IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 31, 2021, accepted February 14, 2021, date of publication February 16, 2021, date of current version March 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3059943

A Feature-Cascaded Correntropy LSTM for Tourists Prediction

YUEHAI CHEN[1], (Graduate Student Member, IEEE), JING YANG[1], (Member, IEEE), KUN ZHANG[1], (Graduate Student Member, IEEE), YI XU[1], (Student Member, IEEE), AND YUEWEN LIU[2], (Member, IEEE)

[1]Faculty of Electronic and Information Engineering, School of Automation Science and Engineering, Xi'an Jiaotong University, Xi'an 710049, China
[2]School of Management, Xi'an Jiaotong University, Xi'an 710049, China

Corresponding author: Yuewen Liu (liuyuewen@mail.xjtu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 62073257 and Grant 62041206.

ABSTRACT Forecasting the number of tourists is significant to public safety, which can enable the government to control the sudden influx of tourists timely. The temporal dependence (closeness and period), external factors such as holidays, government policy, as well as outliers in real data, make the prediction challenging. Our data obtain a mixture of short-term contact and long-term repeating patterns and external factors, for which Autoregressive, Exponential Smoothing models and Extreme Learning Machine may fail. In our paper, we propose a novel Feature-Cascaded framework with Correntropy criterion for Long Short-Term Memory network (FC-C-LSTM). For temporal dependence and external factors, we first extract feature information from data in close dates and corresponding period, then integrate them as an independent input of LSTM to simulate temporal pattern and solve the problem of time-lag. In view of an important and unavoidable feature of real datasets, there are amounts of outliers. We adopt the correntropy of Gaussian kernel instead of mean square error as cost function, so that outliers get smaller weights and suppress the influence of outliers back-propagation. Experiments on real tourism datasets of several cities in Yunnan Province show that FC-C-LSTM model achieves the better performance than that of other baselines.

INDEX TERMS Prediction, LSTM, correntropy, outliers.

I. INTRODUCTION

Tourism has become a vibrant new industry, occupying a large share of the tertiary industry. Meanwhile, with the rapid development of tourism, a series of traffic management, risk assessment and public safety problems caused by the rapid increase of urban population flow are also ascending with each passing year. For instance, there was a serious stampede on the Bund in Shanghai on New year's Eve, 2015 due to the influx of tourists and insufficient public safety measures [47]. On the evening of December 31, 2012, large quantities of people streamed into a strip region in the fireworks celebration of new year's Eve in Abidjan, resulting in a catastrophic that kill 61 people. From above two examples, we can find that the situation of massive crowds of people streamed into a strip region is also applicable to the international community. Therefore, accurately predicting the number of tourists that will influx into an area every day, especially on holidays, is crucial for the government to use emergency mechanisms such as traffic control, early warning, or evacuation to mitigate or even prevent these tragedies [46].

Forecast the number of tourists every day often faces a major research challenge, that is, how to capture and exploit the dependence among multiple variables. Specifically, our data often obtain a mixture of short-term contact and long-term repeating patterns, as shown in Fig. 1 which plots the number of tourists in 2015 and 2016 in DaLi City. Apparently, there are two main patterns containing short-term contact and long-term repeating yearly. The former describes relation among temporal close days, while the latter reflects the periodic pattern. Take an example, the number of tourists on December 31st in 2016 whose abscissa is 365 is not only related to December 30th in 2016, but also very similar to last year's December 31st. It is crucial to acquire the periodic information in such cases. Consider short-term pattern only will not only lead to the prediction accuracy not high, but also make the prediction appear time-delay phenomenon. Time-delay phenomenon in other words is, what should have

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

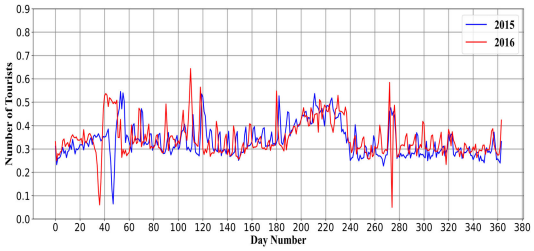


FIGURE 1. The number of tourists in 2015 and 2016 in DaLi City. The x-axis represents the day number in a year, ranking from 1 to 365, and the y-axis represents the normalized number of tourists.

happened would have been true with a lag of a few steps because the model will habitually give greater weight to the step before prediction. A successful time series forecasting model should be able to capture both kinds of patterns for accurate predictions. However, most of traditional approaches such as Auto Regressive Integrated Moving-Average (ARIMA) models and Exponential Smoothing (ES) methods fall short in this aspect, as they do not distinguish between the two patterns, nor do they explicitly and dynamically model their interactions [1]–[9]. Due to the smoothness of these models, they are usually used to simulate the trend of data whose accuracy is one month or one year, which is not suitable for our long-term data of each day.

Therefore, we focus on the artificial neural network (ANN). The artificial neural network (ANN), adjusts the connected relationship between many internal nodes to achieve the goal of information processing without regard to the internal mechanism. Therefore, it has the characteristics of fast response and functions as complex nonlinear approximation, etc. As another example, extreme learning machine (ELM) can well approximate any nonlinear function through randomly generating the learning parameters of neuron in hidden layer and calculating the Moore Penrose generalized inverse matrix [55]. Therefore, ELM has been widely used in regression and prediction [56], [59]–[63]. However, we aim to predict the number of tourist per day under the influence of various factors. As ELM falls short in considering the relevance between the data, the prediction accuracy and generalization performance of the model will be limited. To address such limitations of above methods, we proposed a novel framework that takes advantages of recent developments in deep neural network.

