

# PatientFlowNet: A Deep Learning Approach to Patient Flow Prediction in Emergency Departments

ALI R. SHARAFAT<sup>1</sup> AND MOHSEN BAYATI<sup>2</sup>

<sup>1</sup>Department of Electrical Engineering, Stanford University, Stanford, CA 94305, USA

<sup>2</sup>Graduate School of Business, Stanford University, Stanford, CA 94305, USA

Corresponding author: Ali R. Sharafat (sharafat@stanford.edu)

**ABSTRACT** Emergency Department (ED) crowding is a major public health challenge since it can seriously impact patient outcomes; and accurate prediction of patient flow in EDs is essential for improving operational efficiency and quality of care. We present a deep learning framework to predict patient flow rates in EDs, namely the rates of arrival, treatment, and discharge for patients across all triage levels. Our model detects short-term and long-term temporal dependencies within the time-series data of a given patient-flow variable, as well as dependencies between time-series data of different patient-flow variables. We implement this framework as a convolutional neural network, which we call PatientFlowNet. Our proposed model learns simultaneously from multiple flow variables over a long temporal window and predicts future values of arrival, treatment, and discharge rates in the ED. We benchmark our model against state-of-the-art methods on data from EDs in three different hospitals. Results show that PatientFlowNet achieves superior prediction accuracy, compared to the baseline methods, and yields a mean absolute error that is 4.8% lower than the leading baseline. Furthermore, we provide a visual and interpretable representation of the learned dependencies by our model, between patient-flow variables in EDs.

**INDEX TERMS** Health information management, machine learning, neural networks, public healthcare, supervised learning.

## I. INTRODUCTION


Emergency Department (ED) crowding is a public health challenge [1], [2]. It is caused by external factors, such as fluctuations in patient arrivals, and by internal factors, such as lack of available beds or unexpected human delays [3]. This problem is also exacerbated by unexpected external shocks such as a global pandemic [4], [5] due to unusually increased patient flow [6], [7]. In such circumstances, the need to rapidly modify admission procedures [8], [9] demands accurate forecast of the patients' arrival rate and their movements through the system until their discharge.

A large body of research has been dedicated to prediction of ED crowding [10]–[12] and to mitigate its effects [3], [13]. Sources of ED crowding include unexpected large volume of patients arriving at the ED, delays in triaging and starting their treatment, and impediments in discharging patients whose treatment has been completed. Accurate prediction of arrival, treatment, and discharge rates helps understand when and

where such crowding occurs, leading to efficient allocation of resources or development of cost-effective interventions to streamline the processes.

Accurate estimation of flow rates can also help predicting other ED workflow variables, such as time-to-treatment and length-of-stay [14] when the ED is modeled as a queueing system. In [15], flow variables that mimic arrival and treatment rates are combined with machine learning to estimate time-to-treatment for each patient, and in [16], [17], a queueing theoretic version of that approach is used to estimate the length-of-stay for each patient. Approaches that model the ED as a queueing system are hindered by unexplained queue pre-emption and delays (see Section III). Thus, a more aggregated approach where per-interval patient-flow rates are considered instead of per-patient timelines is more desirable.

Most studies on prediction of patient-flow rates use either time-series methods such as variations of autoregressive integrated moving average (ARIMA) [18]–[21], or use parameters that encode some seasonality information (such as day of the week or moving window average) and perform a regression [22], [23]. Such models either have a rigid structure

The associate editor coordinating the review of this manuscript and approving it for publication was Hualong Yu .

or require manual feature engineering. ARIMA's use of a small number of *difference terms* leads to only short-term dependencies being considered, while the use of manually encoded parameters leads to loss of information prior to the application of regression. Such deficiencies suggest that a more general model that does not require feature engineering (like deep learning models) may be more suitable.

Deep learning models are more generalized than classical machine learning models and can better capture the dependencies when data is not directly amenable to feature engineering is. In fact, recurrent neural networks (RNNs) have been used to predict patient-flow rates with mixed degrees of success [24]–[26]. The models used in these papers are vanilla version of long short-term memory (LSTM) networks and they outperformed baseline time-series methods, as they used sequence input data and can learn from a large temporal window. Nevertheless, they suffer from two main shortcomings: (1) slow training and (2) lack of cross-learning from multiple time-series data. The latter is significant, as a patient flow variable in the ED depends on other factors, besides its own history (see Section III).

Convolutional models that use dilations, such as WaveNet [27], offer a viable solution to the problem (1) compared to RNNs, as they can be trained faster. In addition, exponentially large receptive fields of WaveNets enables them to learn long-term dependencies. The size of such models increases in the log order of dilation size, i.e., they can capture long term dependencies without being excessively large. In [28], convolutional models are outperform RNNs on a variety of sequence modeling tasks. Variations of WaveNet are used in other time-series tasks, e.g., in [29], parametrized skip connections are added to WaveNet to extract dependencies between financial tickers; in [30], pre- and post-processing tools in addition to WaveNet are used to estimate blood glucose levels; and in [31], interleaving dilated temporal and spatial convolutions are proposed to classify multiple time-series variables.

However, WaveNet is very effective in single variable sequence-prediction tasks such as audio synthesis, it only uses historical values of a single variable to predict its future values, not addressing the aforementioned shortcoming (2) of RNNs. In this paper, we address this by proposing a deep learning model that can *simultaneously* learn temporally (from the history of a flow variable) and spatially (by extracting dependencies between different flow variables). We implement this model as a convolutional neural network, which we call PatientFlowNet. This model relies on stacks of *flow convolutions*, i.e., 2-dimensional convolutional filters that are exponentially dilated in time. This enables our model to learn from multiple patient-flow variables in a large temporal window. Furthermore, we will show that the convolutional filters in our design allow for interpretability by visualizing the dependencies between patient-flow variables in EDs.

Using ED data from electronic medical records of three different hospitals, we will show that PatientFlowNet outperforms the state-of-the-art supervised learning methods in

one-step-ahead prediction of patient-flow rates. We also use our model to extract dependencies between different patient-flow variables in EDs.

The remainder of this paper is organized as follows. We discuss the existing methods for patient-flow prediction in Section II. Section III contains problem statement and describes ED workflow as well as variables-of-interest in the dataset. In Section IV, we develop a mathematical model for multivariate time-series forecasting and describe how it manifests into PatientFlowNet as a deep learning model. We describe our dataset in Section V, and in Section VI, we benchmark PatientFlowNet against existing methods by using data from 3 different EDs. We conclude with remarks in Section VII.

## II. OTHER RELATED WORKS

**Classical Machine Learning Models:** Classical machine-learning approaches to patient-flow prediction have been mostly centered around variations of ARIMA, exponential smoothing, and regression [22], [23], [32], [33]. In [18], flow data from one ED in Brazil was used to show that simple seasonal exponential smoothing was the most accurate at jointly predicting arrival rates for all triage levels, while seasonal ARIMA was more accurate at predicting arrival rates for a specific triage level. In [34], a hybrid model of ARIMA and linear regression was developed that outperforms variations of either model in predicting arrival rates at two EDs in China. In [19], flow data from one ED in China was used to show that a combination of single exponential smoothing and seasonal ARIMA is more accurate than either model alone. In [21], a host of machine learning methods were used on data from one ED in Portugal to show that seasonal ARIMA outperforms moving window average and exponential smoothing in predicting daily arrival counts. The main drawback of the models in the above cited works is that their input size is small, meaning that either predictions were based on a short temporal window or manual feature engineering was needed. As we will show next, more flexible models that can use a larger input size can provide more accurate predictions.

