

[101] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2016. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530* (2016).

[102] Liang Zhao, Jiangzhuo Chen, Feng Chen, Wei Wang, Chang-Tien Lu, and Naren Ramakrishnan. 2015. Simnest: Social media nested epidemic simulation via online semi-supervised deep learning. In *Proceedings of the 2015 IEEE International Conference on Data Mining*. IEEE, 639–648.

[103] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 2223–2232.

Received May 2019; revised September 2019; accepted January 2020