Deep neural networks have been intensively studied in related domains [65], [66]. Compared with the methods above, deep neural networks have the ability to adapt to the rapid changes in the trend by calculating the correlation of observations in the time series. In deep learning research, recurrent neural networks (RNNs) with dynamic memory function are naturally suitable for sequence data modeling because they use the output of this time step as the input of the next time step [10]–[12]. As an excellent variant model of RNN, long short-term memory (LSTM) inherits most of the characteristics of RNN model and achieve good prediction accuracy on their respective time series problems [16]–[20].

Due to the character of natural adaptation sequence data modeling, LSTM have also received an increasing amount of attention in time series analysis [21]–[27]. Unique gate structure makes LSTM suitable for the patterns containing short-term contact and long-term repeating yearly in our task. And in LSTM, researchers can well take the influencing factors into account and usually collect this additional information and use it as part of the input to deal with external factors [28]–[33]. Chang *et al.* collected external information in advance, such as a large-scale event [28]. Then the external data and vehicle detector data are input into the LSTM for training to predict traffic flow. However, it is difficult to collect information of external factors for our task that involves various contents including text, pictures, etc. Time-series decomposition is another popular strategy to process time series with complex factors. Xu *et al.* introduced a decomposed model consisting of trend, seasonality, and holiday components of traffic [29]. In addition, a modified k-means algorithm was proposed to cluster residual holiday data. These methods achieve good performance on dealing with external factors. However, as our holidays are in the lunar calendar, their distribution on dates is irregular. The method of decomposition does not seem to work that well.

Meanwhile, the existence of a large number of outliers will increase the difficulty of prediction, which is an important feature of real datasets. Data cleaning (i.e., removing abnormal data points) and the interpolation-based method are two widely applied approaches to deal with anomalous data points [33]–[36], [39], [49]. Nair *et al.* proposed a hybrid pre-processing technique including k nearest neighbor (kNN) classifier for imbalancing data and outliers, then identified and deleted these outliers [39]. Data cleaning causes serious data loss and reduces the accuracy of the overall trend estimation of the number of tourists because the time scale of the tourist data is not large. Sun *et al.* introduced a least squares support vector machine (LS-SVM) method to realize the recognition of wind turbine abnormal data and selected the appropriate interpolation method to compensate [40]. Although the interpolation-based method does not cause data loss, it loses the dynamic evolution laws of abnormal dates and affects the accuracy of prediction [37]–[40]. This data preprocessing method is a bit cumbersome when faced with large amounts of data. Moreover, researchers will generally calculate the Mean Square Error(MSE) between the predicted value and the ground truth in the presence of Gaussian noise to carry out error back-propagation. It is not robust to data corruption or loss caused by various factors. Since MSE deals with only second-order statistics and minimizes the energy of the error signal so that it is not able to reduce the discrepancy between higher order statistics [41].

In order to solve the limitations mentioned above, we propose a novel feature-cascaded framework for LSTM with correntropy (FC-C-LSTM), as illustrated in Fig. 3. Firstly, we extract the feature information from temporal closeness and corresponding period. Next, the information of two features will be put into an integration as one independent input

of LSTM. The cascaded feature including temporal closeness feature and periodic feature is able to contain and express the impact of these external factors (holidays and government policy) on the number of tourists. Moreover, the cascade method can simulate a mixture of short-term contact and long-term repeating patterns and make the model respond quickly to solve the time-delay problem. In addition, we take correntropy as cost function to reduce influence caused by outliers in real datasets.

Our contributions are as follows:

- FC-C-LSTM can extract the feature information consisting of temporal closeness and corresponding period. FC-C-LSTM cascades the two different feature information to model long- and short-term temporal patterns for ensuring the model's prediction accuracy and solving temporal delay problem.
- We develop the correntropy with Gaussian Kernel function as cost function to reduce impacts on the model's prediction accuracy caused by outliers.

The rest of this paper is organized as follows. Section 2 outlines the related works, including ARIMA models, ES models, ELM models and deep neural networks (DNN) models. Section 3 presents the related concepts of LSTM and correntropy. Section 4 defines our problem and describes our proposed FC-C-LSTM framework. Section 5 presents the study of our experiments and section 6 concludes this paper.

II. RELATED WORKS

The ARIMA model is one of the most popular univariate models for time series forecasting [3]–[5]. ARIMA model is flexible enough to subsume other types of time series models including autoregression(AR), moving average (MA) and Autoregressive Moving Average(ARMA) [4]. Zhu proposed an ARIMA prediction model for Shanghai metro passenger flow based on n-day average traffic volume [6]. In an early study developed by Kharista *et al.*, Gray Model (1,1) and ARIMA were introduced to forecast foreign tourists to Indonesia each year [7].

Exponential Smoothing (ES) is another common forecasting method. For example, the approach using Holt's Weighted Exponential Moving Average (H-WENA) was presented by Hansun *et al.* to forecast the domestic tourist arrivals to Bali province each month [8]. Wang proposed the Holt-Winters' seasonal method to adapt to the short-term forecast of urban rail transit passenger flow [9].

Extreme learning machine (ELM) was proposed by Huang *et al.* in 2004 [58]. Different from the other artificial neural network, the parameters of neuron in the hidden layer are randomly generated, while the output weights are determined by calculating the Moore Penrose generalized inverse matrix [57]. Therefore, ELM has the characteristic of function as complex nonlinear approximation and has been widely applied to build the prediction model such as temperature prediction [59], electricity price prediction [60] and traffic flow prediction [61]–[63].

