EpiDeep: Exploiting Embeddings for Epidemic Forecasting

Bijaya Adhikari*, Xinfeng Xu†, Naren Ramakrishnan* and B. Aditya Prakash*
*Department of Computer Science, Virginia Tech.
†Department of Physics, Virginia Tech.
Email: *[bijaya, naren, badityap]@cs.vt.edu, †xinfeng@vt.edu

ABSTRACT

Influenza leads to regular losses of lives annually and requires careful monitoring and control by health organizations. Annual influenza forecasts help policymakers implement effective counter-measures to control both seasonal and pandemic outbreaks. Existing forecasting techniques suffer from problems such as poor forecasting performance, lack of modeling flexibility, data sparsity, and/or lack of intepretability. We propose EpiDeep, a novel deep neural network approach for epidemic forecasting which tackles all of these issues by learning meaningful representations of incidence curves in a continuous feature space and accurately predicting future incidences, peak intensity, peak time, and onset of the upcoming season. We present extensive experiments on forecasting ILI (influenza-like illnesses) in the United States, leveraging multiple metrics to quantify success. Our results demonstrate that EpiDeep is successful at learning meaningful embeddings and, more importantly, that these embeddings evolve as the season progresses. Furthermore, our approach outperforms non-trivial baselines by up to 40%.

CCS CONCEPTS

• **Information systems** → **Data stream mining**; • **Computing methodologies** → **Neural networks**; • **Applied computing** → **Health informatics**.

ACM Reference Format:
Bijaya Adhikari, Xinfeng Xu, Naren Ramakrishnan and B. Aditya Prakash. 2019. EpiDeep: Exploiting Embeddings for Epidemic Forecasting. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19), August 4–8, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3292500.3330917

1 INTRODUCTION

Seasonal influenza is a major health issue that affects many people across the world. The US national Centers for Disease Control and Prevention (CDC) reports that there were 30,453 laboratory-confirmed influenza related hospitalizations in the 2017/18 influenza season in the United States alone. According to the same estimate, the 2017-18 season saw a larger number of deaths due to influenza than in any of the past five seasons. These statistics reveal that despite years of efforts, accurately predicting key indicators of the flu season and employing counter-measures remain major challenges.

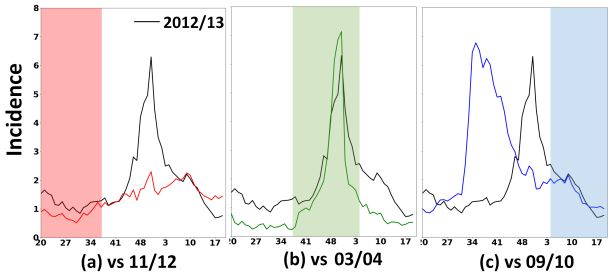


Figure 1: Changing similarities: wILI incidence curve for 2012/13 season in HHS Region 4 (black curve in all three figures) is most similar to that of 2011/12 season in the beginning of the season (red band), to 2003/04 in the middle of the season (green band), and finally to 2009/10 at the end of the season (blue band). Note that EpiDeep *automatically* learns to infer these closest seasons at various stages.

FluSight task. To encourage research into making more accurate forecasts, the CDC has been hosting the 'FluSight' challenge for seasonal influenza forecasting at the national and regional levels [1, 2]. It involves predicting on a weekly basis, multiple aspects of the current influenza season, which is represented as a time series of the weighted Influenza-like Illness (wILI) data. The wILI data released by the CDC is collected by the Outpatient Influenza-like Illness Surveillance Network (ILINet) which consists of more than 3,500 outpatient healthcare providers all over the United States. Each week the healthcare providers voluntarily report the percentage of patients visiting for Influenza-like Illness (ILI). ILI is defined as "fever (temperature of 100°F [37.8°C] or greater) and a cough and/or a sore throat without a known cause other than influenza"[1]. The CDC compiles the ILI reports, weights the total percentage visits by the state population and computes the resulting wILI values for the national and local regions. It then releases this data typically with a delay of two weeks (weekly wILI incidence curves for each season since 1997/98 are publicly available[2]).

The FluSight challenge typically starts on week 40 of the calendar year and lasts till week 20 of the following year when the influenza activity is high [1, 2]. Given the wILI data, the forecasting tasks include predicting the Future Incidences, Seasonal Peak Intensity, Seasonal Peak Time and Onset Week (we will discuss each forecasting target in more detail in the next section).

[1]https://www.cdc.gov/flu/weekly/overview.htm
[2]https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html

KDD '19, August 4–8, 2019, Anchorage, AK, USA

ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330917

Prior work. There has been a surge of research interest in developing methods for the flu forecasting tasks in recent times. Statistical methods for flu forecasting such as [4, 7] fit a predefined statistical model on historical data and use it for forecasting. These methods typically are either too simple, or fail to incorporate domain knowledge like the epidemiological dynamics, or require laborious feature engineering or are not interpretable. They often act as a "black-box" and fail in providing any interpretation or explanation of the forecasts. Note that interpretability is very important, as it provides an insight on how to fine tune the model for better predictions and also as it guides decision makers. For example, knowing that this season is similar to another one helps the policy experts to focus efforts [21, 22]: these similarities may be due to similarities in the environmental, geographical or biological (like similar strains of the virus) or other factors.

On the other hand, mechanistic models [21, 26, 35] are motivated by domain knowledge; they typically include various factors such as the epidemiological and associated human mobility models and make forecasts based on simulation and/or some simple aggregation. While insightful and usually interpretable, they require a lot of calibration, and are also usually too rigid to generalize well and accurately fit the data [22]. We can see our work in context of forecasting general time-series with sparse data as well. Prior works typically rely on leveraging correlated time-series instances to overcome sparsity [15, 23]. While useful in domains like e-commerce and real estate, where there are a large number of sparse instances (like sales for different items, housing price for each zip-code e.t.c), they do not directly apply to influenza forecasting as there is sparsity in the number of instances as well. Hence, in this paper, inspired by the above significant gaps in literature, we develop an end-to-end neural learning model `EpiDeep`, which is flexible, does not require any feature engineering, and aids in interpretation while also maintaining an excellent forecasting performance.

Challenges. We focus on interpretability through finding similarities between seasons (as discussed before, knowing similar seasons is especially helpful for decision makers). This is challenging to capture, as influenza seasons are highly dynamic [21], which can be due to weather patterns, dominant virus types etc. For example, consider the wILI incidence curve for season 2012/13 for HHS Region 4 in Figure 1—the curve is similar to that of 2011/12 in the beginning of the season. However as the season progresses, the two curves start differing from each other. In the middle of the flu season, we find that the 2012/13 wILI curve matches closely with the 2003/04 season instead, and is closest to the much different 2009/10 season in the end. Hence traditional time-series forecasting methods like `ARIMA` are unlikely to perform well, as they do not have the flexibility to capture this dynamic nature without significant modelling input [22]. Neural networks, on the other hand, can overcome this issue as they can infer meaningful representations from the data (historical seasons) itself. However, data sparsity is also a major issue here (e.g. wILI data surveillance began only in the late 1990s). Hence sequence prediction neural models such as Long Short Term Memory (LSTM) [14], which typically require a large amount of data, are not an straightforward fit here. Therefore, we need an approach which can explicitly leverage the evolving similarities between historical seasons and yet be able to make accurate forecasts with the sparse data.

Approach and Contributions. Our main idea is that learning season similarities should help in both forecasting and interpretation. To this end, we design `EpiDeep` (Figure 2 summarizes the overall architecture). We feed the historical wILI incidence data to our model. It then learns to *embed* the historical seasons in a feature space and leverages the embeddings together with the current season's wILI incidence curve to forecast all the CDC targets. Given the embeddings of the seasons, we can easily answer questions like which season was the closest to the current season at the time of prediction, how does the similarity of the current season with others evolve over time, and so on. These patterns are non-trivial to find directly from the incidence curves as we may have to check all possible snippets and combinations of the historical data. Note in Figure 1, `EpiDeep` can automatically infer these closest seasons at various stages.