**Deep Learning Models:** Feed forward and recurrent neural networks have also been used to predict patient flow in hospitals. In [35], a hybrid model of ARIMA and feed forward neural network was developed that outperforms ARIMA, linear regression, or a hybrid of both in predicting arrival rates in one ED in Turkey. In [36], a feed forward network whose feature selector is based on a genetic algorithm was developed that outperforms a host of models including ARIMA and linear regression in predicting flow rates in one ED in Hong Kong. In [25], it was noted that random forest is more accurate than long short-term memory (LSTM) in predicting discharge rates, as it can combine data from multiple flow variables. An extensive study of neural network models was done in [24], where it was shown that a sequence-to-sequence LSTM which uses a long history of flow variables outperforms a host of baselines including seasonal ARIMA, linear regression, and feed forward network in predicting

arrival rates in one pediatric clinic. The two drawbacks of this approach is that LSTM does not combine information from multiple variables and is slow to train. In contrast, our model can provide more accurate predictions by learning from multiple variables, while having faster training due to its convolutional design.

While convolutional models have been used for other time-series prediction and classification tasks [27]–[29], [31], they have not been used for patient flow prediction. Our convolutional model considers dependencies between multiple patient-flow variables and utilizes their long-term history to provide accurate predictions.

### III. BACKGROUND AND PROBLEM STATEMENT

#### A. ED WORKFLOW

Patient flow in EDs is shown in Figure 1. Once a patient arrives at the ED, she is first registered (arrival time) and then triaged (triage time), where she is assigned a triage level from 1 to 5 denoting the acuity of her condition. Level 1 indicates the highest and level 5 denotes the least acuity. The patient then waits in a first-come-first-serve (FCFS) queue to start her treatment (treatment time). High acuity patients (triage levels 1 and 2) are given preemptive priority over low acuity patients (triage levels 3 to 5) and usually jump to the front of the queue. Hospitals in this study implement a separate fast-track queue during certain periods of the day, whereby they process patients who require minimal treatment (usually triage levels 4 and 5) separately to reduce the overall wait time. Fast-track is in effect from 8am to 11pm daily. Once treatment of a patient is completed, she is discharged and leaves the ED.

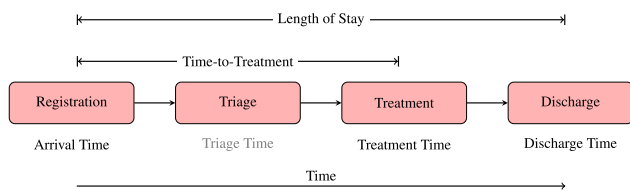


FIGURE 1. Patient flow in EDs. Triage time is not available.

From the above, EDs may appear to be simple queueing systems, but our data in Section V indicates that this paradigm is not closely followed. Table 1 shows the probability of a patient jumping ahead of the line, grouped by expected scenarios. Red cells correspond to high acuity patients being admitted, where the probability of jumping ahead is *expected* to be close to 1. Yellow cells correspond to patients admitted via fast-track. Since fast-track is in operation only during 8am to 11pm of weekdays, we *expect* these values to be somewhere between 0 and 1. Green cells correspond to patients not expected to jump ahead, and values are *expected* to be near zero.

Red cells have generally larger values than the rest, but there are numerous unexplained violations of the aforementioned expectations. Factors such as unavailability of facilities or deteriorating condition of a waiting patient may be

TABLE 1. The value in row  $i$ , column  $j$  denotes the probability that in Hospital 1 an arriving patient in triage level  $i$  will be served ahead of a patient in triage level  $j$  who has arrived earlier and is in the queue.

	1	2	3	4	5
1	0.25	0.67	0.76	0.75	0.65
2	0.32	0.37	0.70	0.71	0.59
3	0.12	0.11	0.28	0.34	0.28
4	0.12	0.11	0.27	0.25	0.15
5	0.25	0.19	0.42	0.37	0.19

conducive to such violations. The above examples indicate that a model-based (e.g., queueing model) approach in which per-patient flow metrics are considered may not work well because violations of the model (e.g., preemptions) are unexplained. Thus, we focus on a more data-driven approach for prediction of patients’ arrival, treatment, and discharge rates in fixed-length consecutive intervals.

#### B. PROBLEM STATEMENT

We wish to accurately forecast patient flow in EDs. Specifically, our focus is on predicting future values of arrival ( $\lambda$ ), treatment ( $\mu$ ), and discharge ( $\delta$ ) rates by utilizing historical values of these flow variables. The term *rate* is defined as the raw count of patients in a specific triage level for a fixed interval of 1 hour. For instance,  $\mu_{i,t}$  is the number of patients of triage level  $i$  who are treated in the 1 hour interval of  $t - 1$  to  $t$ . Our aim is to predict the aforementioned rates for all triage levels at time  $t + 1$ , using the history of all rates at all triage levels in a fixed window of length  $k$ . Suppose we have a flow variable  $\alpha$  that we wish to predict, where  $\alpha \in \{\lambda, \mu, \delta\}$ . Given a set  $S = \{(x_t, y_{t+1}) | t_1 \leq t \leq t_2\}$ , where  $x_t = \lambda_{1,t}, \dots, \lambda_{5,t}, \mu_{1,t}, \dots, \mu_{5,t}, \delta_{1,t}, \dots, \delta_{5,t}$  and  $y_{t+1} = \alpha_{1,t+1}, \dots, \alpha_{5,t+1}$ , we want to find the mapping  $f^* : \mathbb{R}^{15} \rightarrow \mathbb{R}^5$  that minimizes the loss. That is,

$$f^* = \arg \min_f \sum_{(x,y) \in S} g(f(x), y),$$

where  $g$  is a loss function. This enables us to predict the number of patients in different triage levels who arrive, are treated, and are discharged from the ED in the next hour. The one-hour-ahead prediction window is sufficient for predicting short-term ED workflow variables such as wait times. For longer-term predictions, one can continue feeding the predicted values back into the model to obtain farther estimates as is done in sequence-to-sequence models.

### IV. METHOD

The ED has no control over the arrival rates and once a patient arrives, it is not possible to turn her away. Thus, the arrival rate is *exogenous* to the system, but the treatment and discharge rates depend on the system, and are therefore *endogenous*. In what follows, we first develop a model to predict future values of an exogenous variable (e.g.,  $\lambda$ ) by using its own history. We then expand this model to predict future values of an endogenous variable (e.g.,  $\mu$  and  $\delta$ ) from the history of multiple time-series variables. Finally, we explain how this model is implemented as a neural network, namely PatientFlowNet.

### A. PREDICTING EXOGENOUS VARIABLES

Let  $a$  be an integer valued time-series variable. We wish to predict its value at time  $t + 1$ , denoted by  $a_{t+1}$ , from its  $k$  previous observations. One can use conditional probability  $p(a_{t+1}|a_{t-k+1}, \dots, a_t)$  to predict  $a_{t+1}$ . By sliding a window of length  $k$  over the time-series data, one can estimate the conditional probability distribution (CPD). Let  $N = \max_i(a_i)$  denote the upper bound of  $a$ . There are  $N^k$  possible combinations of values  $a_{t-k+1}, \dots, a_t$ . At the same time,  $k$  must be large to observe a long enough history. Hence, a fully non-parametric approach to CPD estimation requires exponentially large data (scaling with  $N^k$  for a large  $k$ ), which is infeasible.

To mitigate the above mentioned infeasibility, we use a parametric paradigm in which parameters are inter-related via a tree structure. We first model  $a_{t+1}$  as a linear combination of past observations of  $a$ , i.e.,

$$a_{t+1} = \sum_{i=1}^k \theta_i a_{i+t-k}, \quad (1)$$

where the values of  $\theta_i$  are to be learned. This approach is computationally more efficient in the sense that it only needs  $k$  parameters instead of  $N^k$  parameters to predict  $a_{t+1}$ , but may be less accurate when the future value is not best modeled by a linear combination of past observations.