Deep neural networks have been intensively studied in predictive domains [65], [66]. For instance, a research predicted tourist arrival by examining time-series data on tourist arrival in Lombok by using Recurrent Neural Network with a training algorithm Extended Kalman Filte [10]. Hu *et al.* proposed a deep neural network featuring spatial RNN, which models the spatial dependency of pixels as sequential dynamics to generate better prediction signals in video coding [11]. Lotfidereshgi *et al.* introduced an adaptive RNN that captured all sorts of dependencies between samples for speech signal prediction [12]. Due to the advantage of short-term memory, RNN can achieve high precision for some sequential tasks. Although RNNs can connect the previous information to the current task, they are powerless to process the long-distance dependence well. When the time step is large, RNNs inevitably encounter gradient explosion and gradient disappearance. To deal with this problem, Hochreiter *et al.* proposed LSTM based on RNN modification [13].

LSTM is an excellent variant model of RNN, which inherits most of the characteristics of RNN model and solves the problem of gradient disappearance caused by the gradual decrease of gradient in back-propagation process [14]. More recently, LSTM has attracted considerable attention among researchers studying prediction problems. Particularly, LSTM has presented promising results, outperforming many state-of-the-art statistical forecasting methods [6]–[9]. There is substantial literature available on LSTM for time-series forecasting, including forecasting the number of tourists [15]–[20]. Poonia *et al.* had applied the Long Short-Term Memory Networks (LSTM) for instantaneous traffic stream forecast [15]. The model can memorize information for a long period of time and provide an appropriate decision basis for traffic congestion prediction in peak hours. By considering the short-term and long-term factors, Yu *et al.* proposed a hybrid model of CNN-LSTM to predict the number of passengers dispatched [16]. In order to learn the importance of each past value to the current value from the long sequence of traffic data at the past moment, Chen *et al.* improved LSTM based on attention mechanism [17]. Here, the improved model was able to extract more valuable features. The model proposed by Chen *et al.* can memorize information for a long period of time and provide an appropriate decision basis for traffic congestion prediction in peak hours [19].

III. PRELIMINARY

Our method is improved on the basic framework of LSTM. The cost function to make outliers get smaller weight and reduce the impact of outlier back-propagation to obtain better prediction accuracy.

A. LSTM

LSTM was proposed by Hochreiter *et al.* to solve the gradient vanishment or gradient explosion issue [13], [64]. LSTM has various successful application in dealing with highly related problems in time series such as Natural Language

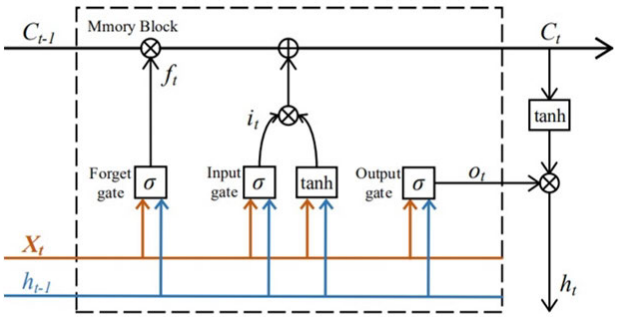


FIGURE 2. LSTM model.

Process [50], [51], Dialogue Generation [52], Encoding and Decoding [53], [54], and so on. The cell structure of LSTM model is shown in Fig. 2. LSTM adds gates to control the memory and forgetting of temporal information. Suppose that in each time step t, X_t is the input vector and C_t is the cell state vector, and h_t is the hidden state vector of the cell. σ is the sigmoid activation function. Forget gate is used to determine the extent of retention of the previous cell state. The forget gate is calculated by (1):

$$f_t = \sigma(W_f \cdot [h_{t-1}, X_t] + b_f) \tag{1}$$

where W_f represents weight matrix of forget gate, b_f represents bias matrix of forget gate.

Next, determine what information is stored in the LSTM unit, which contains two parts. First, input gate uses sigmoid activation function σ to determine the extent to which the current cell state is retained. Second, a new candidate value vector $\tilde{C}_t$ is created by the tanh layer as shown in (3). The new state C_t of the cell will be updated by the old state C_{t-1} after being forgotten and a new candidate value vector $\tilde{C}_t$ shown in (4). The input gate is calculated by (2), (3), (4):

$$i_t = \sigma(W_i \cdot [h_{t-1}, X_t] + b_i) \tag{2}$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, X_t] + b_c) \tag{3}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4}$$

where W_i and W_c represent weight matrix of input gate, b_i and b_c represent bias matrix of input gate.

Finally, the status of the LSTM cell unit is updated by the output gate. First, using a sigmoid activation function σ to determine which parts of the current cell state C_t require output. Then, the cell state C_t is processed by tanh, and multiplied by the sigmoid activation function output o_t to obtain the output value. The output gate is calculated by (5) and (6):

$$o_t = \sigma(W_o \cdot [h_{t-1}, X_t] + b_o) \tag{5}$$

$$h_t = o_t * \tanh(C_t) \tag{6}$$

where W_o represents weight matrix of output gate, b_o represents bias matrix of output gate.

The LSTM evolved from the RNN. It solves the problem that traditional RNN is sensitive only to short-term memory, but "gradient disappears" in long-term sequence. In our work, we extend the LSTM network as feature extractor to capture features from temporal closeness and corresponding period.

B. CORRENTROPY

Correntropy is a measurement standard information relevance introduced by (7) [41]–[44]:

$$V(X, Z) = \iint k_\delta(x, z) P_{x,z}(x, z)\, dz \tag{7}$$

where $k_\delta(x, z)$ is any positive definite Mercer kernel function of the random variables x and z variables, $P_{x,z}(x, z)$ is the joint Probability Density Function (PDF) of them.

In the past few years, there has been a growing interest in the information theoretic learning measure called correntropy. The maximum correntropy criterion (MCC) has been proven to perform robust adaptation in adaptive filter training when the additive noises are heavy-tailed nonGaussian [43]. Heravi *et al.* proposed correntropy-based conjugate gradient backpropagation algorithms that had better performance than the common conjugate gradient backpropagation based on Mean Square Error, especially in nonGaussian environments and in cases with impulsive noise or heavy-tailed distributions noise [44]. A gradient-based correntropy algorithm is widely applied in signal processing and machine learning due to its robustness against outliers.