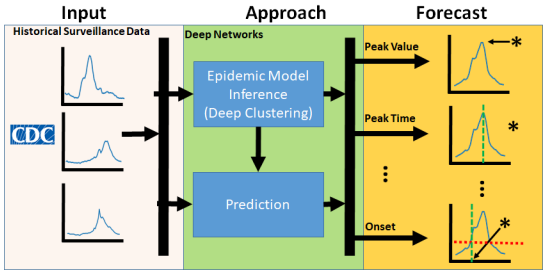


Figure 2: A visual representation of our approach. We leverage historical surveillance data to learn meaningful representations via evolving clustering and forecast multiple metrics of interest.

Our main contributions are as follows:

- **A Novel Deep Learning Approach:** We leverage recent development in deep clustering to design an end-to-end time-series embedding, clustering and forecasting approach. To the best of our knowledge, we are the first to do so.
- **Interpretability with Performance:** Our learnt embeddings help in determining the closest historical seasons to the current season at the time of prediction, helping in both interpretability and performance. To the best of our knowledge, we are the first to propose a non-trivial deep learning model for epidemic forecasting which ensures both interpretability and excellent forecasting performance.
- **Rigorous Empirical studies:** We conduct extensive experiments in multiple settings using real ILI data. Our results show that `EpiDeep` outperforms non-trivial baselines by up to 40%, consistently across multiple metrics, and is also interpretable.

Next, we discuss the CDC challenge in detail and formulate the problems. We then describe `EpiDeep` in Section 3, discuss empirical studies in Section 4, related literature in Section 5 and finally conclude in Section 6. We defer some additional experiments and details to the supplementary for sake of space and readability.

2 PROBLEM STATEMENT

The CDC FluSight challenge provides a uniform prediction goal for influenza forecasting and hence we model our problem based on that task. As mentioned earlier, the historical wILI incidence reports

are available till week $t-1$ while making predictions for week t. There are four forecasting targets: Future Incidence, Seasonal Peak Time, Seasonal Peak Intensity, and Onset Week. All the target descriptions are based on the CDC FluSight 2018/2019 challenge[3].

Future Incidence: This refers to short term future incidence predictions. This includes the prediction of wILI data one to four weeks ahead of the latest wILI data release. Since the ILINet data is delayed by two weeks, at week t in the season, the short term forecasts correspond to predicting wILI values for week $t-1$, t, $t+1$, and $t+2$, given the data till week $t-2$.

Seasonal Peak Intensity: The peak intensity measures the maximum intensity of the influenza in the given season (i.e. the highest numerical value of wILI in the given season). Since the amount of resources needed to influenza prevention is directly affected by the peak intensity, it is an important task.

Seasonal Peak Week: The peak week is the time when the peak intensity is observed. The CDC defines the seasonal peak week as the surveillance week when the wILI value rounded to one decimal point is the highest. The peak week prediction is an important problem as it allows for planning ahead and strategic resource allocation.

Onset Week: The onset week represents the week when the flu season 'takes-off'. The CDC defines it as the surveillance week when percentage of visits for influenza-like illness (ILI) reaches or exceeds a pre-defined baseline value for three consecutive weeks. The onset week is the first week of such three weeks. The baseline value may vary from year to year and from region to region. The onset week prediction is important as the start of the flu season determines when many precautionary and preventive measures are deployed. It also gives healthcare providers an early notification that a rise in ILI cases is impending.

At week $t+2$, we are given the time-series $\mathcal{Y}_c = \{y_c^1, y_c^2, \ldots, y_c^t\}$ representing initial stage of the current season c till week t. The values y_c^i represent the wILI values for the week i. Our goal is, given $\mathcal{Y}_c$, predict all four targets for the season c: short term forecasts, peak intensity, peak week, and onset. Formally:

Problem 1. *EpidemicPrediction*

***Given**: a time-series $\mathcal{Y}_c = \{y_c^1, y_c^2, \ldots, y_c^t\}$ representing the current season c till week t and a CDC baseline b_c*

Predict:

- ***Task 1:** Future Incidence Prediction: $\forall_{i=t+1}^{t+4} y_c^i$*
- ***Task 2:** Peak Intensity Prediction: $\max y_c^i \forall_{i=1}^{T}$, where T is the last week of the season.*
- ***Task 3:** Peak Time Prediction: $\arg\max_i y_c^i \forall_{i=1}^{T}$, where T is the last week of the season.*
- ***Task 4:** Onset Prediction: Week j such that $\forall_{i=j}^{j+3}, y_c^i \geq b_c$*

3 OUR APPROACH

Overview. We first give an overview of EpiDeep. Here, we are given historical wILI time-series (the 'training set'), using which we propose a learning based approach to solve Problem 1. Specifically, we design a deep neural network to encode various aspects of our data. At a high level, our model tries to leverage similarities between the observed stage of the current season with the past seasons to make accurate future predictions (using the future observed trends of the past seasons). We train the deep model by leveraging a set of historical wILI incidence $Y = \{\mathcal{Y}_1, \mathcal{Y}_2, \mathcal{Y}_3, \ldots, \mathcal{Y}_{c-1}\}$. Once the model is trained, we use it to predict various metrics of interest for the current season $\mathcal{Y}_c$. Following literature [4], we define each season $\mathcal{Y}_i$ to begin at week 20 of the calendar year.

The main challenge in implementing our idea of 'evolving similarities for prediction' is that the current season $\mathcal{Y}_c$ is observed only till week t, whereas the historical seasons in Y are fully observed. Hence, naïve ideas like computing distance between the curves are not suitable here. We resolve this issue in two steps. First, we learn the embeddings to capture similarities between the current season and historical seasons restricted till time t. However, this is not enough as the entire season-length historical data will have useful information to aid in forecasting of the current season. Hence after that, we also find embeddings to capture similarities between the full length historical seasons *without* the current season. We finally learn a mapping function to map between these representations and then further leverage the embeddings for the forecasting tasks. Figure 3 (a) shows the overall architecture of EpiDeep. Next, we discuss these steps in detail.

Our first step is accomplished by the 'Query Length Data Clustering' module, which leverages deep clustering techniques to learn feature representations. Here, we treat the observed part of the current season $\mathcal{Y}_c$ (for which we need to forecast) as a 'query' to the historical fully observed season-length incidence curves. We refer to the snippets of the historical season till the observed week t as query length historical wILI data. This module learns the feature representation of query length historical wILI data with the current season $\mathcal{Y}_c$. To capture the similarity between the current season and historical seasons, this module learns embeddings for each season in Y in conjunction with $\mathcal{Y}_c$ and clusters them by leveraging a deep clustering method.

Similarly for the second step, the 'Full length Data clustering' component embeds the full length historical data in a continuous space, such that the clustering of the embeddings are meaningful. Since, we do not have access to the complete incidence curve for the current season $\mathcal{Y}_c$, we learn a mapping function to convert the embeddings from the query length space to the space representing complete seasons. This allows us to learn embeddings of the current season in the space of fully observed historical wILI data.

We also have the 'Input Encoder' module, which provides a succinct representation of the current season $\mathcal{Y}_c$. The input encoder is designed in a way such that it can extract important information from any snippets in $\mathcal{Y}_c$. Finally, the 'Decoder' module combines outputs from the clustering layers and the encoder to make prediction for the final forecasting targets. In the following, we describe the various components of our model.

