Combining Deep Neural Networks and Classical Time Series Regression Models for Forecasting Patient Flows in Hong Kong

Jiang, Shancheng; Xiao, Ran; Wang, Long; Luo, Xiong; Huang, Chao; Wang, Jenq-Haur; Chin, Kwai-Sang; Nie, Ximing

Published in:
IEEE Access

Published: 01/01/2019

Document Version:
Final Published version, also known as Publisher’s PDF, Publisher’s Final version or Version of Record

License:
CC BY

Publication record in CityU Scholars:
Go to record

Published version (DOI):
10.1109/ACCESS.2019.2936550

Publication details:

Citing this paper
Please note that where the full-text provided on CityU Scholars is the Post-print version (also known as Accepted Author Manuscript, Peer-reviewed or Author Final version), it may differ from the Final Published version. When citing, ensure that you check and use the publisher's definitive version for pagination and other details.

General rights
Copyright for the publications made accessible via the CityU Scholars portal is retained by the author(s) and/or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights. Users may not further distribute the material or use it for any profit-making activity or commercial gain.

Publisher permission
Permission for previously published items are in accordance with publisher's copyright policies sourced from the SHERPA RoMEO database. Links to full text versions (either Published or Post-print) are only available if corresponding publishers allow open access.

Take down policy
Contact lbscholars@cityu.edu.hk if you believe that this document breaches copyright and provide us with details. We will remove access to the work immediately and investigate your claim.

Download date: 25/12/2022
Combining Deep Neural Networks and Classical Time Series Regression Models for Forecasting Patient Flows in Hong Kong

SHANCHENG JIANG\textsuperscript{1,2,3}, RAN XIAO\textsuperscript{4}, LONG WANG\textsuperscript{1,2,5} (Member, IEEE), XIONG LUO\textsuperscript{1,2}, CHAO HUANG\textsuperscript{1,2} (Member, IEEE), JENQ-HAUR WANG\textsuperscript{6}, KWAI-SANG CHIN\textsuperscript{7}, AND XIMING NIE\textsuperscript{8}

\textsuperscript{1}School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China \\
\textsuperscript{2}Guangdong Provincial Key Laboratory of New and Renewable Energy Research and Development, Guangzhou 510640, China \\
\textsuperscript{3}School of Intelligent Systems Engineering, Sun Yat-sen University, Guangzhou 510275, China \\
\textsuperscript{4}College of Materials Science and Engineering, Shenzhen University, Shenzhen 518060, China \\
\textsuperscript{5}Key Laboratory of Fluid and Power Machinery, Ministry of Education, Xihua University, Chengdu 610039, China \\
\textsuperscript{6}Department of Computer Science and Information Engineering, National Taipei University of Technology, Taipei 10608, Taiwan \\
\textsuperscript{7}Department of Systems Engineering and Engineering Management, City University of Hong Kong, Hong Kong \\
\textsuperscript{8}Department of Neurology, Beijing Tiantan Hospital, Capital Medical University, Beijing 100054, China

Corresponding author: Chao Huang (chao.huang@my.cityu.edu.hk)

Shancheng Jiang and Ran Xiao contributed equally to this work and should be considered co-first authors.

This work was supported in part by the Guangdong Provincial Key Laboratory of New and Renewable Research and Development under Grant Y80761001, in part by the University of Science and Technology Beijing through the National Taipei University of Technology Joint Research Program under Grant TW2018008, in part by the Fundamental Research Funds for the Central Universities under Grant 06500103 and Grant 06500078, in part by the Hong Kong RGC under Grant T32-102/14N, in part by National Nature Science and Foundation of China under Grant 71801031, in part by the City University of Hong Kong SRG under Grant 7004698, in part by the National Key Research and Development Program of China under Grant 2018YFC0810601, and in part by the Open Research Subject of Key Laboratory of Fluid and Power Machinery, Xihua University, Ministry of Education, under Grant szjj2019-011.