IV. METHODS

A. PROBLEM DEFINITION

In this paper, we are interested in the task of number of tourists forecasting. The factors of date and holidays are important for our task. Consider $\mathbf{i}_t = (y_t, l_t, d_t)^T$, $\mathbf{i}_t \in \mathbf{R}^3$ is the state variable, where y_t, l_t and d_t are number of tourists, labels and date marking at day t separately. The label indicates whether the date is a holiday or not and date markings contain date information. Then, we describe the problem of predicting the visitors flow of tourists as (8):

$$\mathbf{y} = f(\mathbf{P}, \overline{\mathbf{P}}, \mathbf{W}^*) \tag{8}$$

where $\mathbf{y} = (y_{obs+1}, y_{obs+2}, \ldots y_{obs+pred})$ is the future number of tourists at day step $t = obs + 1, \ldots, obs + pred$. $f(\cdot)$ is the prediction model. $\mathbf{P} = (\mathbf{i}_1, \mathbf{i}_2, \ldots, \mathbf{i}_{obs})$ is the data of this year at time $t = 1, \ldots, obs$ and $\overline{\mathbf{P}} = (\bar{\mathbf{i}}_1, \bar{\mathbf{i}}_2, \ldots, \bar{\mathbf{i}}_{obs})$ is the corresponding data of last year. $\mathbf{W}^*$ represents all parameters learned in the model.

Our purpose is to learn the parameters $\mathbf{W}^*$ of the model $f(\cdot)$ in order to predict the future number of tourists $\mathbf{y}$ in the future between $t = obs + 1$ and $obs + pred$.

In order to solve the problem of the above definition, we design a novel feature-cascaded framework for LSTM with correntropy cost function (FC-C-LSTM) which is shown in Fig. 3. The specific functions are introduced in Part B.

B. FC-C-LSTM FRAMEWORK

The concept map of our proposed framework is shown in the Fig. 3. FC-C-LSTM framework consists of four key

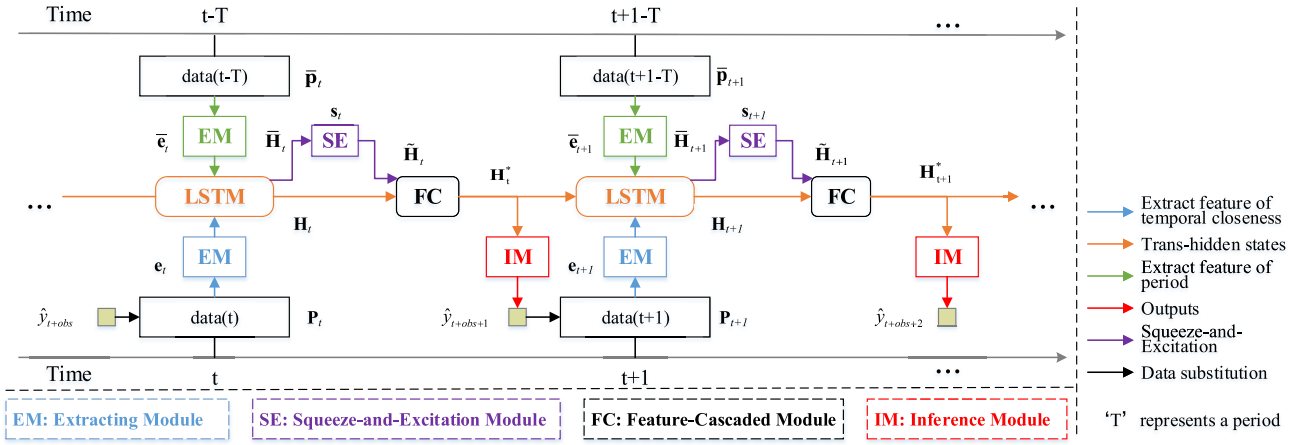


FIGURE 3. The framework of the proposed FC-C-LSTM.

components: (1) Extracting Module (EM), (2) Squeeze-and-Excitation Module (SE) [45] (3) Feature-Cascaded Module (FC), (4) Inference Module (IM).

In our problem, the information of the tourists $\mathbf{i}_t = (y_t, l_t, d_t)^T$, $\mathbf{i}_t \in \mathbf{R}^3$ includes the number of tourists y_t, label for holidays l_t (indicates whether the date is a holiday or not), date marking d_t (indicates the location of the day in a year). First, the Extracting Module is designed for capturing dependence between time series to get feature information $\mathbf{H}_t$ and $\overline{\mathbf{H}}_t$. Then, the hidden states $\mathbf{H}_t = \{\mathbf{h}_{t+1}, \ldots, \mathbf{h}_{t+obs}\}$ stand for the feature information of the number of tourists $\mathbf{P}_t = (\mathbf{i}_{t+1}, \mathbf{i}_{t+2}, \ldots, \mathbf{i}_{t+obs})$ at time $\{t+1, \ldots, t+obs\}$ are extracted. In the same way, the feature information in last year that is represented as $\overline{\mathbf{H}}_t = \{\overline{\mathbf{h}}_{t+1}, \ldots, \overline{\mathbf{h}}_{t+obs}\}$. As the feature information $\overline{\mathbf{H}}_t = \{\overline{\mathbf{h}}_{t+1}, \ldots, \overline{\mathbf{h}}_{t+obs}\}$ in last year is similar to that $\mathbf{H}_t = \{\mathbf{h}_{t+1}, \ldots, \mathbf{h}_{t+obs}\}$ of this year. We put feature information $\overline{\mathbf{H}}_t$ into Squeeze-and-Excitation Module to get $\tilde{\mathbf{H}}_t$ to make that effective feature information has a large weight. And then, we get the joint feature information $\mathbf{H}_t^*$ by cascading extracted feature information $\mathbf{H}_t$ with the processed feature information $\tilde{\mathbf{H}}_t$ in the Feature-Cascaded Module. Finally, we pass the joint feature information $\mathbf{H}_t^*$ through the Inference Module getting the predicted value $\hat{y}_{t+obs+1}$ to estimate the actual number of tourists $y_{t+obs+1}$ at time $t + obs + 1$.