3.1 Encoding the Input

Here we adopt a recurrent neural network (RNN) to capture temporal dynamics of of the time-series $\mathcal{Y}_c = \{y_c^1, y_c^2, \ldots, y_c^t\}$. RNN processes variable length input sequence by maintaining a hidden state $h_j \in \mathrm{R}^K$ and continuously updating it as $h_j = f(y_c^j, h_{j-1})$, where f is a non-linear activation function. Here, we leverage Long short-term memory networks (LSTM network) [11], a special type of RNN designed to work better with long sequences of

[3] https://predict.cdc.gov/post/5ba1504e5619f003acb7e18f

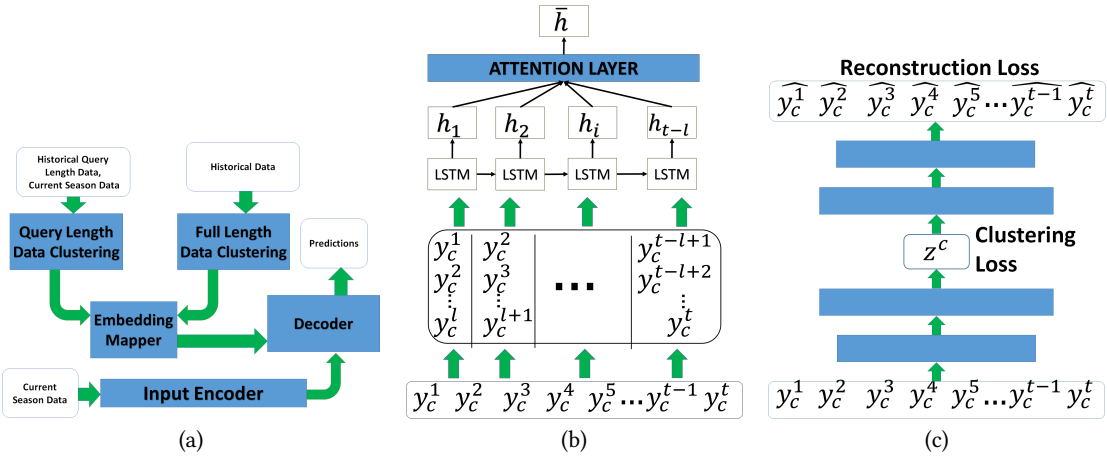


Figure 3: (a) The overall architecture of EpiDeep. It consists of clustering/embedding, encoder, and decoder modules. (b) The architecture of the encoder module. (c) The architecture of the deep clustering module.

data. Given a sequence, $\mathcal{Y}_c = \{y_c^1, y_c^2, \ldots, y_c^t\}$, we first convert it to a matrix $\mathbf{Y} \in \mathrm{R}^{l\times(t-l+1)}$, such that j^{th} column of Y consists of $[y_c^j, y_c^{j+1}, \ldots, y_c^{j+l-1}]$. Now, the j^{th} input to our LSTM network is the j^{th} column of the matrix $\mathbf{Y}$, i.e. $\mathbf{Y}[:j]$. Now, the LSTM equations for j^{th} input are as follows:

$$i_j = \sigma(\mathbf{W}_i\mathbf{Y}[:j] + \mathbf{U}_i\mathbf{h}_{i-1} + \mathbf{b}i) \quad (1)$$

$$f_j = \sigma(\mathbf{W}_f\mathbf{Y}[:j] + \mathbf{U}_f\mathbf{h}_{i-1} + \mathbf{b}f) \quad (2)$$

$$C_j = i_j \odot \tanh(\mathbf{W}_c\mathbf{Y}[:j] + \mathbf{U}_c\mathbf{h}_{i-1} + \mathbf{b}c) + f_j \odot C_{j-1} \quad (3)$$

$$\mathbf{o}_j = \sigma(\mathbf{W}_o\mathbf{Y}[:j] + \mathbf{U}_o\mathbf{h}_{i-1} + \mathbf{b}o) \quad (4)$$

$$\mathbf{h}_j = o_j \odot \tanh(C_j) \quad (5)$$

Here, the matrices $\mathbf{W} \in \mathrm{R}^{\mathrm{h}\times\mathrm{l}}$ and $\mathbf{U} \in \mathrm{R}^{\mathrm{h}\times\mathrm{h}}$ are the weight matrices and h represents the size of hidden units.

Typically, only the last output of the LSTM is leveraged for the prediction. For our query series $\mathcal{Y}_c = \{y_c^1, y_c^2, \ldots, y_c^t\}$, this would mean only $\mathbf{o}_t$ would be leveraged for prediction. However, such an approach has a clear disadvantage. It is known that the official estimates for ILI surveillance data are often delayed and revised multiple times before they stabilize [7]. In such scenario, the data pertaining to the final time-stamp y_c^t is most vulnerable to revision. Therefore, over reliance on the last time-stamp could harm the predictive power of the model. To overcome this issue, we require a mechanism to assign varying degree of importance to the earlier states of the model.

A mechanism typically used in NLP to give partial importance to each state of the RNN is known as the *attention mechanism* [19]. The main idea here is to produce the output of the RNN as a weighted sum of all previous hidden states. The context vector $\bar{\mathbf{h}}_j$, for j^{th} input then is computed as $\bar{\mathbf{h}}_j = \sum_z \alpha_{jz}^a \mathbf{h}_{jz}$, where

$$\alpha_{js}^a = \frac{\exp(u_{jx}^T u_\alpha)}{\sum_z \exp(u_{zx}^T u_a)} \quad (6)$$

$$u_{js} = \tanh(\mathbf{W}_a^T \mathbf{h}_{js} + \mathbf{b}_a) \quad (7)$$

Since, the context vector $\bar{\mathbf{h}}_j$ is the weighted sum of previous hidden states, the model can infer variable weights for each hidden state and has the flexibility to put more/less attention to each input. The context vector $\bar{\mathbf{h}}_j$ can now be fed into the decoder network to make predictions.

The complete encoder module is shown in Figure 3 (b). As mentioned earlier, the input time series $\mathcal{Y}_c$ is converted to the matrix Y. Each column of the matrix is fed into the LSTM network. The output of the LSTM network is combined using the attention mechanism to learn a single context vector $\bar{\mathbf{h}}_j$.

3.2 Closest Season based on Deep Clustering

A simple way to train our model is to feed the context vector $\bar{\mathbf{h}}_j$ from the attention model to the decoder. The assumption here is that the deep model would be able to infer relevant information from the historical incidence time-series Y and leverage it for correct prediction. However, due to the sparsity of the data available, a more explicit model to extract similarities between season is warranted. Our main idea is to learn the embedding $\mathbf{z}_c$ of the partially observed current season $\mathcal{Y}_c$ in the latent space such that the distance between z_c and other 'similar' historical season is minimized. To this end, we develop an embedding layer with multiple deep components.

Clustering Query Length Data: Here we are concerned with embedding the similar seasons such that seasons which are closer to each other are embedded together. Recall that in the current season $\mathcal{Y}_c = \{y_c^1, y_c^2, \ldots, y_c^t\}$, we have only observed incidence till time t. Hence, we leverage deep clustering [33] of the set $\mathrm{Y}_t = \{\mathcal{Y}_i[0:t] | \forall_{i=1}^{c-1}\} \cup \{\mathcal{Y}_c\}$ to learn meaningful embedding of $\mathcal{Y}_c$. Here we adopt the Improved Deep Embedded clustering (IDEC) [13] method to cluster Y_t and to learn embeddings. IDEC clusters given input by augmenting clustering loss to the reconstruction loss.

Let, the vector $\mathbf{z}_i^t$ be the encoding of the the season $\mathcal{Y}_i$ given by the encoder. Let, μ_j be the cluster center for cluster j. Now, our clustering objective is $L_c^t = KL(P||Q) = \sum_i \sum_j p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right)$, where q_{ij} is the similarity between the embedding $\mathbf{z}_i$ and cluster center μ_j as given the the Student's t-distribution, i.e.

$$q_{ij} = \frac{(1 + ||\mathbf{z}_i^t - \mu_j||^2)^{-1}}{\sum_j (1 + ||\mathbf{z}_i^t - \mu_j||^2)^{-1}} \quad (8)$$

and p_{ij} is the target distribution given by

$$p_{ij} = \frac{q_{ij}^2 / \sum_i q_{ij}}{\sum_j (q_{ij}^2 / \sum_i q_{ij})} \tag{9}$$

In addition to the clustering loss L_c^t parameterized by the target distribution, we also minimizes the reconstruction loss L_r^t, as a regular auto-encoder, to preserve the local structure of the data.