\textbf{ABSTRACT} Deep Neural Networks (DNNs) has been dominating recent data mining related tasks with better performances. This article proposes a hybrid approach that exploits the unique predictive power of DNN and classical time series regression models, including Generalized Linear Model (GLM), Seasonal AutoRegressive Integrated Moving Average model (SARIMA) and AutoRegressive Integrated Moving Average with eXplanatory variable (ARIMAX) method, in forecasting time series in reality. For each selected time series regression model, three different hybrid strategies are designed in order to merge its results with DNNs, namely, Zhang’s method, Khasei’s method, and moving average filter-based method. The real seasonal time series data of patient arrival volume in a Hong Kong A&ED center was collected for the period July 1, 2009, through June 30, 2011 and is used for comparing the forecast accuracy of proposed hybrid strategies. The mean absolute percentage error (MAPE) is set as the metric and the result indicates that all hybrid models achieved higher accuracy than original single models. Among 3 hybrid strategies, generally, Khasei’s method and moving average filter-based method achieve lower MAPE than Zhang’s method. Furthermore, the predicted value is an important prerequisite of conducting the rostering and scheduling in A&ED center, either in the simulation-based approach or in the mathematical programming approach.

\textbf{INDEX TERMS} Data mining, deep neural networks, hybrid approach, time series regression, data mining.

\section*{I. INTRODUCTION}
Nowadays, Accident and Emergency Departments (AEDs) play a significant role in the urgent assessment and treatment of illness and injury. Especially in Hong Kong, the AED can be regarded as a special healthcare department, which provides high-quality specialized emergency care, with the help of available resources, to patients by a team of well-trained and qualified staff. According to the Hong Kong Hospital Authority, AEDs deal with all kinds of emergencies and adopt a 5-point Triage system to assess and prioritize patients in need of urgent treatment. Under the challenges of
all of the forecast models abstracted from reviewed articles can be labeled as the single classical prediction model. According to incomplete statistics, the ARIMA model [6], [7] is the most widely used forecast model. Among those regression-based models, most approaches choose the Multiple Linear Regression (MLR) model [8], [9] and get acceptable outcomes though the model seems quite simple. Some approaches [10], [11] choose an improved version of Poisson regression with the idea that the patient arrival volume is count number. It is notable that a number of Artificial Neural Network (ANN)-based methods [12]–[14] (Aladag & Aladag, 2012; Menke et al., 2014; Xu, Wong, & Chin, 2013) are applied in this problem because of its flexibility of approximating any linear and nonlinear function to any desired degree of accuracy. However, all these approaches choose traditional shallow ANNs which have only one hidden layer and cannot learn the deep structure of real data sets. Until now, we have not found a proposed approach using hybrid model.

B. FORECAST MODEL COMBINATION

Based on the result of Hibon’s study [15], not only do the best possible combinations perform better than the best possible single model, but also combined forecasts reduces the risk of selecting an individual forecast model in practice. After a retrospective study of proposed hybrid models via literature review, the basic idea of combining forecast models is to utilize each model’s unique feature to capture different patterns in the data. Wedding and Cios [16] combined the result of Radial Basis Function (RBF) neural networks [17] with Box-Jenkins (UBJ) models [18] to predict future values of data. Terui and Van Dijk [19] investigated combinations of forecasts generated by linear and some nonlinear models and indicate that combined forecasts perform well. It is worth to mention that the method with time varying coefficients dominates marginal forecasts for inside sample performance. In the meantime, a huge number of hybrid strategies are proposed to improve the performance of model combination. Zhang [3] proposes a hybrid strategy that combines both ARIMA and ANN models to take advantage of the unique strength of ARIMA and ANN models in linear and nonlinear modeling. Then, a novel hybridization of artificial neural networks and ARIMA model is proposed by Khashei and Bijari [4], which identify and magnify the existing linear structure in data. Babu and Reddy [5] propose a moving-average filter based method, in which the nature of volatility was explored using a moving-average filter, and then an ARIMA and an ANN model were suitably applied. Therefore, in our approach, all these three kinds of strategies are applied to our models and compared by real data sets.

II. RELATED WORKS

A. PATIENT FLOW PREDICTION

After conducting a systematic literature review of articles concerning patient flow prediction, we find that almost
ANNs that have only one hidden layer, DNNs can still deal with complex linear and non-linear relationship. Although the universal approximation theorem states that a feedforward network with at least one hidden layer with any activation function can approximate any function, the theorem does not restrict the size of this network, and exponential number of hidden units might be required with the increasing complexity of objective function. Hence, the shallow ANN may fail to fit and generalize correctly with an infeasibly large number of hidden units whereas the DNN can reduce the number of hidden units and also the amount of generalization error. Moreover, when each sequential hidden layer is connected with the one below it, each hidden layer becomes a projection of the layer below it as well. The information from the layer below is nicely transferred by a non-linear, optimally weighted, projection in each subsequent layer of the DNN. Therefore, the extra hidden layers ensure the potential capacity of modeling complex data structure with fewer units than a traditional shallow network.