1) EXTRACTING MODULE

In order to extract feature information and dependence of the tourists in the temporal closeness and corresponding period, we use the LSTM network to capture temporal dependence of the number of tourists and map them to higher dimensional feature space.

In our case, the state information $\mathbf{P}_t = (\mathbf{i}_{t+1}, \mathbf{i}_{t+2}, \ldots, \mathbf{i}_{t+obs})$ at time $\{t+1, \ldots, t+obs\}$ is embedded into a vector $\mathbf{e}_t$ as shown in (9):

$$\mathbf{e}_t = \phi(\mathbf{P}_t; \mathbf{W}_e) \tag{9}$$

where ϕ is the embedding function (Rectified Linear Unit, ReLU), $\mathbf{W}_e$ are the embedding parameters.

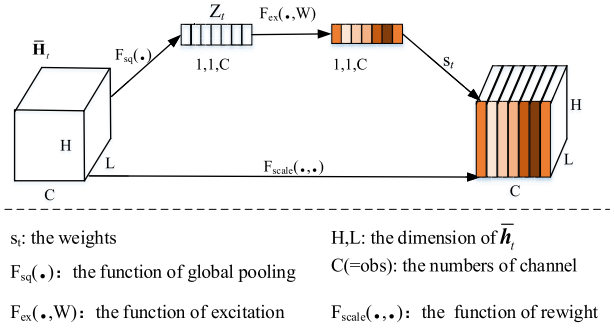


FIGURE 4. Squeeze-and-excitation module.

The vector $\mathbf{e}_t$ stands for the temporal feature information of the number of tourists at time $\{t+1, \ldots, t+obs\}$ and is the output of the extracting module. We get the vector $\overline{\mathbf{e}}_t$ in the same way. $\mathbf{e}_t$ is defined as one input of LSTM network in the Fig. 3. And we get the hidden state $\mathbf{H}_t$ and $\overline{\mathbf{H}}_t$ and by putting the $\mathbf{e}_t$ and $\overline{\mathbf{e}}_t$ into LSTM cell:

$$\mathbf{H}_t = LSTM(\mathbf{H}_{t-1}, \mathbf{e}_t, \mathbf{W}^*) \tag{10}$$
$$\overline{\mathbf{H}}_t = LSTM(\overline{\mathbf{H}}_{t-1}, \overline{\mathbf{e}}_t, \mathbf{W}^*) \tag{11}$$

where $\mathbf{W}^*$ are the parameters of the LSTM, as stated in (1) to (7) in the preliminary of LSTM.

2) SQUEEZE-AND-EXCITATION MODULE

In order to extract more refined feature information, we consider the implementation by dealing with the dependence between channels. The Squeeze-and-Excitation block is a computational unit which can tackle the issue of exploiting channel dependencies [45]. A diagram of an SE building block including two main operations is shown in Fig. 4.

- Squeeze: It can capture the data features from the global receptive field to describe the data by using global pooling. Formally, a statistic variable $\mathbf{Z}_t \in R^C$ $(C = obs)$ is generated by shrinking $\overline{\mathbf{H}}_t = [\overline{\mathbf{h}}_{t+1}, \overline{\mathbf{h}}_{t+2}, \ldots, \overline{\mathbf{h}}_{t+obs}]$ through spatial dimensions $H \times L$, where the c-th

element of $\mathbf{Z}_t$ is calculated by (12):

$$\mathbf{z}_c = F_{sq}(\overline{\mathbf{h}}_c) = \frac{1}{H \times L} \sum_{i=1}^{H} \sum_{j=1}^{L} \overline{h}_c(i,j) \tag{12}$$

where $\mathbf{Z}_t = [\mathbf{z}_{t+1}, \mathbf{z}_{t+2}, \ldots, \mathbf{z}_{t+obs}]$

- Excitation: To make use of the information aggregated in the squeeze operation, we follow it with a second operation which aims to fully capture channel-wise dependencies. The importance of each feature channel is calculated by (13):

$$\mathbf{s}_t = F_{ex}(\mathbf{Z}_t, W) = \sigma(W_2 \phi(\mathbf{Z}_t, W_1)) \tag{13}$$

where $\mathbf{s}_t = [s_{t+1}, s_{t+2}, \ldots, s_{t+obs}]$ refers to the weight, σ is a sigmoid activation, ϕ refers to the ReLU function. $W_1, W_2 \in R^{C \times C}$ are the parameters of fully connected layer. The final output of the block $\tilde{\mathbf{H}}_t$ is obtained by rescaling the output $\overline{\mathbf{H}}_t$ with the activations $\mathbf{s}$ shown as in (14).

$$\tilde{\mathbf{h}}_{\mathbf{c}} = F_{scale}(\overline{\mathbf{h}}_{\mathbf{c}}, s_c) = \overline{\mathbf{h}}_{\mathbf{c}} \cdot s_c \tag{14}$$

where $\tilde{\mathbf{H}}_t = [\tilde{\mathbf{h}}_{t+1}, \tilde{\mathbf{h}}_{t+2}, \ldots, \tilde{\mathbf{h}}_{t+obs}]$, $F_{scale}(\overline{\mathbf{h}}_c, s_c)$ refers to channel-wise multiplication between the feature map $\overline{\mathbf{h}}_c \in R^{H \times L}$ and the scalar s_c

The activations $\mathbf{s}_t$ act as channel weights adapted to the input-specific descriptor $\mathbf{Z}_t$. In this regard, SE blocks intrinsically introduce dynamics conditioned on the input, helping to boost feature discriminability.