The embedding z_c^t generated by the deep clustering network only tries to capture the similarities between the historical season Y and the current season $\mathcal{Y}_c$ only till time t, as the current season is observed only till time t.

Clustering Full Length Data: Once trained, the vectors $\mathbf{z}^t$ give us meaningful embedding of the set Y_t. However, our main goal is to infer how similar the seasons would look like at the end of the season rather than at the prediction time t. In other words, given the current season $\mathcal{Y}_c$ at time t, can we learn embedding of the $\mathcal{Y}_c$ in the space of Y rather than Y_t. Since we have not observed $\mathcal{Y}_C$ entirely, we are unable to embed it together with Y.

To avert this issue, our idea is to cluster and embed the historical seasons in Y which are all observed entirely till time T using the same architecture as earlier. To this end, we optimize for the clustering loss L_c^T and the reconstruction loss L_r^T for the historical seasons Y (in the same manner as above). Hence, for each season $y_i \in$ Y, we obtain a full-length embedding $\mathbf{z}_i^T$.

Mapping the Embeddings: Now, our problem reduces to translating the embedding learned from query length data to the space of full length data, i.e, mapping $\mathbf{z}_c^t$ to $\mathbf{z}_c^T$. To this end, we learn the mapping function f_{emd} to map z_i^t to z_i^T. Our idea is to leverage $\mathbf{z}_i^t$ and $\mathbf{z}_i^T$ for each historical season in Y to learn the function f_{emd}. To this end, we optimize the objective $L_{emd} = \sum_i ||z_i^T - f_{emd}(z_i^t)||_2^2$

Here, $f_{emd}(z_i^t)$ is the translation of z_i^t to the space of z_i^T. We represent the function f_{emd} as a feed-forward neural network. Once the complete network is trained, we obtain $z_c^T = f_{emd}(z_c^t)$ as our embedding for the current season $\mathcal{Y}_c$.

3.3 Prediction

The next component of our deep model EpiDeep, leverages the encoding $\bar{\mathbf{h}}_j$ of the input $\mathcal{Y}_c$ and the embedding z_c^T to predict the metrics of interest. As explained earlier, we focus on four types of predictions, namely *future incidence*, *peak time*, *peak intensity*, and *onset*. We train our model for both point and binning-based probabilistic predictions. Let us first focus on the point predictions.

Task 1: Future Incidence Prediction: Here, given the current season $\mathcal{Y}_c = \{y_c^1, y_c^2, \ldots, y_c^t\}$, the goal is to predict y_c^i for $i \in \{t+1, t+2, t+3, t+4\}$. For simplicity we explain the training process for y_c^{t+1}. Our goal here is to learn function f_{next} that maps the encoding learned to the output $\hat{y} \in$ R, i.e. $\hat{y} = f_{next}(\bar{\mathbf{h}}_j, z_c^T)$.

We represent f_{next} as a feed-forward neural network. We train the network by leveraging the historical data Y. Hence, our objective function becomes $L_{pred} = \sum_{k \in Y} ||y_k^{t+1} - \hat{y}||_2^2$.

Task 2: Peak Intensity Prediction: Here the approach is similar to future incidence intensity prediction. Instead of training to predict the next incidence, we train the network to directly predict the peak intensity.

Task 3: Peak Time Prediction: As in previous metrics, the goal here is to leverage the encoding to predict peak time (in weeks). Here, the prediction process is slightly different. We have $x_t = \mathbf{W} f_{next}(\bar{\mathbf{h}}_j, z_c^T)$ and $P(t|x_t) = \frac{\exp(x_t)}{\sum_i \exp(x_i)}$ where $\mathbf{W}$ is a weight matrix and $f_{next}(\cdot)$ is represented as a feed-forward neural network. The term $P(t|x_t)$ represents the probability that the peak occurs at time t. As shown above, we use the softmax function to compute the probability. Finally the peak time is given by $\hat{t} = \arg\max P(t|x_t)$. As above, historical data Y is leveraged for the training. Here we adopt the cross-entropy as the objective function L_{pred}.

Task 4: Onset Prediction: We adopt similar approach as Peak Time prediction for the onset prediction.

The overall objective is given by:

$$\theta^* = \arg\min_\theta \left[L_{emd} + L_c^t + L_r^t + L_c^T + L_r^T + L_{pred} \right]$$

Note that the prediction loss L_{pred} is different for each task as mentioned above. We train separate networks for each task. We start by pre-training the clustering and mapping layers first and then jointly training the entire model. The adaptive moment estimation (Adam) optimization algorithm [16] was used to infer the model parameters. The model was coded using the automated differentiation package in PyTorch[4].

Note: We train for the CDC binning-based probabilistic predictions in a similar fashion. Instead of predicting point estimates, we assign probabilities to each bin pre-defined by the CDC.

4 EMPIRICAL STUDY

4.1 Setup

We describe our experimental setup in the appendix.

Data: Data is described briefly in Section 1 and in more detail in the appendix.

Baselines: While several methods exist for influenza epidemic forecasting, most of them require additional data such as twitter feeds, weather data, and so on. In contrast, EpiDeep forecasts given only the historical wILI data. Hence, we compare our performance against many non-trivial baselines which can forecast given only the wILI data.

- Hist: It is inspired by the traditional approach for flu forecasting. Here, we compute historical average of all previous seasons and make predictions using the average.
- ARIMA is a popular auto-regressive method typically used for prediction on time-series data. Here we leveraged ARIMA (7,0,1) as it performed the best.
- KNN: Here, we select the top k closest historical seasons to the current season and make predictions based on the average. Many model based approaches for flu forecasting [21] leverage a similar strategy of utilizing the closest historical season.
- LSTM: We leverage Long Short Term Memory network for forecasting. Note that it is a version of [31] without climate and geographical data and can be considered a simpler version of EpiDeep without the embeddings and attention.
- EB is an empirical bayes framework for epidemic prediction [4]. In this approach, translation of a historical season is fitted to the observed part of the current season to make a prediction. It is

[4] https://pytorch.org/docs/stable/autograd.html

Table 1: EpiDeep consistently performs well across all the tasks, outperforming all the methods in majority of the scenarios. Comparison of performance of all the methods for all the four tasks for seasons starting from 2010/11 till 2016/17. R is RMSE, M is MAPE and LS is the average Log-Score. A "–" means that the method can not be used for that prediction. For the 2011/12 season, the national wILI incidence curve did not cross the baseline, so there was no onset & we mark the cells with "×" signs.