In our approach, we adopt two types of DNNs. One is a Multi-Layer Perception (MLP) with a rescaled and shifted logistic activation function:

$$f(\alpha) = (e^{\alpha} - e^{-\alpha})/(e^{\alpha} + e^{-\alpha}).$$

This activation function is symmetric around 0 and allows the training algorithm to converge faster. We implement the DNNs by using an R package called “H2o” in R x64 3.1.2. The optimizer for weights update is a parallelized version of Stochastic Gradient Descent (SGD) [20]. This algorithm is faster and more efficient than the standard SGD as it is implemented in a multi-threaded computer system.

The other one is a fully connected recurrent neural network (RNN) where the output is to be fed back to input. Each node in the fully connected RNN is connected with a one-way connection to every other node in the next layer. Similar with MLP, each node has a time-varying real-valued activation and each connection has a modifiable real-valued weight. The feedback structure of RNN allows it to exhibit temporal dynamic behavior for a time sequence. Unlike MLP, RNN can use its internal state to process sequences of inputs. The activation function is as same as the one of MLP and we introduced a state-of-the-art optimizer Kingma and Ba [21] as the learning algorithm of RNN. Both MLP and RNN contains three hidden layers, which is determined based on the scale of training datasets and some initial comparative tests.

### B. TECHNIQUES TO AVOID OVERFITTING

With more hidden layers, it is obvious that DNNs are often much harder to train than shallow neural networks. DNNs are prone to get over-fitting because of the added layers of abstraction, which allow them to model rare dependencies in the training data. In general, a model containing high-complexity does not necessarily include the target function or the true data generating process. In the modeling process, we almost never have access to the true data generating process, which means that it is usually difficult to adjust the complexity of the model with deep architecture. Not simply to control the size or the number of parameters, in this approach, we aim to fit a large DNN which has been regularized appropriately. In this way, over-fitting can be avoided while the flexibility of the DNN is maintained.

Based on the motivation mentioned above, we apply two kinds of techniques to avoid over-fitting. The first one is $L_1$ and $L_2$ regularization methods, in which we modify the loss function as

$$L(W, B[j]) = L(W, B[j]) + \lambda_1 R_1(W, B[j]) + \lambda_2 R_2(W, B[j]).$$

In this way, those weights which are used to fit the sampling error are cut off and then a smoother model is formulated. $L_2$ regularization represents the sum of squares of all the weights and biases in the network. This parameter norm penalty is commonly known as weight decay and the mechanism is summarized as follows. The input $X$ is perceived to have higher variance by adding the $L_2$ regularization. This makes the learning algorithm shrink the weights on features whose covariance with the output target is low. $L_1$ regularization represents the sum of absolute values of the weights and biases in the DNNs. Compared with $L_2$ regularization, $L_1$ regularization results in a solution that is sparser (i.e. more parameters have an optimal value of zero). Similar to LASSO, the sparsity property induced by $L_1$ regularization has been considered as a feature selection mechanism. The constant $\lambda_1$ and $\lambda_2$ control the degree of penalties.

The second one is a more recent technique named as dropout method [22]. In dropout, some number of units are randomly omitted from the hidden layers during training. In detail, with different training examples, each neuron in the input layer or the hidden layers can suppress its activation with a given probability. Then, an exponentially large number of models are produced because each training example trains a different model. Finally, the weights and biases in these models are averaged as an ensemble. Therefore, the rare dependencies which is possible to occur in the training data are prevented.