3) FEATURE-CASCADED MODULE

we design a cascaded feature module to obtain the feature information of the previous spans and corresponding period. Note that the hidden state of the LSTM $\mathbf{H}_t = \{\mathbf{h}_{t+1}, \ldots, \mathbf{h}_{t+obs}\}$ of the number of the tourists in the previous span, $\tilde{\mathbf{H}}_t = \{\tilde{\mathbf{h}}_{t+1}, \ldots, \tilde{\mathbf{h}}_{t+obs}\}$ of the number of the tourists in corresponding period which gotten from SE module. The hidden states $\{\mathbf{h}_{t+1}, \ldots, \mathbf{h}_{t+obs}\}$ and $\{\tilde{\mathbf{h}}_{t+1}, \ldots, \tilde{\mathbf{h}}_{t+obs}\}$ are put into the cascaded feature module for integration. The hidden state dimensions of the two feature information are the same and set as D, the hidden states $\mathbf{H}_t^*$ is calculated shown as in (15).

$$\mathbf{H}_t^* = \overrightarrow{\alpha_t} \cdot \mathbf{H}_t + \overrightarrow{\beta_t} \cdot \tilde{\mathbf{H}}_t \tag{15}$$

where $\overrightarrow{\alpha_t}$, $\overrightarrow{\beta_t}$ representing the integration factors of $\mathbf{H}_t$ and $\tilde{\mathbf{H}}_t$ are the learnable parameters that adjust the degrees affected by closeness and period respectively. So the $\mathbf{H}_t^*$ extracts not only the information in the previous span but also information in corresponding period. Then the $\mathbf{H}_t^*$ will be put into Inference Module.

4) INFERENCE MODULE

In Inference Module, we use correntropy based on a Guassian kernel $G_\delta(x) = \frac{1}{\sqrt{2\pi}\delta} e^{\frac{-x^2}{2\delta^2}}$ to calculate the loss between predicted values and ground truth. The predicted value $\hat{y}_{t+obs+1}$ at time $t+obs+1$ is determined by the hidden state $\mathbf{H}_t^*$ shown as in (16):

$$\hat{y}_{t+obs+1} = \varphi(\mathbf{H}_t^*, W_r) \tag{16}$$

where φ is the function of dimension reduction, $\mathbf{W}_r$ are the dimension reduction parameters.

Our model is trained by minimizing correntropy $C(\hat{\mathbf{y}}, \mathbf{y})$ based on a Guassian kernel $G_\delta(X, Z)$ shown as in (17):

$$C(\hat{\mathbf{y}}, \mathbf{y}) = \iint G_\delta(\hat{y}, y) P_{\hat{y},y}(\hat{y}, y)\, dy \tag{17}$$

where $P_{\hat{y},y}(\hat{y}, y)$ is the joint Probability Density Function (PDF) of $\hat{y}$ and y, y is the ground truth, $\hat{y}$ is the predicted value, δ is the parameter of the Gaussian function representing the bandwidth of Gaussian function (this parameter choice is discussed in sec. V-D). In practice, the joint PDF is unknown. Therefore, according to Parzen-Windows method, we rewrite the function shown as in (18):

$$C_\delta(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{N} \sum_{n=1}^{N} G_\delta(\hat{y}(n) - y(n)) \tag{18}$$

where N is the length of the predicted values $\hat{\mathbf{y}}$.

Next, we explain the reason that correntropy can alleviate the problems caused by outliers. For a linear logistics models, assume $\mathbf{h}(\theta) = \sum_{j=1}^{n} \theta_j x_j$ is the fitting function and n is the number of features. $J(\theta)$ is the loss function for Mean Squared Error (MSE) and $C_\delta(\theta)$ is the loss function for correntropy. Consider common loss MSE and correntropy for batch gradient descent(BGD):

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} (y^i - h_\theta(x^i))^2 \tag{19}$$

$$C_\delta(\theta) = \frac{1}{m} \sum_{n=1}^{m} G_\delta(y^i - h_\theta(x^i)) \tag{20}$$

where m is the number of training set samples. y^i is the prediction value in the i-th training set sample. $h_\theta(x^i)$ is the desired outputs in the i-th training set sample. G_δ is Guassian kernel function.

Deviate loss function J and C from θ to get the gradient for each θ:

$$\frac{\partial J(\theta)}{\partial \theta} = -\frac{2}{m} \sum_{i=1}^{m} (y^i - h_\theta(x^i)) x_j^i \tag{21}$$

$$\begin{aligned} \frac{\partial C_\delta(\theta)}{\partial \theta} &= -\frac{2}{m} \sum_{i=1}^{m} \left\{ \frac{1}{\sqrt{2\pi}\delta} e^{\frac{-(y^i - h_\theta(x^i))^2}{2\delta^2}} \frac{(y^i - h_\theta(x^i)) x_j^i}{2\delta^2} \right\} \\ &= \sum_{i=1}^{m} \left\{ \frac{1}{\delta^2 \sqrt{2\pi}\delta} e^{\frac{-(y^i - h_\theta(x^i))^2}{2\delta^2}} \right\} \frac{\partial J(\theta)}{\partial \theta} \end{aligned} \tag{22}$$