	10/11			11/12			12/13			13/14			14/15			15/16			16/17		
Method	**R**	**M**	**LS**	**R**	**M**	**LS**	**R**	**M**	**LS**	**R**	**M**	**LS**	**R**	**M**	**LS**	**R**	**M**	**LS**	**R**	**M**	**LS**
Task 1: Future Incidence Prediction																					
Hist	1.29	0.4	39.82	0.44	0.21	49.57	1.4	0.36	46.92	0.77	0.23	54.88	1.12	0.26	61.96	0.65	0.23	39.0	0.9	0.22	53.12
ARIMA	0.65	**0.15**	–	**0.28**	**0.12**	–	0.89	**0.17**	–	0.65	**0.12**	–	0.88	0.17	–	0.42	0.13	–	0.67	0.15	–
KNN	0.76	0.26	57.32	1.57	0.71	75.11	0.81	0.24	75.97	1.06	0.37	76.86	**0.61**	0.24	75.1	0.98	0.36	80.41	0.65	0.2	77.75
LSTM	0.92	0.36	40.12	0.72	0.32	58.41	0.93	0.32	54.46	1.25	0.40	79.76	0.82	0.19	64.45	0.78	0.33	56.97	0.98	0.31	72.06
EB	0.81	0.31	43.29	0.97	0.5	60.11	1.04	0.24	65.42	0.67	0.24	58.37	0.87	0.21	61.8	0.93	0.39	47.79	1.06	0.32	56.61
EpiDeep	**0.59**	0.17	**26.61**	0.36	0.16	**32.2**	**0.68**	**0.17**	**29.89**	**0.45**	**0.12**	**36.03**	0.73	**0.15**	**41.01**	**0.41**	**0.13**	**29.75**	**0.58**	**0.15**	**35.15**
Task 2: Peak Intensity Prediction																					
Hist	1.39	0.31	∞	1.0	0.42	∞	2.55	0.42	∞	0.78	0.17	1.18	1.94	0.32	1.2	0.78	0.22	0.93	**0.54**	**0.11**	∞
ARIMA	2.47	0.52	–	0.64	0.25	–	4.01	0.65	–	2.76	0.6	–	3.93	0.65	–	1.55	0.41	–	2.82	0.54	–
KNN	1.31	0.28	∞	3.28	1.35	∞	0.61	0.1	∞	0.9	0.19	32.47	**0.18**	**0.03**	∞	1.49	0.4	53.73	0.58	0.09	∞
LSTM	1.43	0.32	89.91	1.35	0.56	77.84	1.94	0.51	56.92	0.84	0.17	22.41	2.83	0.42	0.94	1.42	0.37	43.77	1.98	0.38	95.41
EB	**0.96**	0.21	**60.73**	**1.21**	**0.48**	82.16	2.48	0.41	**42.86**	1.1	0.24	0.46	2.3	0.38	0.45	**0.24**	**0.06**	**11.24**	1.41	0.28	64.29
EpiDeep	0.99	**0.2**	71.4	1.59	0.6	**71.03**	**1.36**	**0.21**	71.43	**0.56**	**0.1**	**0.37**	2.26	0.37	**0.34**	0.46	0.11	18.64	0.87	0.15	**43.9**
Task 3: Peak Time Prediction																					
Hist	17.0	0.3	∞	21.0	0.33	∞	8.0	0.15	0.67	6.0	0.12	0.57	5.0	0.1	0.51	13.0	0.21	∞	7.0	0.12	0.95
ARIMA	33.53	0.58	–	37.1	0.58	–	27.3	0.51	–	27.08	0.51	–	26.93	0.5	–	38.69	0.62	–	34.8	0.59	–
KNN	12.0	0.21	∞	5.9	0.09	∞	16.82	0.32	0.47	16.82	0.32	0.33	11.0	0.21	0.14	**6.6**	**0.1**	∞	10.17	0.17	∞
LSTM	7.09	0.14	64.13	5.6	0.08	81.32	9.35	0.24	1.48	9.73	0.22	0.29	19.24	0.41	21.25	11.45	0.23	50.44	8.85	0.32	88.4
EB	1.09	**0.02**	60.7	**4.5**	**0.07**	78.6	**5.4**	**0.1**	**0.3**	**1.0**	**0.02**	**0.2**	**1.4**	**0.02**	**0.2**	8.04	0.11	75.0	**6.3**	**0.1**	64.2
EpiDeep	**1.0**	**0.02**	**33.2**	5.1	0.08	**29.2**	6.0	0.12	0.33	6.0	0.12	0.26	3.71	0.05	0.28	10.3	0.16	**29.6**	6.65	0.11	**21.6**
Task 4: Onset Prediction																					
Hist	23.0	0.43	∞	×	×	×	15.0	0.29	∞	12.0	0.24	0.57	8.0	0.16	∞	14.0	0.25	∞	6.0	0.12	∞
ARIMA	51.21	0.96	–	×	×	×	48.66	0.94	–	49.18	0.98	–	47.96	0.97	–	55.89	0.97	–	51.5	0.98	–
KNN	17.19	0.32	60.9	×	×	×	21.63	0.42	∞	23.0	0.46	0.16	19.0	0.39	∞	15.13	0.26	∞	19.96	0.38	60.9
LSTM	6.71	0.28	92.4	×	×	×	5.41	0.64	58.43	17.40	0.33	21.48	11.22	0.24	56.58	9.56	0.15	56.6	8.28	0.29	63.79
EB	**2.38**	**0.04**	∞	×	×	×	2.35	0.05	64.63	3.59	0.07	68.03	3.0	0.06	∞	8.04	**0.13**	47.01	3.67	0.07	61.15
EpiDeep	4.0	0.08	**31.99**	×	×	×	**2.0**	**0.04**	**17.47**	**0.88**	**0.02**	**0.0**	**0.35**	**0.0**	**24.33**	**7.76**	**0.13**	**40.28**	**2.92**	**0.06**	**24.66**

the publicly available version of the approach which has won several of the past iterations of the FluSight challenge.

Evaluation Metrics: There has been much discussion on the correct metric for evaluating models for epidemic predictions [28]. Hence we utilize the following multiple metrics to evaluate the predictive power of all the methods:

- **RMSE:** The root mean squared error is the square root of the average squared error i.e. RMSE $= \sqrt{\frac{\sum_{i=1}^{N} e_i^2}{N}}$.
- **MAPE:** The mean absolute percentage error measures the average of absolute percentage error, i.e. MAPE $= \frac{1}{N}\sum_{i=1}^{N}\left|\frac{e_i}{y_i}\right|$.
- **Log Score:** For the probabilistic predictions, we leverage the logarithmic scoring rule leveraged by the CDC [24]. The performance is measured by the negative log score, defined as $\log(p, i) = -\log(p_i)$ for the probability assigned in the bin i (containing the ground-truth).

4.2 National Predictions

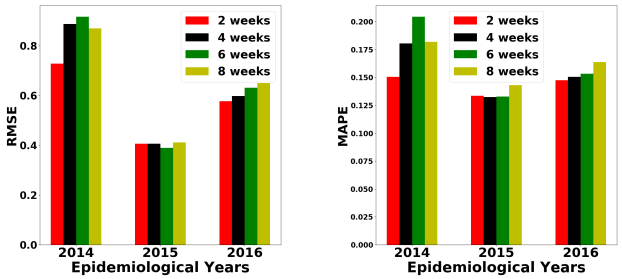
Here, we compare the performance of all the methods for all four tasks at the United States national level starting from 2010/11 till 2016/17. For the Future Incidence Prediction task, we ran all the methods to predict future wILI values starting from epidemiological week 40, when the flu season typically starts, till epidemiological week 20, when the season ends. For Peak (Intensity and Time) and Onset Prediction tasks, we predicted the metric starting from week 40 until it was observed for each season. The results for each method is summarized in Table 1. Since ARIMA does not produce probabilistic predictions, we were unable to compute the log-score.

As shown in the table, EpiDeep outperforms all the baselines in the majority of the settings. It actually outperforms non-trivial baseline EB in three of the four tasks, namely Peak Intensity, Onset, and Future wILI prediction by an impressive margin of 16%, 14%, and 40% on average in terms of RMSE. This is partly due to the fact that EB is constrained by a rigid base function, whereas EpiDeep is not. Overall, EpiDeep outperforms all the baselines in 17 out of 21 measures for Future wILI prediction, in 10 out of 21 measures for Peak Intensity Prediction, in 7 of 21 measures for Peak Time Prediction, and in 16 of the 21 measurements for Onset Prediction. It is a close second/third in the rest. Simpler baselines such as Hist, ARIMA, and KNN have reasonably satisfactory and stable performance for the Future Incidence Prediction task. However, the performance of these methods are at two different extremes for all other tasks. On the other hand, LSTM, EB, and EpiDeep have a stable performance across all tasks. Note that, EpiDeep outperforms LSTM in almost all measurements. This is due to the fact that EpiDeep has the flexibility of LSTM in addition to the meaningful embeddings which it can exploit for accurate forecasting.