Since the size of  $\theta$  scales with  $k$ , if one aims to look at a large enough (e.g.,  $2^{11}$ ) window of past variables, it soon becomes infeasible. Thus, in what follows, we propose a second assumption to reduce the number of parameters representing  $\theta$ . A linear combination can be viewed as applying a 1-dimensional convolution filter in a neural network with no padding, so we write (1) as  $\theta \otimes a$ , where  $\theta, a \in \mathbb{R}^k$ . Next, we enrich this convolution by a series of nested convolutions that are explained below. Let  $k = 2^\ell$  for some  $\ell$ . Normally, the value of  $\theta \otimes a$  is obtained for some  $\theta$ , which needs  $2^\ell$  parameter values. In what follows, we show how to stack convolutions together and use  $2\ell$  parameter values instead. Specifically, for  $\gamma^{(1)}, \dots, \gamma^{(\ell)} \in \mathbb{R}^2$ , we obtain convolutions in a serial manner. Let  $a^{(0)} = a$ , and obtain  $a_j^{(i)}$  for  $i = 1, \dots, \ell$  and  $j = 2^{i-1} + 1, \dots, 2^i$  by

$$a_j^{(i)} = \gamma_1^{(i)} a_j^{(i-1)} + \gamma_2^{(i)} a_{j-2^{i-1}}^{(i-1)}. \quad (2)$$

Since the interval between indices of  $a$  increases exponentially with  $i$ , they are *exponentially dilated* convolutions. This is graphically shown in Figure 2. It follows that

$$a_j^{(\ell)} = \sum_{i=1}^{2^\ell} \hat{\theta}_i a_{j-i+1}^{(0)} = \hat{\theta} \otimes a,$$

where

$$\hat{\theta}_i = \prod_{k=1}^{\ell} \gamma_{f_{i,k}+1}^{(k)},$$

and  $f_{i,k}$  is the  $k$ th digit from the right in the binary representation of  $i$ . Note that  $a_j^{(\ell)}$  is the convolution of  $\hat{\theta}$

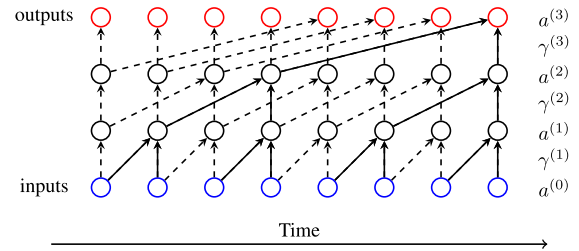


FIGURE 2. Causal and exponentially dilated convolutions. Note that dependencies form a binary tree and the last element of  $a^{(3)}$  is the convolution of all 8 elements of  $a^{(0)}$ .

and  $a$ , obtained by using all elements of  $a$  and the products of  $\gamma^{(1)}, \dots, \gamma^{(\ell)}$ . This means that the convolution is obtained by using  $2\ell$  parameter values instead of  $2^\ell$  parameter values. To put things in perspective, we made two assumptions in (1) and (2) that reduce the required data from  $N^{2^\ell}$  to  $2^\ell$  and then from  $2^\ell$  to  $2\ell$ , respectively.

The set up in (2) ensures that convolutions are *causal*, i.e., they only look at past values. This allows us to slide the stacked convolutions over the input stream to obtain a causal output stream, as shown by dotted arrows in Figure 2. This scheme is also used in WaveNet [27], where residual and skip connections are added to a stack of causal and exponentially dilated convolutions, where convolutions are interleaved by an activation function mimicking the switching mechanism in gated recurrent units [37] to improve accuracy.

### B. PREDICTING ENDOGENOUS VARIABLES

While the above scheme works for an exogenous variable such as the arrival rate, it lacks cross-learning that is needed for endogenous variables. For instance, the treatment rate depends not only on its own history, but also on that of the arrival rate. If there is an uptick or a slowdown in arrivals, a similar pattern will be observed in treatment rates with a delay. Thus, we need a model that can learn from multiple time-series variables to predict future values of an endogenous variable.

We adapt our scheme to accommodate for endogenous variables that depend not only on their own history, but on other time-series variables as well. In doing so, we predict the value of endogenous variable  $b$  at time  $t + 1$ , denoted by  $b_{t+1}$ , from the previous  $k$  observations of  $m$  input variables, where one of these variables is the past observations of  $b$  itself. Collectively, we denote these inputs as  $a$ , where  $a_{i,j}$  for  $i = 1, \dots, m$  and  $j = t - k + 1, \dots, t$ . Note that the input size is  $m \times k$ . One can estimate the conditional probability  $p(b_{t+1}|a_{1,t-k+1}, \dots, a_{m,t})$  by sliding a window of size  $k$  over the  $m$  time-series variables. Let  $N = \max_{i,j}(a_{i,j})$  be an upper limit to  $a$ . We need to collect probabilities for  $m \times k$  combinations of  $N$  variables, resulting in  $N^{mk}$  probability values. As this is infeasible, we model  $b_{t+1}$  as a linear combination of past values, i.e.,

$$b_{t+1} = \sum_{i=1}^m \sum_{j=1}^k \theta_{i,j} a_{j+t-k}, \quad (3)$$

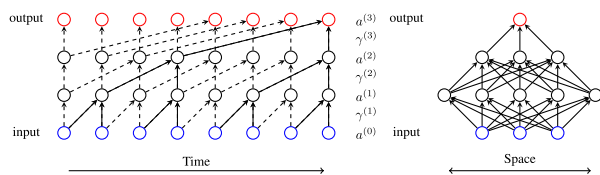


where  $\theta \in \mathbb{R}^{mk}$  is to be learned. This can be represented as applying a 2-dimensional convolution filter in a neural network with no padding on the input data, meaning that (3) can be represented by  $\theta \circledast a$ .

We aim to learn long-term dependencies of time-series variables, which requires  $k$  to be large. Since the size of  $\theta$  scales with  $k$ , learning  $\theta$  soon becomes infeasible. To mitigate this, we propose a systemic representation of  $\theta$  that requires the number of parameters to be in the order of  $\log k$ . To do so, we use a stack of 2-dimensional convolutions, spanning across multiple time-series variables (spatial dimension) over a fixed time window (temporal dimension). These convolutions are causal and exponentially dilated in the temporal dimension to create a receptive window large enough to utilize long-term history and maintain causality between the input and the output. We call such convolutions flow convolutions (FCs). In what follows, we describe how FCs can be stacked to represent  $\theta$ . Let  $k = 2^\ell$  and  $\gamma^{(i)}$  be a set of  $m_i$  2D convolutional filters of size  $m_{i-1} \times 2$  for some  $m_i$ , where  $i = 1, \dots, \ell$ ,  $m_0 = m$  and  $m_\ell = 1$ , where  $\gamma^{(i)} \in \mathbb{R}^{m_i \times m_{i-1} \times 2}$ . Let  $a^{(0)} = (a_{t-k+1}, \dots, a_t)$ , where  $a_i$  is the  $i$ th column of  $a$ . We obtain  $a^{(i)}$  for  $i = 1, \dots, \ell$  by

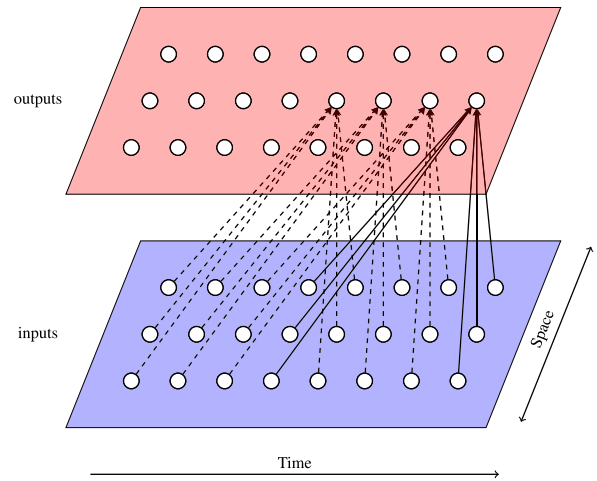
$$a_j^{(i)} = \gamma_1^{(i)} a_j^{(i-1)} + \gamma_2^{(i)} a_{j-2^{i-1}}^{(i-1)}. \quad (4)$$