### C. THREE COMBINATION STRATEGIES

1) ZHANG’S METHOD

First, we design a naive hybrid strategy which is based on G.P. Zhang’s idea. He assumes that it may be reasonable to consider a given time series to be composed of a linear component and a nonlinear component, which can be described as

$$y_t = L_t + N_t.$$  

In equation (3), $L_t$ denotes the linear component and $N_t$ denotes the nonlinear component. These two components can be estimated from the data as followed. We let $k$ linear models, GLM, SARIMA and ARIMAX, as mentioned above, to model the linear component respectively, and then the residuals from the linear model will contain only the nonlinear component with the assumption that the linear model can...
fit the linear component exactly. Let $e_t$ denote the residual at time $t$ from the linear model, as is written in equation (4), where $\hat{L}_t$ is the forecast value for time $t$ from the fitted linear model.

$$ e_t = y_t - \hat{L}_t. \quad (4) $$

After modeling $e_t$ using DNNs, nonlinear relationships can be discovered. Denoting the forecast value from DNNs as $\hat{N}_t$, the combined forecast will be

$$ \hat{y}_t = \hat{L}_t + \hat{N}_t. \quad (5) $$

2) KHASHEI’S METHOD

This combination strategy can be regarded as a modified version of Zhang’s method, in which the existing linear structure in data is magnified and DNNs is used to determine a model to capture the potential data generating process. Unlike the Zhang’s assumption shown in equation (3), Khashei et al. points out that a time series should be considered as function of a linear and a nonlinear component

$$ y_t = f(L_t, N_t). \quad (6) $$

Similar as Zhang’s method, the main aim of the first stage is linear modeling. Therefore, 3 linear models are used to model the linear component and the residual series $e_t$ is acquired. In second stage, the main aim is nonlinear modeling and the data generating process modeling. The DNNs are utilized to conduct this task and the combined forecast is shown in equation (7),

$$ y_t = f(e_{t-1}, \ldots, e_{t-n}, \hat{L}_t, \hat{L}_{t+1}, \ldots, \hat{L}_{t+l}, z_{t-1}, \ldots, z_{t-n}) \quad (7) $$

where $e_t$ is the residual at time $t$ from the three linear models, and $\hat{L}_t$ is the forecast value for time $t$; $z_t = (1-B)^d(y_t-\mu)$ can be regarded as a linear structure abstracted from time series; $m, l$ and $n$ are integers that are determined in design process of final DNNs.

3) MOVING AVERAGE FILTER-BASED METHOD

We design a moving-average filter based hybrid strategy, which is based on C. Narendra Babu’s study [5]. With the opinion that whether a given sequence is normally distributed or not must be examined via Jarque–Bera normality test, a time series is firstly divided into a low-volatility component and a high-volatility component by moving-average filter. In this process, the order of moving-average filter is set by checking the kurtosis of the sequence. Then, the kurtosis of the sequence. Then, the low-volatility component is fitted by three linear models, respectively, and the high-volatility component is fitted by the DNNs. Finally, the results of two components are combined and the detail is shown in Figure 1.
Administrative data was obtained from a certain A&ED center in Hong Kong from 1st July 2009 to 31st March 2011. According to the “Triage System in Hong Kong A&EDs”, patients are divided into five categories according to their injury severity, and in this study, we concerned patients in Category 3 and 4, and also the total volume. The reason is that the number of patients in Category 1 and 2 is far less than the total volume, and the corresponding patient arrival pattern is usually fixed. In the meantime, the patients in Category 5 do not need the service at once thus the related patient flow does not influence the schedule plan of A&ED staff.
Figure 2 summarizes a series of basic statistical graph of acquired data set.

From Figure 2, we can capture obvious distinct monthly pattern and weekly pattern on daily patient flow. Therefore, we add both “month of a year” and “day of a week” in the initial feature set, as the calendar-based factors. As is mentioned in literature part, we also include the holiday related factors (i.e. whether it is holiday and whether it is before or after holiday). All the calendar-based co-variables are converted into binary dummy variables. For the meteorological factors, we use air temperature, mean dew point, mean relative humidity, mean amount of cloud, total rainfall, mean wind speed as relevant features. We add the change of above index into the initial feature set as the lag effect of these meteorological factors.