4.3 Delayed Data Arrival

As mentioned in Section 1, the ILINet data has a delay of about 2 weeks. An interesting question is how the performance of EpiDeep varies with the delay. Will the performance of EpiDeep vastly vary if the delay is increased significantly? Will it remain stable?

To answer these questions, we performed experiments with simulated larger delays. Specifically, we leveraged EpiDeep to forecast future wILI incidence with delay of 2, 4, 6, and 8 weeks. We repeated the experiments for three seasons, namely 2014/15, 2015/16, and 2016/17. Since peak (time/intensity) predictions and onset prediction already have a large gap between the time when prediction is made and the time when the data is observed, we conducted this study only for future incidence prediction for the national data.

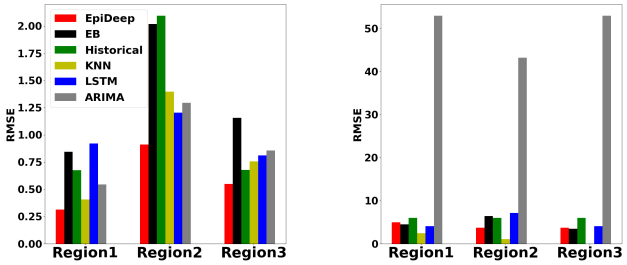


(a) RMSE (b) MAPE

Figure 4: RMSE and MAPE for future wILI incidence predictions for delayed data arrival. EpiDeep's performance remains stable even when data is delayed by up to 8 weeks.

The result is summarized in Figure 4. As shown in figure, there is minimal change in performance of EpiDeep even when the data is delayed by up to 8 weeks. It highlights the fact that EpiDeep makes stable predictions even in scenarios where there is a bigger delay in data arrival.

4.4 Regional Forecasting



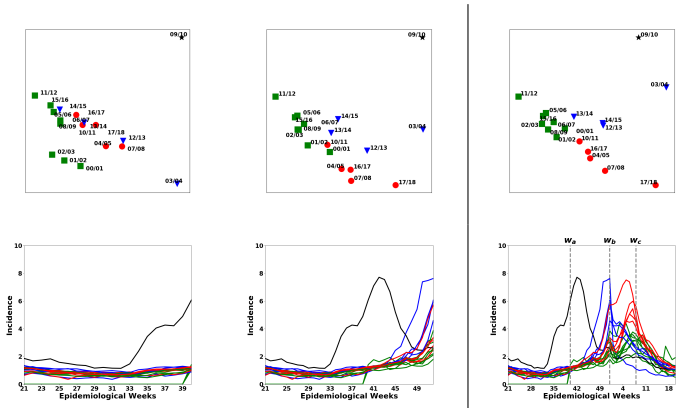
(a) Future Incidence Prediction (d) Onset Prediction

Figure 5: RMSE for regional predictions of two of the tasks for 2016/17 season. EpiDeep consistently performs well. The figure is best viewed in colour.

The US Department of Health and Human Services has divided the country into 10 regions, commonly referred to as the HHS regions. The CDC reports ILINet wILI values for each of these regions individually as well. Here we leverage all methods for influenza forecasting for different regions. For different regions the influenza pattern can be different. We want to see if EpiDeep and other methods can detect these differences and perform well in each regions. Hence, here we leveraged all the methods for all four tasks in all the regions.

In summary, we observe that EpiDeep consistently performs well in all predictions. For space, we report RMSE results for the 2016/17 season only for HHS regions 1, 2 and 3 for Future Incidence and Onset prediction tasks. We observe similar results in all other tasks across other metrics. See Figure 5: EpiDeep outperforms all the baselines in future incidence prediction in all three HHS regions. Similarly, we observe that the all the methods except ARIMA perform well in onset prediction for all three regions. EpiDeep's performance is competitive in all three regions.

4.5 Interpretability



(a) $w_a = 40$ (b) $w_b = 52$ (c) Whole Season

Figure 6: (c)-bottom row shows full season length wILI curve for all historical seasons. Bottom row (a) and (b) shows snippets of the historical seasons till week w_a and w_b respectively. The top row shows 2-d projection of learnt embeddings of corresponding snippets.

An important advantage of EpiDeep is that its embedding/clustering components help in interpretability of the forecasts. Here we demonstrate how to leverage the embeddings learnt by EpiDeep for interpreting the influenza forecasting. Our experiments are designed to answer questions that epidemiologists and authorities like the CDC have. We focus on the following questions:

- *Question 1:* Can we infer 'clusters' of historical seasons based on their incidence curve even when partially observed?
- *Question 2:* What relationship can we infer between different HHS regions across multiple seasons?
- *Question 3:* Which historical season is closest to the current 'query' season at the time of forecasting? Does the closest season evolve over time as more data is observed?

4.5.1 Question 1: Qualitative Evaluation of the Embeddings and Inferred Clustering. We can leverage EpiDeep to learn the embeddings of the historical seasons. Figure 6 shows the 2-d projection of the embeddings generated by EpiDeep at various weeks w (when partial data is observed) in the top row and corresponding incidence curves in the bottom row. The colors of the markers in the top plot represent the cluster memberships. For each cluster, the same color is used to draw the incident curves in the bottom row. The top row in Fig 6 (c) shows the embeddings generated when the complete data is observed. From the figure, we can make the following key observations.

Observation 1. ***Season Clusters:*** EpiDeep *embeddings showcase different meaningful clusters of wILI trends.*

The first cluster (in black) has only one member, the 2009/10 H1N1 pandemic season. Clearly from the incidence curves (bottom row Fig 6 (c)), it is obvious that the 2009/10 season was very different than the rest [5]. The cluster in green, representing seasons such as seasons 2011/22, 2005/06, 2015-14, and so on, have a distinct characteristics: they all have a low peak (green curves in (bottom row Fig 6 (c)). As reported by the CDC these were the only seasons to peak in March [6]. The defining characteristic of the blue cluster is that the seasons peak relatively early (late December/early January) and have high intensity. Finally, we observe that the seasons in the red clusters have a late peak and a high intensity. These intuitive clustering of the seasons shows that EpiDeep learns meaningful embeddings of the historical seasons.

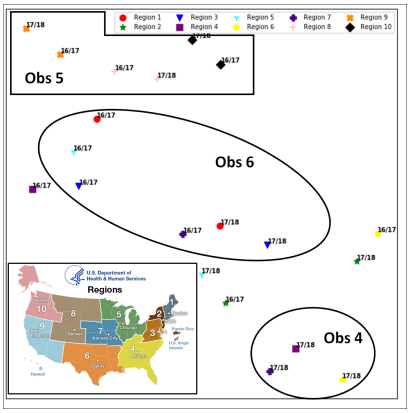


Figure 7: 2-d projections of embeddings of 2016/17 and 2017/18 seasons' wILI curves for all 10 HHS regions (inset).

4.5.2 Question 2: Regional and Seasonal Embeddings. In the previous question, we explored the quality of embeddings and their evolution. Here, we study the embeddings generated by EpiDeep for each HHS region over two seasons (2016/17 and 2017/18). We are interested in questions like does EpiDeep capture meaningful geographical relationship between HHS regions? Which regions are similar to each other and so on. The 2-d projections of the embeddings learnt by EpiDeep is shown in Figure 7. The first observation we make from the figure is as follows.