Since the risk function is to be minimized, update each θ according to the negative gradient direction of each parameter θ:

$$\theta_j' = \theta_j + \frac{2}{m} \sum_{i=1}^{m} (y^i - h_\theta(x^i)) x_j^i \tag{23}$$

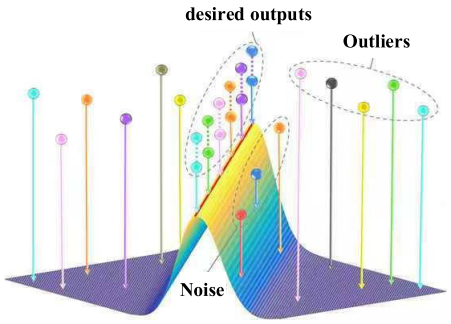


FIGURE 5. Correntropy with Gaussian kernel function for outliers.

$$\theta_g^{'} = \theta_g + \frac{2}{m}\sum_{i=1}^{m}(y^i - h_\theta(x^i))x_j^i \left\{ \frac{1}{\delta^2\sqrt{2\pi}\delta} e^{\frac{-(y^i - h_\theta(x^i))^2}{2\delta^2}} \right\} \tag{24}$$

When $h_\theta(x^k)$ is outlier, the value of $\|y^k - h_\theta(x^k)\|$ is large. In our perception, the $h_\theta(x^k)$ should not be involved in BP. While using MSE, θ will update to wrong direction. For correntropy, the update direction $\frac{\partial C_\delta(\theta)}{\partial\theta}$ will distribute smaller weights to outliers as shown in Fig. 5.

The correntropy with Guassian kernel function in the defined (18) reaches the maximum when $\hat{\mathbf{y}} = \mathbf{y}$ so that it can be used for measuring the similarity of two random sequences in adaptive learning. In addition, using the correntropy with Guassian kernel function as cost function is able to obtain kernel Hilbert Space properties. Furthermore, correntropy is able to overcome outliers or noises by distributing smaller weights. This means the predicted value contribute little to the loss back propagation.

Note that the loss is calculated at every time step and tuning the parameters to minimize the loss in the training datasets.

V. EXPERIMENT AND ANALYSIS

In this section, we demonstrate the experiment results of our proposed method in six datasets (GuCheng, GuanDu, DaLi, XiShan, PanLong and JingHong, the cities of Yunnan Province). The datasets contain data on the number of tourists in each city from January 2015 to March 2019. These datasets are real data collected manually, so they contain a large number of outliers. Moreover, due to the fact that these datasets come from the real world, holiday, policy, periodicity and other complex factors have seriously affected the accuracy of the prediction. A lot of experiments show that our method can achieve better performance and accuracy than ARIMA, ELM and LSTM algorithms on these real datasets.

A. EVALUATION METRICS

We use the following metric in (25) to evaluate the performance of our method. Assume *pred* is the number of days we predict in the testing process, N is the number of samples, $\hat{y}_t^i$ is the predicted number of the tourists at time t in i-th sample, y_t^i is the observed number of the tourists at time t in i-th sample.

1) AVERAGE NUMBERS ERROR (ANE)

This error calculates the mean quantity difference between all predicted number and the actual number in testing process.

$$ANE = \frac{\sum_{i=1}^{N}\sum_{t=obs+1}^{obs+pred}(|y_t^i - \hat{y}_t^i|)}{N \times pred} \tag{25}$$

2) BASELINE

We compare our model with several representative existing models:

- The Autoregressive Integrated Moving Average model (ARIMA).
- The Extreme Learning Machine model (ELM).
- The ordinary long short-term memory network model using Mean Square Error as cost function (LSTM).
- The ordinary long short-term memory network model using Mean Absolute Error as cost function (M-LSTM).

3) OUR MODELS

- The long short-term memory network with correntropy criterion(C-LSTM).
- The feature-cascaded long short-term memory network with Mean Absolute Error (FC-M-LSTM).
- The feature-cascaded long short-term memory network with correntropy criterion (FC-C-LSTM).

B. IMPLEMENTATION DETAIL

During the training process, we use many to one approach (we use data $(\mathbf{i}_{t+1}, \mathbf{i}_{t+2}, \ldots, \mathbf{i}_{t+obs})$ to predict the number of tourists $y_{t+obs+1}$ at time $t + obs + 1$) where we train and validate our model on the data of 2015 to 2017 in the dataset. For the number of tourists y_t, we normalize them to the interval [0,1], and the results shown below are all normalized data.

In the training process, we extract the feature information in 12 days observed and predict the number in the next 8 days (the choice of observation days is discussed in sec. V-C). We set the dimension of the hidden states to 256 for the LSTM model. All the inputs are embedded into a 128-dimension vector with ReLU nonlinearity. The batch size is 8 and the model is trained for 250 epochs using Adam with an initial learning rate of 0.003. The bandwidth of Guassian kernel function is 0.8 for all trainings.

During testing process, we use our model to predict the number of tourists. From time $t = obs + 1$ to $t = obs + pred$, we replace the actual number y_t with the predicted $\hat{y}_t$ to make a new prediction.

C. SELECTION OF OBSERVATION DAYS

Different choice of numerical values for observation days as model input will affect the predictive results. In this part, we first select *obs* observation days to explore the correlation between model input and forecast error **ANE**; then, we analyze influence of choosing different numerical observation days as model input on the predictive results.