To validate the performance of proposed hybrid model, the whole data set is firstly divided into training set and testing set, with the ratio around 7:3. The prediction horizon is fixed to 28 days ahead, which satisfies the real demand from the decision makers in the healthcare system. Considering
the application environment and the sequential instances, cross-validation is not a good choice in this approach because it is impractical to shuffle the training set and testing set after the model are implemented in hospital. Based on this constraint, to maintain the reliability of testing error, the cumulative training set is implemented. The whole testing set is firstly divided into several sequential segments and then each segment is added to the current training set after the last prediction is finished. In this way, each proposed hybrid model is evaluated five times. A series of hybrid models are established by combining two types of DNNs and Poisson Regression, SARIMA, ARIMAX, respectively. For each pair, 3 hybrid strategies are used and compared with the benchmark algorithm, i.e., the original single model. The metric is set as the Mean Absolute Percentage Error (MAPE) of five testing sets and prediction horizon is fixed as 28 days ahead, which is based on the real demand. The result is shown in Figures 3 to 5. Corresponding values are listed in Tables II to IV, in which relative increment with respect to the single time series regression model is also calculated in order to show the improvement of hybrid models. For the similarity between results achieved by MLP and RNN in all three datasets (see Figures 3 to 5), only values related with RNN are recorded in this paper.

The result demonstrate that all hybrid models perform better than original single models, except for a few abnormal data points. Based on different comparisons between hybrid models and the original single model, it can be concluded that introducing DNNs to linear models benefits the performance with lower MAPE, which validates the predictive power and

---

**TABLE 3. MAPE achieved by involved models and relative increment of hybrid models (for the patient in category 3).**

<table>
<thead>
<tr>
<th></th>
<th>Single SARIMA</th>
<th>Zhang's Method</th>
<th>Khasher's Method</th>
<th>MA Filter Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNN Combined with SARIMA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0823</td>
<td>0.0865 (5.10%)</td>
<td>0.0882 (7.17%)</td>
<td>0.0808 (-1.82%)</td>
<td></td>
</tr>
<tr>
<td>0.0787</td>
<td>0.072 (-8.51%)</td>
<td>0.0673 (-14.49%)</td>
<td>0.0695 (-11.69%)</td>
<td></td>
</tr>
<tr>
<td>0.0798</td>
<td>0.0734 (-8.02%)</td>
<td>0.0738 (-7.52%)</td>
<td>0.0723 (-9.40%)</td>
<td></td>
</tr>
<tr>
<td>0.1051</td>
<td>0.1082 (2.95%)</td>
<td>0.0914 (-13.04%)</td>
<td>0.1059 (0.76%)</td>
<td></td>
</tr>
<tr>
<td>0.1418</td>
<td>0.1393 (-1.76%)</td>
<td>0.076 (-46.40%)</td>
<td>0.1248 (-1.199%)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Single ARIMAX</th>
<th>Zhang's Method</th>
<th>Khasher's Method</th>
<th>MA Filter Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNN Combined with ARIMAX</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0824</td>
<td>0.0821 (-0.36%)</td>
<td>0.0844 (2.43%)</td>
<td>0.1165 (41.38%)</td>
<td></td>
</tr>
<tr>
<td>0.0814</td>
<td>0.0782 (-5.93%)</td>
<td>0.066 (-18.92%)</td>
<td>0.071 (-12.78%)</td>
<td></td>
</tr>
<tr>
<td>0.0788</td>
<td>0.0774 (-1.78%)</td>
<td>0.0643 (-18.40%)</td>
<td>0.0724 (-8.12%)</td>
<td></td>
</tr>
<tr>
<td>0.1198</td>
<td>0.1132 (-5.51%)</td>
<td>0.0977 (-18.45%)</td>
<td>0.0987 (-17.61%)</td>
<td></td>
</tr>
<tr>
<td>0.0857</td>
<td>0.0844 (-5.2%)</td>
<td>0.0768 (-10.39%)</td>
<td>0.1673 (95.22%)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Single PR</th>
<th>Zhang's Method</th>
<th>Khasher's Method</th>
<th>MA Filter Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNN Combined with Poisson Regression</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0807</td>
<td>0.0807 (0.00%)</td>
<td>0.0787 (-2.48%)</td>
<td>0.0798 (-1.12%)</td>
<td></td>
</tr>
<tr>
<td>0.0809</td>
<td>0.0809 (0.00%)</td>
<td>0.0679 (-16.07%)</td>
<td>0.0749 (-7.42%)</td>
<td></td>
</tr>
<tr>
<td>0.0799</td>
<td>0.0799 (0.00%)</td>
<td>0.066 (-17.40%)</td>
<td>0.0808 (1.13%)</td>
<td></td>
</tr>
<tr>
<td>0.1238</td>
<td>0.1238 (0.00%)</td>
<td>0.1006 (-18.74%)</td>
<td>0.1172 (-5.33%)</td>
<td></td>
</tr>
<tr>
<td>0.0766</td>
<td>0.0766 (0.00%)</td>
<td>0.0752 (-1.83%)</td>
<td>0.0752 (-1.83%)</td>
<td></td>
</tr>
</tbody>
</table>