Observation 2. ***Neighbor Similarities:*** *Learnt embeddings reveal neighboring HHS regions have very similar incidence curves.*

The neighboring regions 4, 6, and 7 witnessed a very similar influenza season in 2017/18. It turns out all three of these regions peaked at week 4 and had a very high peak intensity. Similar observations were made by prior works [20] for multiple diseases. Geographical correlation between HHS regions are leveraged for forecasting as well [20, 31]. Another nontrivial observation from the embeddings is that the incidence curve for some of the non-neighboring seasons are similar to each other.

Observation 3. ***Long Distance Similarities:*** EpiDeep *embeddings discover geographically distant regions having similar influenza incidence curves for multiple seasons.*

We observe that regions 1, 3, and 5 are embedded close to each other for both seasons. Their similarity is explained by the fact that all three seasons saw the influenza intensity peak at week 6 and 7 for 2016/17 season and peak at week 6 for 2017/18 season. Similarly, all three regions saw significant rise in the peak intensity in 2017/18 season as compared to that of 2016/17 season.

4.5.3 Additional Observations: We make the following interesting observations additional to the ones discussed. Observation 4 is with respect to Question 1, Observation 5 is with respect to Question 2, and Observation 6 is with respect to Question 3. Due to the lack of space, we omit detailed discussion on the significance of these observations.

Observation 4. ***Intensity Separation:*** EpiDeep *embeddings distinguish seasons with higher intensities from the ones with lower intensities.*

Observation 5. ***Temporal Similarities:*** *The learnt embeddings reveal temporal similarities between different seasons in the same region.*

Observation 6. ***Evolution of Season Similarity:*** *The similarity/distances between the seasons evolve as more data is observed and* EpiDeep *is able to capture this phenomenon.*

Note that all the observations, from 1 to 6, are made directly from the embeddings learned by EpiDeep. It is quite challenging to extract all these patterns from the raw incidence curves as it is hard to compare all possible snippets of all the historical curves in each region.

5 RELATED WORK

Epidemic Forecasting: In addition to the closely related works discussed earlier, there are other statistical [9, 30] and modelling based approaches [27, 35] for flu forecasting, which suffer from the challenges discussed before. Additionally, orthogonal to this paper, there has also been much interest in leveraging signals from external data sources such as search engine [12, 34], social media [8, 17], environmental and weather reports [26, 29], and a combination of heterogeneous data [7]. Deep learning for flu forecasting has barely been explored except for [31] which basically uses a simple LSTM with geographical and climate constraints and [32] which uses LSTM to predict influenza activities specifically in the military population by incorporating twitter data. As we saw in Section 4, LSTM does not perform well as it requires a large amount of data. In contrast, we give a novel architecture which ensures excellent performance even with sparse data.

Time Series Analysis: Time-series prediction is a well-studied area with several methods from different perspectives including auto-regression, kalman-filters and groups/panels [3, 15, 25]. Recently recurrent neural architectures [10, 14] have also become popular. However these prediction methods are ill suited for flu forecasting as they are too specialized or usually not flexible enough to capture the seasonal inconsistency in wILI activity [22]. In contrast, we design an end-to-end approach which automatically embeds, clusters, and forecasts giving it the flexibility to capture the seasonal inconsistency in the data.

6 DISCUSSIONS AND CONCLUSIONS

Here we proposed a novel deep learning model EpiDeep to learn feature representations of historical epidemic seasons in conjunction with the observed current season and leveraged it for four epidemic forecasting tasks. We compared the performance of EpiDeep against multiple baselines on extensive historical data and showed that it outperforms non-trivial baselines by up to 40%. It also promises gains in *interpretability*. The embeddings learnt by EpiDeep are meaningful and non-trivial. We also observed that these embeddings evolve as the season progresses to capture the most meaningful relationship between the historical seasons.

Our method was designed to overcome specific challenges in influenza forecasting like the data sparsity issue and leveraging some domain knowledge for interpretability, but it also flexible and extensible. Due to its modular neural structure, as future work, we believe it has the potential to be useful in overcoming other challenges in a systematic manner as well. For example, since EpiDeep uses an end-to-end representation learning based framework, we can try to learn to jointly embed multiple heterogeneous data sources in addition to ILINet (say social media, weather data etc) and leverage these embeddings for prediction. We can also try to directly take data from epidemiological models as inputs into our model. Further, usually ILI data has geographical structure (e.g. flu incidence in nearby states would be expected to be similar [18]). These types of constraints can also be explicitly codified in the loss functions of the predictor module of EpiDeep (though notably, as we saw, it discovers many of these relationships automatically). We believe our techniques of using embedded clustering for forecasting can help with other sparse time-series data as well.

Acknowledgments. This paper is based on work supported by the NSF (CAREER IIS-1750407, DGE-1545362, and IIS-1633363), the NEH (HG-229283-15), ORNL and a Facebook faculty gift.

REFERENCES

[1] Matthew Biggerstaff, David Alper, Mark Dredze, Spencer Fox, Isaac Chun-Hai Fung, Kyle S Hickmann, Bryan Lewis, Roni Rosenfeld, Jeffrey Shaman, Ming-Hsiang Tsou, and others. 2016. Results from the centers for disease control and prevention's predict the 2013–2014 Influenza Season Challenge. *BMC infectious diseases* 16, 1 (2016), 357.

[2] Matthew Biggerstaff, Michael Johansson, David Alper, Logan C Brooks, Prithwish Chakraborty, David C Farrow, Sangwon Hyun, Sasikiran Kandula, Craig McGowan, Naren Ramakrishnan, and others. 2018. Results from the second year of a collaborative effort to forecast influenza seasons in the United States. *Epidemics* (2018).

[3] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. 2015. *Time series analysis: forecasting and control*. John Wiley & Sons.

[4] Logan C Brooks, David C Farrow, Sangwon Hyun, Ryan J Tibshirani, and Roni Rosenfeld. 2015. Flexible modeling of epidemics with an empirical Bayes framework. *PLoS computational biology* 11, 8 (2015), e1004382.

[5] CDC. 2010. Summary of the 2009-2010 Influenza Season. https://www.cdc.gov/flu/pastseasons/0910season.html. (2010). Accessed: 2018-11-05.

[6] CDC. 2016. Summary of the 2015-2016 Influenza Season. https://www.cdc.gov/flu/about/season/flu-season-2015-2016.html. (2016). Accessed: 2018-11-05.

[7] Prithwish Chakraborty, Pejman Khadivi, Bryan Lewis, Aravindan Mahendiran, Jiangzhuo Chen, Patrick Butler, Elaine O Nsoesie, Sumiko R Mekaru, John S Brownstein, Madhav V Marathe, and others. Forecasting a moving target: Ensemble models for ILI case count predictions. In *SDM 2014*. SIAM, 262–270.

[8] Liangzhe Chen, KSM Tozammel Hossain, Patrick Butler, Naren Ramakrishnan, and B Aditya Prakash. 2016. Syndromic surveillance of Flu on Twitter using weakly supervised temporal topic models. *Data mining and knowledge discovery* 30, 3 (2016), 681–710.

[9] Rumi Chunara, Edward Goldstein, Oscar Patterson-Lomba, and John S Brownstein. 2015. Estimating influenza attack rates in the United States using a participatory cohort. *Scientific reports* 5 (2015), 9540.

[10] Jerome T Connor, R Douglas Martin, and Les E Atlas. 1994. Recurrent neural networks and robust time series prediction. *IEEE transactions on neural networks* 5, 2 (1994), 240–254.

[11] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. 1999. Learning to forget: Continual prediction with LSTM. (1999).

[12] Jeremy Ginsberg, Matthew H Mohebbi, Rajan S Patel, Lynnette Brammer, Mark S Smolinski, and Larry Brilliant. 2009. Detecting influenza epidemics using search engine query data. *Nature* 457, 7232 (2009), 1012.

[13] Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. Improved deep embedded clustering with local structure preservation. In *IJCAI 2017*. 1753–1759.

[14] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[15] Abhay Jha, Shubhankar Ray, Brian Seaman, and Inderjit S Dhillon. Clustering to forecast sparse time-series data. In *ICDE 2015*. IEEE, 1388–1399.