Note that  $\gamma_1^{(i)}, \gamma_2^{(i)} \in \mathbb{R}^{m_i \times m_{i-1}}$ , meaning that  $a^{(i)} \in \mathbb{R}^{m_i \times k}$  with  $a^{(\ell)} = b$ . The indexing of elements in (4) ensures that the elements of  $a^{(i)}$  are derived from past values in the lower layer, hence causality is maintained. The exponentially increasing dilation ensures that the temporal receptive window increases as we go higher in the stack, while the spatial length is  $m_i$ . This stacking of flow convolutions is shown in Figure 3. Thus, the output of the stack ( $a_{2^\ell}^{(\ell)}$ ) is the result of convolutions of the past  $2^\ell$  observations of  $a^{(0)}$ . Note that by setting  $a_{2^\ell}^{(\ell)} = b_{t+1}$ , we have replicated (3) by using a stack of  $\gamma^{(i)}$ s. Since the size of  $\gamma^{(i)}$  is  $2m_{i-1}m_i$ , the number of parameters needed for this scheme is  $2 \sum_{i=1}^{\ell} m_{i-1}m_i$ , which is in order of  $\ell = \log k$ . Thus, by expressing future values as a linear combination of past values and utilizing a stack of flow convolutions, we reduce the number of required parameters from an order of  $N^k$ , to an order of  $\log k$ .



**FIGURE 3.** Flow convolutions in the temporal (left) and spatial (right) dimensions. In the temporal view, a binary and causal spanning tree is formed; and in the spatial view, a fully connected neural network is formed.

This structure ensures that the output is learned from multiple time-series variables on an exponentially large temporal window. Besides, this structure depends on  $\gamma^{(i)}$  that represent a set of flow convolutions. This is illustrated in Figure 4. Learning in the temporal dimension is as it was in



**FIGURE 4.** Depiction of flow convolutions, in both temporal and spatial dimensions. The output layer (top) is obtained by applying a 2D filter that is causal and exponentially dilated in the temporal dimension on the input. For simplicity, we only show how the middle element in the output layer is obtained.

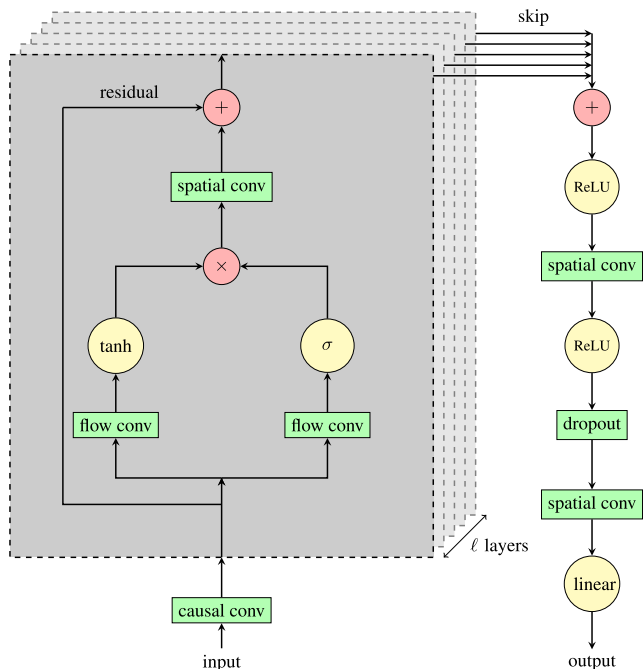
Section IV-A for a single time-series variable, but now we have concurrent learning in the spatial dimension across multiple time-series variables. By using a stack of such convolutions, we temporally use an exponentially large receptive field. We can change the size of spatial dimension as we go higher in the stack by changing the value of  $m_i$ .

### C. PatientFlowNet ARCHITECTURE AND IMPLEMENTATION

We use the framework that we have developed so far, and inspire from [27] to build our convolutional neural network, namely PatientFlowNet. The core idea behind PatientFlowNet is using a stack of flow convolutions. The input is passed to the first layer of the stack, where in each layer, we first apply the flow convolution filter, followed by independent applications of the tanh and sigmoid activation functions and subsequent multiplication of the results, which mimics the activation function in [27]. We discovered that an additional 1-dimensional convolution across time-series variables (i.e., in the spatial dimension) improves the performance. The output of this spatial convolution is passed to the next layer, and with each layer the size of the receptive field is doubled. We continue adding layers until the size of the receptive field matches the temporal input size that we desire. We also add residual and skip connections between and within layers to enable training deeper networks. The output of the topmost layer is the result of a causal convolution of all elements in the receptive field as noted in Section IV-B. We subject this output to a series of additional spatial convolutions and ReLU activation. Finally, we apply a dropout layer to avoid overfitting, followed by a 1-dimensional spatial convolution and a linear activation function to provide a continuous output. This architecture is shown in Figure 5.

### V. DATASET

The data used in this paper comes from EDs in three teaching hospitals in New York City that did not provide permission to



**FIGURE 5.** PatientFlowNet applies flow convolutions in each layer, followed by a switching activation function. It then applies spatial convolution to the results and passes the output to the higher level. Residual and skip connections are added to train deeper models. The outputs of all layers are combined in the post-processing step as shown on the right-hand side of dashed boxes.

be named, and so are called Hospitals 1, 2, and 3. The datasets have per-patient information, namely triage level, arrival time, treatment time, and discharge time over a roughly 2 year period from 2011 to 2013. We discard invalid patient data, which include those whose discharge time is prior to their treatment time or whose treatment time is prior to their arrival time. We also discard the data of any patient whose time-to-treatment is longer than 24 hours. The above exclusions leads to discarding of 1.23% of data.

For each hospital, we extract the number of arrivals ( $\lambda$ ), treatments ( $\mu$ ), and discharges ( $\delta$ ) for different triage levels over fixed consecutive intervals of  $\tau = 60$  minutes. Let  $\lambda_{i,k}$  be the number of patients in triage level  $i$  who arrive during the  $k$ th interval. Similarly, let  $\mu_{i,k}$  and  $\delta_{i,k}$  be the number of patients in triage level  $i$  who are treated and discharged during the  $k$ th interval, respectively. Thus, when there are  $T$  intervals, we have  $\lambda, \mu, \delta \in \mathbb{R}^{5 \times T}$ . The data has a long tail distribution as shown in Figure 6a. Hence, we apply a “log 1p” transformation (defined by  $\log 1p(x) \equiv \log(1 + x)$ ) to shrink the long tail.

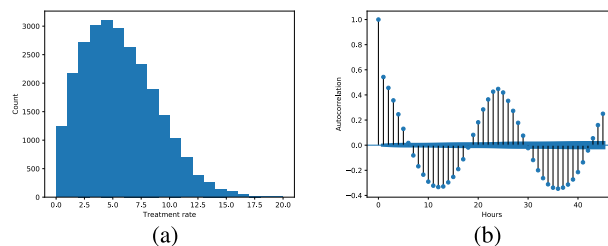
**VI. RESULTS**

In this section, we present experiments on the three datasets described in Section V. For each dataset, we report the prediction error on arrival ( $\lambda$ ), treatment ( $\mu$ ), and discharge ( $\delta$ ) rates. In each experiment, we predict the one-step-ahead values of each of the above rates for all triage levels, given the values of the last  $k$  observations of all rates and triage levels for a given value of  $k$ . That is, the input data consists of the past  $k$

observations of  $\lambda, \mu$ , and  $\delta$  for triage levels  $1, \dots, 5$ , and the output is the next observation of the variable of interest for triage levels  $1, \dots, 5$ . Since some of the baseline methods cannot effectively utilize a long input size, we conduct two sets of experiments:

- 1) Short-term experiments, whose input size  $k$  is short.
- 2) Long-term experiments, whose input size  $k$  is long.