**TABLE 4. MAPE achieved by involved models and relative increment of hybrid models (for the patient in category 4).**

<table>
<thead>
<tr>
<th></th>
<th>Single SARIMA</th>
<th>Zhang's Method</th>
<th>Khasher's Method</th>
<th>MA Filter Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNN Combined with SARIMA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0732</td>
<td>0.0743 (1.50%)</td>
<td>0.0693 (-5.33%)</td>
<td>0.0731 (-0.14%)</td>
<td></td>
</tr>
<tr>
<td>0.0728</td>
<td>0.0705 (-3.16%)</td>
<td>0.0694 (-4.67%)</td>
<td>0.0707 (-2.88%)</td>
<td></td>
</tr>
<tr>
<td>0.0555</td>
<td>0.0563 (1.44%)</td>
<td>0.058 (4.50%)</td>
<td>0.0541 (-2.52%)</td>
<td></td>
</tr>
<tr>
<td>0.1657</td>
<td>0.1461 (-11.83%)</td>
<td>0.0975 (-41.16%)</td>
<td>0.1563 (-5.67%)</td>
<td></td>
</tr>
<tr>
<td>0.1376</td>
<td>0.1188 (-13.66%)</td>
<td>0.0664 (-51.74%)</td>
<td>0.1178 (-14.39%)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Single ARIMAX</th>
<th>Zhang's Method</th>
<th>Khasher's Method</th>
<th>MA Filter Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNN Combined with ARIMAX</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0637</td>
<td>0.0673 (5.65%)</td>
<td>0.0688 (8.01%)</td>
<td>0.0771 (21.04%)</td>
<td></td>
</tr>
<tr>
<td>0.0792</td>
<td>0.0788 (-0.88%)</td>
<td>0.0767 (-3.16%)</td>
<td>0.0731 (-7.70%)</td>
<td></td>
</tr>
<tr>
<td>0.0569</td>
<td>0.0618 (8.61%)</td>
<td>0.0622 (9.31%)</td>
<td>0.0557 (-2.11%)</td>
<td></td>
</tr>
<tr>
<td>0.142</td>
<td>0.1397 (-1.62%)</td>
<td>0.1025 (-27.82%)</td>
<td>0.1553 (9.37%)</td>
<td></td>
</tr>
<tr>
<td>0.1309</td>
<td>0.1277 (-2.44%)</td>
<td>0.056 (-57.22%)</td>
<td>0.0965 (-26.28%)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Single PR</th>
<th>Zhang's Method</th>
<th>Khasher's Method</th>
<th>MA Filter Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNN Combined with Poisson Regression</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.084</td>
<td>0.084 (0.00%)</td>
<td>0.0729 (-13.21%)</td>
<td>0.0846 (0.71%)</td>
<td></td>
</tr>
<tr>
<td>0.1104</td>
<td>0.1104 (0.00%)</td>
<td>0.0934 (-15.40%)</td>
<td>0.1059 (-4.08%)</td>
<td></td>
</tr>
<tr>
<td>0.0522</td>
<td>0.0522 (0.00%)</td>
<td>0.0428 (-18.01%)</td>
<td>0.0526 (0.77%)</td>
<td></td>
</tr>
<tr>
<td>0.1418</td>
<td>0.1418 (0.00%)</td>
<td>0.0867 (-38.86%)</td>
<td>0.1344 (-5.22%)</td>
<td></td>
</tr>
<tr>
<td>0.0573</td>
<td>0.0573 (0.00%)</td>
<td>0.0501 (-12.57%)</td>
<td>0.0546 (-4.71%)</td>
<td></td>
</tr>
</tbody>
</table>
universal property of DNNs indirectly. By comparing the performance obtained by MLP with different models and RNN with these models, it indicates that RNN helps the hybrid model to achieve slightly lower MAPE in some cases (4th testing segment in the case that DNNs combined with Poisson Regression). By comparing three hybrid strategies, generally, Khashei’s method and MA filter-based method achieve lower MAPE than Zhang’s method. As for MA filter-based method, it may occasionally bring some unexpected forecast values on the 5th testing segment in the
case that DNNs are combined with ARIMAX or SARIMA model. By using the Khashei’s method, hybrid models obtain comparatively low MAPEs with the highest robustness. Therefore, the combination between DNN and classical forecast models via Khashei’s method possess strong predictive power and universal property, which can be deployed as a universal tool to conduct patient flow forecasting in Hong Kong A&ED.