[16] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[17] Kathy Lee, Ankit Agrawal, and Alok Choudhary. Real-time disease surveillance using twitter data: demonstration on flu and cancer. In *ACM SIGKDD 2013*. ACM, 1474–1477.

[18] Fred S Lu, Mohammad W Hattab, Cesar Leonardo Clemente, Matthew Biggerstaff, and Mauricio Santillana. 2019. Improved state-level influenza nowcasting in the United States leveraging Internet-based data and network approaches. *Nature communications* 10, 1 (2019), 147.

[19] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015).

[20] Yasuko Matsubara, Yasushi Sakurai, Willem G Van Panhuis, and Christos Faloutsos. FUNNEL: automatic mining of spatially coevolving epidemics. In *ACM SIGKDD 2014*. ACM, 105–114.

[21] Elaine O Nsoesie, Richard Beckman, Madhav Marathe, and Bryan Lewis. 2011. Prediction of an epidemic curve: A supervised classification approach. *Statistical communications in infectious diseases* 3, 1 (2011).

[22] 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 and other respiratory viruses* 8, 3 (2014), 309–316.

[23] You Ren, Emily B Fox, Andrew Bruce, and others. 2017. Clustering correlated, sparse data streams to estimate a localized housing price index. *The Annals of Applied Statistics* 11, 2 (2017), 808–839.

[24] Roni Rosenfeld, John Grefenstette, and Don Burke. 2012. A Proposal for Standardized Evaluation of Epidemiological Models. (2012).

[25] Nicholas I Sapankevych and Ravi Sankar. 2009. Time series prediction using support vector machines: a survey. *IEEE Computational Intelligence Magazine* 4, 2 (2009).

[26] Jeffrey Shaman, Edward Goldstein, and Marc Lipsitch. 2010. Absolute humidity and pandemic versus epidemic influenza. *American journal of epidemiology* 173, 2 (2010), 127–135.

[27] Jeffrey Shaman and Alicia Karspeck. 2012. Forecasting seasonal outbreaks of influenza. *Proceedings of the National Academy of Sciences* 109, 50 (2012), 20425–20430.

[28] Farzaneh Sadat Tabataba, Prithwish Chakraborty, Naren Ramakrishnan, Srinivasan Venkatramanan, Jiangzhuo Chen, Bryan Lewis, and Madhav Marathe. 2017. A framework for evaluating epidemic forecasts. *BMC infectious diseases* 17, 1 (2017), 345.

[29] James D Tamerius, Jeffrey Shaman, Wladmir J Alonso, Kimberly Bloom-Feshbach, Christopher K Uejio, Andrew Comrie, and Cécile Viboud. 2013. Environmental predictors of seasonal influenza epidemics across temperate and tropical climates. *PLoS pathogens* 9, 3 (2013), e1003194.

[30] Michele Tizzoni, Paolo Bajardi, Chiara Poletto, José J Ramasco, Duygu Balcan, Bruno Gonçalves, Nicola Perra, Vittoria Colizza, and Alessandro Vespignani. 2012. Real-time numerical forecast of global epidemic spreading: case study of 2009 A/H1N1pdm. *BMC medicine* 10, 1 (2012), 165.

[31] Siva R Venna, Amirhossein Tavanaei, Raju N Gottumukkala, Vijay V Raghavan, Anthony Maida, and Stephen Nichols. 2017. A novel data-driven model for real-time influenza forecasting. *bioRxiv* (2017), 185512.

[32] 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.

[33] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML 2016*. 478–487.

[34] Qingyu Yuan, Elaine O Nsoesie, Benfu Lv, Geng Peng, Rumi Chunara, and John S Brownstein. 2013. Monitoring influenza epidemics in china with search query from baidu. *PloS one* 8, 5 (2013), e64323.

[35] Qian Zhang, Nicola Perra, Daniela Perrotta, Michele Tizzoni, Daniela Paolotti, and Alessandro Vespignani. Forecasting seasonal influenza fusing digital indicators and a mechanistic disease model. In *WWW 2017*. 311–319.

Appendix for EpiDeep: Exploiting Embeddings for Epidemic Forecasting

Bijaya Adhikari*, Xinfeng Xu†, Naren Ramakrishnan* and B. Aditya Prakash*
*Department of Computer Science, Virginia Tech.
†Department of Physics, Virginia Tech.
Email: *[bijaya, naren, badityap]@cs.vt.edu, †xinfeng@vt.edu

1 EXPERIMENTAL SETUP

All experiments are conducted using a 4 Xeon E7-4850 CPU with 512GB of 1066 Mhz main memory.

1.1 Data and Code

The code, implemented in PyTorch, is publicly available[1]. We used wILI surveillance data collected and released by the CDC. The data is publicly available[2]. For each prediction reported, only the historical data observed prior to the time of prediction is leveraged. No other source of data are used in our experiments.

1.2 Setup Details

Note that we use historical wILI incidence $Y = \{\mathcal{Y}_1, \mathcal{Y}_2, \mathcal{Y}_3, \ldots, \mathcal{Y}_{c-1}\}$ to train EpiDeep and use it predict various metrics of interest for the current season $\mathcal{Y}_c$. In our settings, $\mathcal{Y}_c$ is observed till week t whereas the historical seasons in Y are fully observed. All the experiments in the paper follow the setup described here unless mentioned otherwise.

The following setup was used for experiments for all the methods including EpiDeep:

- While forecasting for season $\mathcal{Y}_i$, only the historical incidence data till season $i-1$ were leveraged for training.
- For each season $\mathcal{Y}_i$, all the methods were leveraged to forecast various metrics starting from week 40.
- For onset, peak, and peak-time prediction tasks, predictions were made till they were observed. Future incidence prediction was made till week 20 of the following year.
- For EpiDeep, hyper-parameter search was done before the final model was selected.

1.3 Delayed Data Arrival

Experiments on delayed data arrival was conducted on the Future Incidence Prediction task only. We use EpiDeep to forecast with simulated delay of 2, 4, 6, and 8 weeks. For each season y_i, the experiment began on week 40 and ended on week 20 the following year.

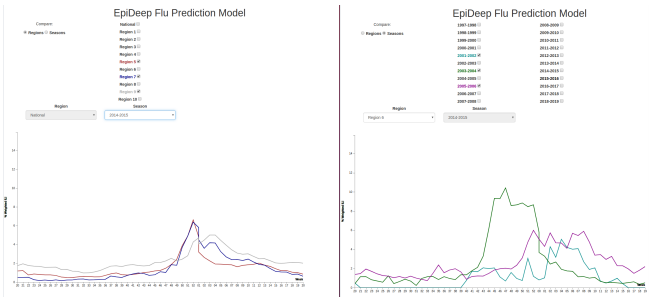
1.4 Regional Forecasting

For each region's prediction, we used the historical data from that particular region only, instead of using the data from all regions for EpiDeep. The same approach was used for other baselines as well.

1.5 Interpretability

The embeddings presented in the paper were obtained using the query length data $Y_t = \{\mathcal{Y}_i[0:t] | \forall_{i=1}^{c}\} \cup \{\mathcal{Y}_c\}$. Once EpiDeep is trained, we feed Y_t to the Query Length Data Clustering module and then feed the encoded representations of Y_t to the embedding mapper. The output of the embedding mapper is the final embedding.

2 PROJECT WEBSITE



(a) Incidence in various regions in 2014/15 season

(b) Various seasons in Region 6

Figure 1: One of the features of the project website is that it enables users to compare incidence curves across regions and seasons.

The project website[3] is under construction. The website allows users to explore the historical wILI data, compare different incidence curves (see Figure 1), and to follow EpiDeep's live forecasts.

[1] https://github.com/epideep/source
[2] https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html
[3] https://epideep.github.io