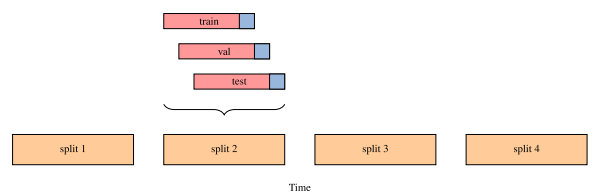
Since the data exhibits a strong 24-hour cyclic behavior as in Figure 6b, we set the input window size to  $k = 24$  for short-term experiments. For long-term experiments, we consider window size of  $k = 2^{11}$ , which roughly includes 3 months of observations.



**FIGURE 6.** (a) Histogram of treatment rate having a long tail, and (b) autocorrelation of treatment rate showing a 24-hour cyclic behavior in Hospital 2.

**A. EXPERIMENT SETUP**

For each hospital, we segment the corresponding dataset (consisting of time-series values of  $\lambda, \mu$ , and  $\delta$  as described in Section V) into 4 non-overlapping partitions, where each partition consists of a train/validation/test split as shown in Figure 7. There are 720 labels (equal to 30 days) in each of the training, validation, and test sets and these labels do not overlap. We use a walk-forward approach [38], where the value of each label is predicted by using  $k$  previous observations.



**FIGURE 7.** The 4-fold training/validation/test walk-forward split used in benchmarking. The labels (in blue) are predicted using previous observations (in red). Each blue tile corresponds to 1 month of labels and each red tile corresponds to 3 months of observations.

Since patient-flow patterns are dependent on external and internal factors in the ED, we expect short-term correlation between patient-flow patterns. Factors such as ambulance routing or patient registration processes may change over time, which in turn may significantly change patient-flow patterns over longer periods, which makes patient-flow rates non-stationary. When the training set and the test set are far apart, their distribution becomes inherently different and a model trained on the training set would not be a good predictor on the test set. To avoid such instances, the train/test/validation splits need to be temporally close

to each other. Hence, the train/validation/test boundaries are chosen in such a way that their corresponding labels (marked in blue in Figure 7) are adjacent to each other and do not overlap. The input data in different sets (marked in red) may overlap, but the blue labels of the test set, in which the prediction errors are reported, does not overlap with the other data. This is in fact standard evaluation practice [38].

For each segment, we train the models on the training set, monitor the errors on the validation set, and report the error over the test set for the model that has the lowest validation error. We use several error metrics to compare the models in our experiments, namely the mean absolute error (MAE), the mean absolute percentage error (MAPE), root mean square error (RMSE) and the coefficient of determination ( $R^2$ ). We use the Adam optimizer [39], and train each model for 4000 epochs using early stopping with tolerance of 100 epochs over loss on the validation set. We repeat each experiment 10 times with random initializations and report the mean error over the test set which offsets the influence of random events.

## B. EXPERIMENTS

We compare the performance of our model against the below-mentioned state-of-the-art baselines in patient-flow prediction:

- Gaussian Process Regression (GPR) consisting of exp-sin-squared and white kernels. The exp-sin-squared kernel has a length scale size of 1 and periodicity of 24 and the white kernel has a noise level of 1.
- Random Forest (RF) [25] with a maximum depth of 4 and 500 estimators.
- Seasonal ARIMA [20] with autoregressive order of 1, differencing order of 1, moving average of 2, and seasonality period of 24.
- Lasso Linear Regression (Lasso-LR) [24] which is linear regression with  $\ell_1$  regularization (Lasso) to avoid overfitting. We used path length of  $10^{-5}$  and tolerance of  $10^{-4}$ .
- LSTM [24] with a sequence-to-sequence architecture and 2 LSTM layers of 32 units.
- Feed forward network (FF) [24] with one hidden layer of 64 units and ReLU activation function.
- WaveNet-Short (WN-S) [27] with 4 layers and filter sizes of 3, 2, 2, and 2 to match input size of  $k = 24$ .
- WaveNet-Long (WN-L) [27] with 11 layers and filter size of 2 in each layer to match input size of  $k = 2^{11}$ .
- PatientFlowNet-Short (PFN-S) as in Section IV-C with input size  $k = 24$ , 4 layers, and 16 filters with temporal length of 3, 2, 2, and 2.
- PatientFlowNet-Long (PFN-L) as in Section IV-C with input size  $k = 2^{11}$ , 11 layers, and 16 filters with temporal length of 2.

Parameter values for GPR, PFN-S, PFN-L were chosen based on cross-validation on the portion of datasets from all three hospitals that were not used for the rest of the experiments. Parameter values for the rest of the baselines

are exactly the same as those in their respective cited papers. We used the TensorFlow and Scikit-learn Python packages for our experiments.

## C. DISCUSSION

In what follows, we present observations based on MAE values in Table 2 for the test sets. Experiments are repeated for other loss metrics such as the mean absolute percentage error (MAPE) and the root mean squared error (RMSE), and the results are included in Table 2. RMSE values are of the same order of magnitude as MAE values, indicating that our log 1p transformation of the values was effective in reshaping the long-tail distribution. The MAPE values are sensitive to small values of labels, and we observe inflated MAPE values due to patient-flow rates being very small for high-acuity triage levels. We observe similar performance gaps between models regardless of the choice of the error metric, Hence, we focus on MAE values.

In short-term experiments, GPR, Lasso-LR, WN-S, and PFN-S are the most accurate in predicting the arrival rate ( $\lambda$ ). Models in long-term experiments such as LSTM, WN-L, and PFN-L outperform those in short-term experiments, as they take as input a longer history to predict future values. Note that with respect to  $\lambda$ , there is not a large gap in prediction accuracy between PFN-L and WN-L despite the fact that PFN-L learns from multiple time-series inputs. This is due to the fact that  $\lambda$  is exogenous and its dependence on other time-series variables is minimal. Thus, cross-learning from other time-series variables is not significant while temporal learning from a longer history provides better predictions for  $\lambda$ .

We observe the reverse when it comes to treatment ( $\mu$ ) and discharge ( $\delta$ ) rates, which are endogenous. As endogenous variables are dependent on other variables, we observe that as expected, cross-learning is significant. Models in short-term experiments that learn from multiple time-series variables, even with a shorter receptive field, generally outperform those in long-term experiments that learn from a single variable, such as LSTM and WN-L. This indicates that dependencies between endogenous variables are mostly short term, and a 24 hour observation can capture most of cross-learnings. Note that PFN-L outperforms the rest of the models in short-term experiments by learning from multiple time-series variables using a long temporal receptive field.

We observe that PatientFlowNet produces more accurate predictions than the rest of the baselines when using a short input window, and it can further improve its accuracy when the length of the input window is increased. PFN-S has a higher prediction accuracy than the rest of the models in short-term experiments for all rates, indicating that it is able to better learn dependencies between different flow variables when using a short input window. Furthermore, WN-L and PFN-L outperform their short-term versions respectively. Therefore, while there is a strong 24-hour cyclic behavior in the data, these models use their large receptive field and learn patterns beyond the 24-hour cycle to make more accurate predictions. This serves as an ablation study of our model,

**TABLE 2.** Comparison of models used in short-term experiments (Gaussian process regression (GPR), Lasso linear regression (Lasso-LR), random forest (RF), feed forward networks (FF), WaveNet-Short (WN-S), and PatientFlowNet-Short (PFN-S)) and long-term experiments (ARIMA (AR), LSTM, WaveNet-Long (WN-L) and PatientFlowNet-Long (PFN-L)) in terms of MAE, MAPE, RMSE, and  $R^2$  (corresponding to models that minimize MAE) for the arrival ( $\lambda$ ), treatment ( $\mu$ ), and discharge ( $\delta$ ) rates in hospitals H1, H2, and H3.