V. CONCLUSION
In this study, a series of hybrid models are designed and compared to forecast daily patient flows in a certain A&ED in Hong Kong. The result indicates that combining DNNs and other linear models really improves the forecast accuracy of original single linear models. Theoretically, our study not only validates Hibon’s findings [15], but also proposes new methodologies to utilize DNNs to model time series containing both linear and nonlinear component. Practically, forecasting varying daily patient flow patterns is a significant premise for other decision makings in A&ED, for example, nurse scheduling, doctor rostering, etc. The present forecast models and corresponding combination strategies can be generalized to other kinds of patient flow forecast problem, or other analogous time series forecast problem.

REFERENCES
LONG WANG (S’16–M’17) received the M.Sc. degree in computer science (Hons.) from University College London, London, U.K., in 2014, and the Ph.D. degree in systems engineering and engineering management from the City University of Hong Kong, Hong Kong, in 2017.

He is currently an Associate Professor with the Department of Computer Science and Technology, University of Science and Technology Beijing, Beijing, China. His research interests include machine learning, computational intelligence, and computer vision. He was an Awardee of the Hong Kong Ph.D. Fellowship, in 2014. He serves as an Associate Editor for IEEE Access and the Canadian Journal of Electrical and Computer Engineering. He serves as an Academic Editor for PLOS One. He is also the Lead Guest Editor of data science-related special issues on Intelligent Automation and Soft Computing and Water.

XIONG LUO received the Ph.D. degree from Central South University, Changsha, China, in 2004. From 2005 to 2006, he was with the Department of Computer Science and Technology, Tsinghua University, China, as a Postdoctoral Fellow. From 2012 to 2013, he was with the School of Electrical, Computer, and Energy Engineering, Arizona State University, USA, as a Visiting Scholar. He is currently a Professor with the Department of Computer Science and Technology, School of Computer and Communication Engineering, University of Science and Technology Beijing, China. His research interests include neural computing and machine learning. He has published extensively in his research areas of interest in many journals, such as Future Generation Computer Systems, the IEEE Transactions on Industrial Informatics, the IEEE Transactions on Human–Machine Systems, Computer Networks, and Neurocomputing.

CHAO HUANG (S’16–M’17) received the B.Eng. degree in electrical engineering and automation from the Harbin Institute of Technology, China, 2011, the M.S. degree in intelligent transport systems from the University of Technology of Compiègne, France, in 2013, and the Ph.D. degree in systems engineering and engineering management from the City University of Hong Kong, Hong Kong, 2017, where he was a Postdoctoral Fellow with the Department of Systems Engineering and Engineering Management. He is currently an Associate Professor with the Department of Computer Science and Technology, University of Science and Technology Beijing, Beijing, China.

His research interests include photovoltaics, lithium-ion battery, machine learning, and computational intelligence.

JENQ-HAUR WANG received the Ph.D. degree from National Taiwan University, in 2002. He is currently an Associate Professor with the Department of Computer Science and Information Engineering, National Taipei University of Technology. His research interests include data mining in social networks and computational intelligence.

KWAI-SANG CHIN received the M.Sc. and Ph.D. degrees in industrial engineering from The University of Hong Kong, Hong Kong, in 1990 and 1996, respectively.

He is currently an Associate Professor and the Associate Head of the Department of Systems Engineering and Engineering Management, City University of Hong Kong. He is currently the Programme Leader of the B.Eng. degree in total quality engineering and systems engineering and management. He has published over 180 international refereed journals and conference papers. He has acquired total research funding as the Principal Investigator: HK$10,980,478 in 26 projects.

XIMING NIE received the M.D. degree from Nanfang Medical University, Guangzhou, China. He is currently a Neurologist with the NICU, Beijing Tiantan Hospital, Capital Medical University, Beijing, China.