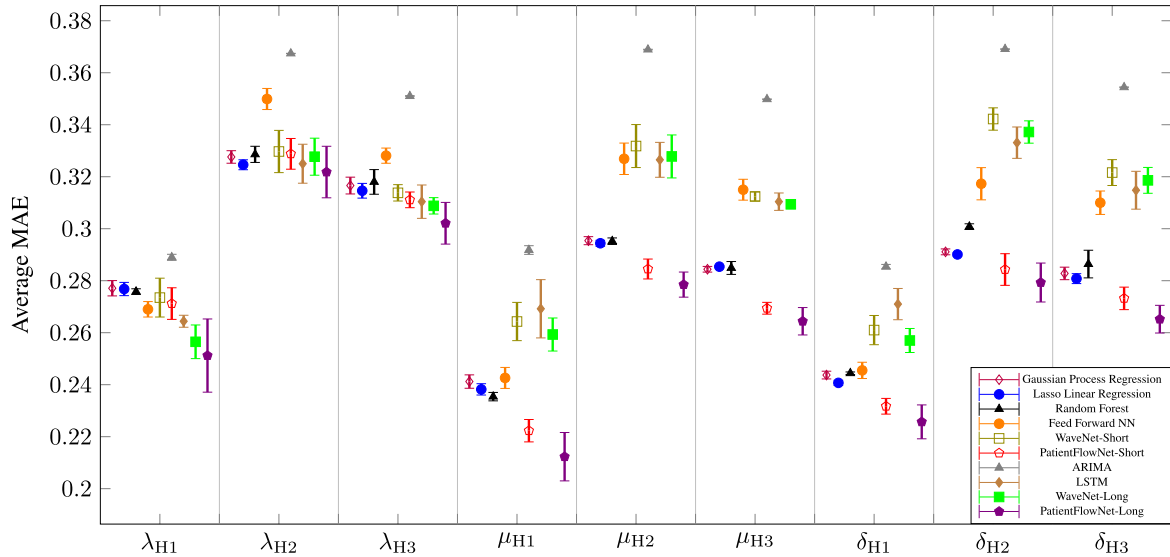
Metric	Model	$\lambda$			$\mu$			$\delta$		
		H1	H2	H3	H1	H2	H3	H1	H2	H3
MAE	GPR	0.277	0.327	0.316	0.241	0.295	0.284	0.243	0.291	0.282
	Lasso-LR	0.276	0.324	0.314	0.238	0.294	0.285	0.240	0.290	0.280
	RF	0.275	0.328	0.318	0.235	0.295	0.284	0.244	0.300	0.286
	FF	0.271	0.343	0.325	0.245	0.329	0.312	0.247	0.314	0.311
	WN-S	0.274	0.329	0.314	0.267	0.330	0.314	0.267	0.337	0.319
	PFN-S	0.269	0.326	0.310	0.221	0.285	0.270	0.231	0.286	0.275
	AR	0.289	0.367	0.350	0.291	0.369	0.349	0.285	0.369	0.354
	LSTM	0.265	0.324	0.311	0.267	0.327	0.311	0.269	0.334	0.316
	WN-L	0.256	0.326	0.309	0.260	0.325	0.309	0.258	0.336	0.318
	PFN-L	<b>0.251</b>	<b>0.320</b>	<b>0.303</b>	<b>0.213</b>	<b>0.277</b>	<b>0.264</b>	<b>0.227</b>	<b>0.280</b>	<b>0.263</b>
MAPE	GPR	60.2	67.4	69.8	53.2	61.5	62.3	52.0	63.0	61.1
	Lasso-LR	61.1	67.4	68.9	52.4	61.1	62.1	59.6	64.6	63.4
	RF	60.6	68.3	71.2	51.5	61.0	62.7	54.3	65.4	63.4
	FF	58.5	72.1	71.7	53.6	67.6	68.5	54.7	70.7	67.9
	WN-S	60.7	68.3	69.3	58.2	70.5	69.1	56.6	75.1	70.0
	PFN-S	59.8	69.1	66.7	47.5	58.9	59.5	50.6	62.5	59.6
	AR	64.3	78.0	78.0	63.2	75.1	75.9	62.7	81.5	77.2
	LSTM	57.9	66.0	69.4	59.7	67.7	68.9	59.9	73.0	70.8
	WN-L	56.9	69.0	67.4	58.0	68.1	69.3	57.2	74.2	69.7
	PFN-L	<b>55.6</b>	<b>66.8</b>	<b>66.2</b>	<b>47.9</b>	<b>58.7</b>	<b>57.7</b>	<b>49.5</b>	<b>61.8</b>	<b>59.6</b>
RMSE	GPR	0.420	0.494	0.482	0.368	0.463	0.439	0.361	0.445	0.422
	Lasso-LR	0.424	0.492	0.481	0.366	0.449	0.434	0.367	0.442	0.429
	RF	0.423	0.501	0.489	0.360	0.451	0.437	0.377	0.458	0.434
	FF	0.408	0.535	0.505	0.372	0.490	0.480	0.376	0.490	0.475
	WN-S	0.411	0.505	0.470	0.402	0.512	0.477	0.402	0.522	0.489
	PFN-S	0.416	0.496	0.475	0.339	0.435	0.411	0.355	0.438	0.412
	AR	0.442	0.553	0.538	0.443	0.567	0.532	0.436	0.570	0.545
	LSTM	0.408	0.500	0.472	0.407	0.505	0.472	0.410	0.511	0.476
	WN-L	0.392	0.494	0.475	0.393	0.499	0.472	0.392	0.514	0.490
	PFN-L	<b>0.386</b>	<b>0.488</b>	<b>0.458</b>	<b>0.325</b>	<b>0.428</b>	<b>0.406</b>	<b>0.341</b>	<b>0.428</b>	<b>0.407</b>
$R^2$	GPR	0.847	0.830	0.805	0.889	0.866	0.835	0.880	0.865	0.848
	Lasso-LR	0.850	0.830	0.802	0.886	0.862	0.840	0.881	0.862	0.842
	RF	0.849	0.826	0.797	0.890	0.860	0.840	0.881	0.855	0.839
	FF	0.856	0.804	0.789	0.883	0.828	0.803	0.881	0.839	0.807
	WN-S	0.848	0.828	0.804	0.859	0.823	0.809	0.864	0.813	0.795
	PFN-S	0.853	0.821	0.811	0.902	0.873	0.856	0.894	0.870	0.855
	AR	0.835	0.787	0.750	0.830	0.779	0.757	0.836	0.777	0.746
	LSTM	0.860	0.828	0.806	0.853	0.829	0.809	0.850	0.819	0.806
	WN-L	0.868	0.828	0.810	0.863	0.827	0.807	0.866	0.812	0.794
	PFN-L	<b>0.870</b>	<b>0.836</b>	<b>0.816</b>	<b>0.906</b>	<b>0.877</b>	<b>0.862</b>	<b>0.897</b>	<b>0.872</b>	<b>0.859</b>

indicating that its superior prediction accuracy derives from both its cross-learning as well as its large temporal window. When averaged over all rates in all hospitals, PFN-L has an MAE that is 4.8% lower than the leading baseline.

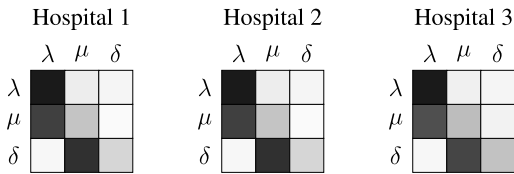
The MAE values along with standard errors are shown in Figure 8. Note that deep learning models generally have higher variations than classical machine learning methods such as regression and random forest as their initialization is more stochastic. For the exogenous variable, i.e.,  $\lambda$ , while deep learning models generally outperform classical machine learning models, the performance is within the margin of error. For the endogenous variables, i.e.,  $\mu$  and  $\delta$ , while the error-bars for deep learning models is still large, the performance gap between PFN-L and the remaining models is significant. Thus, by efficiently learning from multiple flow variables over a long temporal window, PFN-L has a slightly better prediction accuracy for exogenous variables, while having a significantly better accuracy in predicting endogenous variables in the ED.

Since PatientFlowNet can simultaneously learn from multiple time-series variables, we can extract dependencies between patient-flow variables in retrospect. To do so, we examine flow convolution filters in PatientFlowNet. Figure 9 shows normalized values of flow convolution filters in the first layer of PatientFlowNet-Long for each hospital. The values are normalized in such a way that the sum of values in each filter is 1. Thus, darker shades denote stronger dependency. Note that  $\lambda$  is best estimated using its own history,  $\mu$  is best estimated using past values of  $\lambda$  and  $\mu$ , and  $\delta$  is highly dependent on past values of  $\mu$  and  $\delta$ . This validates our earlier assumption on the exogenous nature of  $\lambda$ . Lack of dependency of  $\mu$  on  $\delta$  indicates that slowdowns and speed ups in discharging patients do not significantly affect treatment rates. Besides, lack of dependency of  $\delta$  on  $\lambda$  indicates that patients are not necessarily rushed out when arrival rates spike. This analysis is based on historical data in our data sets that were taken during non-distress times; and can change during distress times.





**FIGURE 8.** Average test error and its confidence interval for predicted rates for different models in 3 hospitals. The length of the error bar for each variable is  $2 \times$  the standard error of test error across 4 folds. Note that for endogenous variables ( $\mu$  and  $\delta$ ), PatientFlowNet-Long outperforms other models by a large margin. Its average MAE for exogenous variables ( $\lambda$ ) is also less than other models, but within the margin of error.



**FIGURE 9.** Normalized values of the first layer flow convolutional filters for all hospitals. The level of darkness in row ( $i, j$ ) indicates the dependence of predicted variable  $i$  on input variable  $j$ . Darker shades indicate larger filter values and higher dependency.

**TABLE 3.** Comparison of MAE values for predicting variables in rows using variables in columns when using PFN-L for the three hospitals in our study (H1, H2, and H3).

	$\lambda$	$\mu$	$\delta$	$(\lambda, \mu)$	$(\lambda, \delta)$	$(\mu, \delta)$	$(\lambda, \mu, \delta)$
H1	$\lambda$	<b>0.250</b>	0.267	0.284	0.251	0.253	0.271
	$\mu$	0.227	0.271	0.279	0.215	0.231	0.273
	$\delta$	0.238	0.234	0.262	0.235	0.244	0.230
H2	$\lambda$	<b>0.319</b>	0.335	0.347	0.320	0.321	0.341
	$\mu$	0.285	0.323	0.341	0.283	0.294	0.331
	$\delta$	0.301	0.285	0.329	0.283	0.287	0.281
H3	$\lambda$	<b>0.299</b>	0.321	0.327	0.301	0.303	0.323
	$\mu$	0.277	0.271	0.279	0.265	0.281	0.273
	$\delta$	0.275	0.270	0.284	0.273	0.270	0.265

We also perform an ablation study on the set of input parameters to our model and to assess their importance in predicting future values. Table 3 shows MAE values of PFN-L for predicting patient-flow parameters using different input configurations in the three hospitals in our study. We observe the same dependency pattern discussed above, where the exogenous variable ( $\lambda$ ) is best predicted using its own history while the endogenous variables ( $\mu$  and  $\delta$ ) are best predicted by using the history of all variables. While the use of additional input parameters is expected to add uncertainty to the model and reduce its prediction accuracy, we note that for the exogenous variable such reduction is minimal as PFN-L learns and adapts to such dependencies, as observed in Figure 9.

### VII. CONCLUSION

In this paper, we presented a convolutional neural network model, called PatientFlowNet, to forecast patient flow in emergency departments. The design of PatientFlowNet enables it to learn simultaneously from multiple flow variables over an exponentially large input window, while keeping the model size manageable. We have shown that our PatientFlowNet achieves better prediction accuracy than the current state-of-the-art models used for patient-flow forecasting. While PatientFlowNet has a slightly better prediction accuracy for exogenous variables such as patient arrival rates, it produces substantially more accurate predictions for the endogenous flow variables such as treatment and discharge rates. The short-term predictions by PatientFlowNet can be used to estimate workflow variables such as wait times. We also described how dependencies between flow variables in the emergency department can be deduced, in a data-driven fashion, by inspecting the learned parameters in the first layer filters of PatientFlowNet.

### REFERENCES

- [1] S. Di Somma, L. Paladino, L. Vaughan, I. Lalle, L. Magrini, and M. Magnanti, "Overcrowding in emergency department: An international issue," *Internal Emergency Med.*, vol. 10, no. 2, pp. 171–175, Mar. 2015.
- [2] S. Verelst, P. Wouters, J.-B. Gillet, and G. Van den Berghe, "Emergency department crowding in relation to in-hospital adverse medical events: A large prospective observational cohort study," *J. Emergency Med.*, vol. 49, no. 6, pp. 949–961, Dec. 2015.
- [3] C. Morley, M. Unwin, G. M. Peterson, J. Stankovich, and L. Kinsman, "Emergency department crowding: A systematic review of causes, consequences and solutions," *PLoS ONE*, vol. 13, no. 8, Aug. 2018, Art. no. e0203316.
- [4] I. Satia, R. Cusack, J. M. Greene, P. M. O'Byrne, K. J. Killian, and N. Johnston, "Prevalence and contribution of respiratory viruses in the community to rates of emergency department visits and hospitalizations with respiratory tract infections, chronic obstructive pulmonary disease and asthma," *PLoS ONE*, vol. 15, no. 2, Feb. 2020, Art. no. e0228544.

- [5] J. E. Wong, Y. S. Leo, and C. C. Tan, "COVID-19 in Singapore—current experience: Critical global issues that require attention and action," *Jama*, vol. 323, pp. 1243–1244, Apr. 2020.
- [6] W.-K. Ming, J. Huang, and C. J. Zhang, "Breaking down of healthcare system: Mathematical modelling for controlling the novel Coronavirus (2019-NCoV) outbreak in Wuhan, China," *bioRxiv*, 2020.
- [7] A. Remuzzi and G. Remuzzi, "COVID-19 and Italy: What next?" *Lancet*, vol. 395, no. 10231, pp. 1225–1228, Apr. 2020.
- [8] L. Pan, L. Wang, and X. Huang, "How to face the novel coronavirus infection during the 2019–2020 epidemic: The experience of sichuan provincial People's hospital," *Intensive Care Med.*, vol. 46, no. 4, pp. 573–575, Apr. 2020.
- [9] D. Coen, C. Paolillo, M. Cavazza, G. Cervellini, A. Bellone, S. Perlini, and I. Casagrande, "Changing emergency department and hospital organization in response to a changing epidemic," *Emergency Care J.*, vol. 16, no. 1, Mar. 2020.
- [10] J. L. Wiler, R. T. Griffey, and T. Olsen, "Review of modeling approaches for emergency department patient flow and crowding research," *Academic Emergency Med.*, vol. 18, no. 12, pp. 1371–1379, Dec. 2011.
- [11] I. S. Cheng, "Emergency department crowding and hospital patient flow: Influential factors and evidence-informed solutions," Ph.D. dissertation, Dept Clin. Sci. Educ., Södersjukhuset, Stockholm, Sweden, 2016.
- [12] P. McKenna, S. M. Heslin, P. Viccellio, W. K. Mallon, C. Hernandez, and E. J. Morley, "Emergency department and hospital crowding: Causes, consequences, and cures," *Clin. Experim. Emergency Med.*, vol. 6, no. 3, p. 189, 2019.
- [13] H. Chan, S. Lo, L. Lee, W. Lo, W. Yu, Y. Wu, S. Ho, R. Yeung, and J. Chan, "Lean techniques for the improvement of patients' flow in emergency department," *World J. Emergency Med.*, vol. 5, no. 1, p. 24, 2014.
- [14] C. Oh, A. M. Novotny, P. L. Carter, R. K. Ready, D. D. Campbell, and M. C. Leckie, "Use of a simulation-based decision support tool to improve emergency department throughput," *Oper. Res. for Health Care*, vol. 9, pp. 29–39, Jun. 2016.
- [15] E. Ang, S. Kwasnick, M. Bayati, E. L. Plambeck, and M. Aratow, "Accurate emergency department wait time prediction," *Manuf. Service Oper. Manage.*, vol. 18, no. 1, pp. 141–156, 2016.
- [16] W. Whitt and X. Zhang, "A data-driven model of an emergency department," *Oper. Res. Health Care*, vol. 12, pp. 1–15, Mar. 2017.
- [17] J. G. Dai and P. Shi, "A two-time-scale approach to time-varying queues in hospital inpatient flow management," *Oper. Res.*, vol. 65, no. 2, pp. 514–536, Apr. 2017.
- [18] R. Calegari, F. S. Fogliatto, F. R. Lucini, J. Neyeloff, R. S. Kuchenbecker, and B. D. Schaan, "Forecasting daily volume and acuity of patients in the emergency department," *Comput. Math. Methods Med.*, Sep. 2016.
- [19] L. Luo, L. Luo, X. Zhang, and X. He, "Hospital daily outpatient visits forecasting using a combinatorial model based on ARIMA and SES models," *BMC Health Services Res.*, vol. 17, no. 1, p. 469, Dec. 2017.
- [20] A. Harper and N. Mustafee, "A hybrid modelling approach using forecasting and real-time simulation to prevent emergency department overcrowding," in *Proc. Winter Simulation Conf. (WSC)*, Dec. 2019, pp. 1208–1219.
- [21] M. Carvalho-Silva, M. T. T. Monteiro, F. D. Sá-Soares, and S. Dória-Nóbrega, "Assessment of forecasting models for patients arrival at emergency department," *Oper. Res. Health Care*, vol. 18, pp. 112–118, Sep. 2018.
- [22] J. S. Peck, J. C. Bennenyan, D. J. Nightingale, and S. A. Gaehde, "Predicting emergency department inpatient admissions to improve same-day patient flow," *Academic Emergency Med.*, vol. 19, no. 9, pp. E1045–E1054, Sep. 2012.
- [23] W. M. Baihaqi, M. Dianingrum, K. A. N. Ramadhan, and T. Hariguna, "Linear regression method to model and forecast the number of patient visits in the hospital," in *Proc. 3rd Int. Conf. Inf. Technol., Inf. Syst. Electr. Eng. (ICITISEE)*, Nov. 2018, pp. 247–252.
- [24] G. Guan and B. E. Engelhardt, "Predicting sick patient volume in a pediatric outpatient setting using time series analysis," in *Proc. Mach. Learn. Healthcare Conf.*, 2019, pp. 271–287.
- [25] L. Luo, X. Xu, J. Li, and W. Shen, "Short-term forecasting of hospital discharge volume based on time series analysis," in *Proc. IEEE 19th Int. Conf. e-Health Netw., Appl. Services (Healthcom)*, Oct. 2017, pp. 1–6.
- [26] F. Kadri, M. Baraoui, and I. Nouaouri, "An LSTM-based deep learning approach with application to predicting hospital emergency department admissions," in *Proc. Int. Conf. Ind. Eng. Syst. Manage. (IESM)*, Sep. 2019, pp. 1–6.
- [27] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," 2016, *arXiv:1609.03499*. [Online]. Available: <http://arxiv.org/abs/1609.03499>
- [28] S. Bai, J. Zico Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, *arXiv:1803.01271*. [Online]. Available: <http://arxiv.org/abs/1803.01271>
- [29] A. Borovykh, S. Bohte, and C. W. Oosterlee, "Conditional time series forecasting with convolutional neural networks," 2017, *arXiv:1703.04691*. [Online]. Available: <http://arxiv.org/abs/1703.04691>
- [30] K. Li, C. Liu, T. Zhu, P. Herrero, and P. Georgiou, "GluNet: A deep learning framework for accurate glucose forecasting," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 2, pp. 414–423, Feb. 2020.
- [31] O. Yazdanbakhsh and S. Dick, "Multivariate time series classification using dilated convolutional neural network," 2019, *arXiv:1905.01697*. [Online]. Available: <http://arxiv.org/abs/1905.01697>
- [32] A. Choudhury and E. Urena, "Forecasting hourly emergency department arrival using time series analysis," *Brit. J. Healthcare Manage.*, vol. 26, no. 1, pp. 34–43, Jan. 2020.
- [33] W. Whitt and X. Zhang, "Forecasting arrivals and occupancy levels in an emergency department," *Operations Res. Health Care*, vol. 21, pp. 1–18, Jun. 2019.
- [34] Q. Xu, K.-L. Tsui, W. Jiang, and H. Guo, "A hybrid approach for forecasting patient visits in emergency department," *Qual. Rel. Eng. Int.*, vol. 32, no. 8, pp. 2751–2759, Dec. 2016.
- [35] M. Yucecan, M. Gul, and E. Celik, "A multi-method patient arrival forecasting outline for hospital emergency departments," *Int. J. Healthcare Manage.*, pp. 1–13, Oct. 2018.
- [36] S. Jiang, K.-S. Chin, and K. L. Tsui, "A universal deep learning approach for modeling the flow of patients under different severities," *Comput. Methods Programs Biomed.*, vol. 154, pp. 191–203, Feb. 2018.
- [37] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*. [Online]. Available: <http://arxiv.org/abs/1406.1078>
- [38] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*. OTexts, 2018.
- [39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: <http://arxiv.org/abs/1412.6980>



**ALI R. SHARAFAT** received the B.Math. degree (Hons.) in combinatorics and optimization and computer science from the University of Waterloo, in 2009, and the M.Sc. degree in electrical engineering from Stanford University, in 2011. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, Stanford University. His research interests include application of deep learning methods in healthcare, hospital workflow optimization, patient outcome prediction, and disease propagation.



**MOHSEN BAYATI** received the B.Sc. degree in mathematics from the Sharif University of Technology and the Ph.D. degree in electrical engineering from Stanford University, in 2007. His dissertation was on algorithms and models for large-scale networks. From 2005 to 2006, he interned at IBM Research and Microsoft Research. From 2007 to 2009, he was a Postdoctoral Researcher with Microsoft Research, working mainly on applications of machine learning and optimization methods in healthcare and online advertising. From 2009 to 2011, he was a Postdoctoral Scholar at Stanford University, with focusing on high-dimensional statistical learning. In 2011, he joined the Graduate School of Business, Stanford University, where he has been an Associate Professor of Operations, Information, and Technology, since 2015. He was awarded the INFORMS Healthcare Applications Society Pierskalla Best Paper Award, in 2014 and 2016, the INFORMS Applied Probability Society Best Paper Award, in 2015, and the National Science Foundation CAREER Award.

• • •