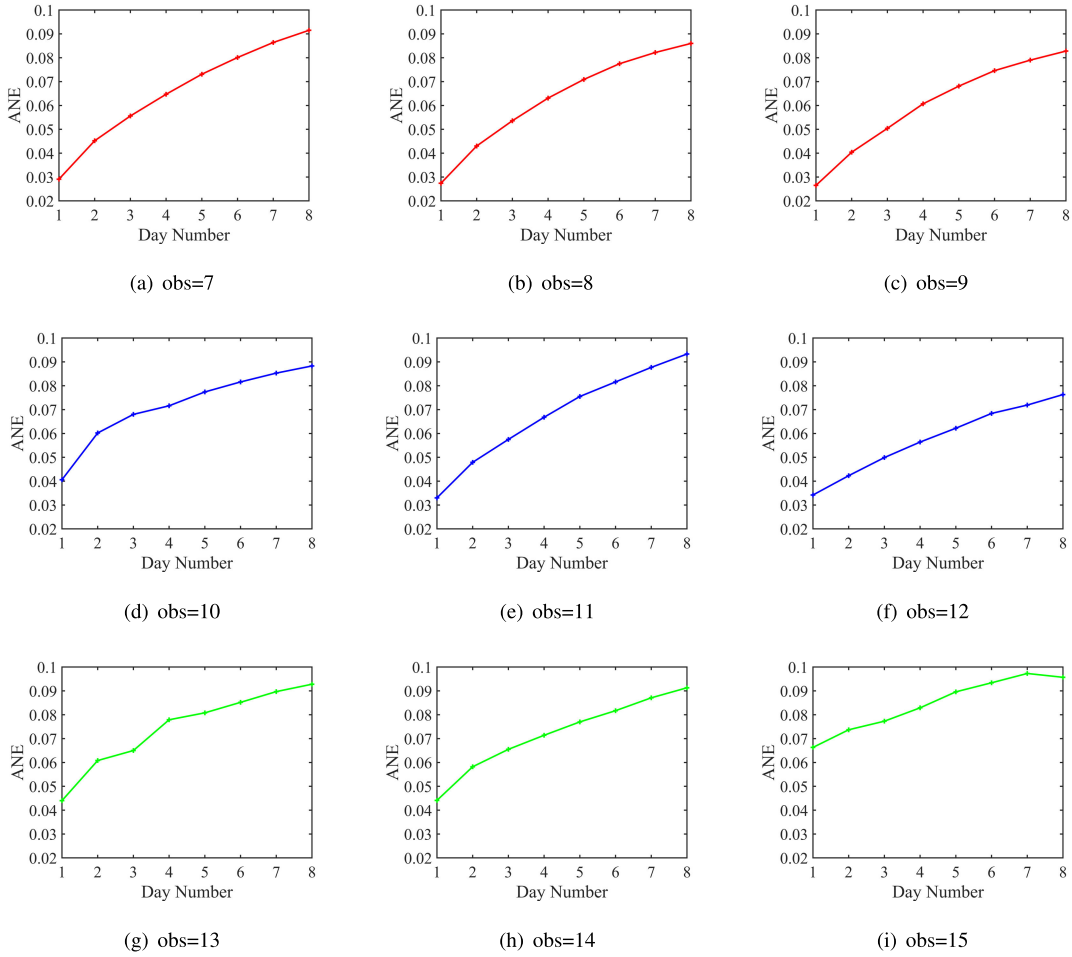


FIGURE 6. Variations of the error of each frame. The 'obs' means observation days.

In our problem, important legal holidays such as May Day, National Day and Spring Festival, have three to seven days holidays. Therefore, we choose 8 days as the forecast days leaving some days' margin to achieve the short-term forecast, so that government can get the information of the number of holiday passengers in advance and take control measures.

Moreover, to quantitatively assess the best value of *obs* determining the predictive results, we represent the average numbers error **ANE** between the number of actual tourists and the number of predicted tourists in GuCheng City in Fig. 6. And Fig. 7 shows the average of **ANE** of different observation days in GuCheng and GuanDu Cities. In order to explain this problem conveniently, we simplify the prediction model to linear model (26) and (27).

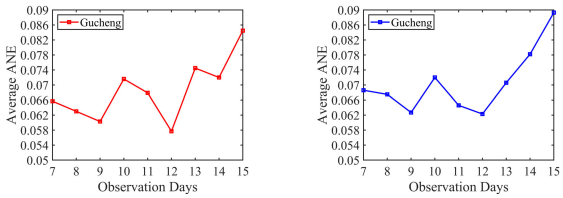
$$\hat{y}_{obs+1} = \sum_{t=1}^{obs} \hat{i}_t * W_t \tag{26}$$

$$\hat{y}_{obs+k} = \sum_{t=k}^{obs} \hat{i}_t * W_t + \sum_{t=obs+1}^{obs+k-1} y_t * W_t \tag{27}$$

$$\varepsilon_{obs+k} = \sum_{t=k}^{obs} \hat{i}_t * W_t + \sum_{t=obs+1}^{obs+k-1} \varepsilon_t * W_t \tag{28}$$

where $\hat{y}_{obs+1}$ is the forecast number of tourists at day step $t = obs+1$. $\hat{i}$ is the model input. W_t represents corresponding parameters learned in the model. *obs* represents observation days. $\varepsilon_{obs+k} = \left|\hat{y}_{obs+k} - y_{obs+k}\right|$ is the ANE between forecast number of tourists $\hat{y}_{obs+k}$ and actual number of tourists y_{obs+k} at day step $t = obs + k$. $k = 2, \ldots, pred$ represents the forecast days.

As shown in Fig. 6 and Fig. 7(a), when observation days is smaller ($obs = 7 \sim 9$), the ANE of the prediction results on the first day will be smaller. This finding shows that the close dates have a strong impact on the predication and when the number of observation days is small, the model will assign a larger weight W_t to each input $\hat{i}_t$. It also meets the real situation, the closer the date is, the closer the relationship is. As observation day gradually increases ($obs = 10 \sim 12$), the average ANE of our model decreases rapidly. The reason is that our prediction method in (27) is to input the prediction results into the model to participate in the later prediction. In the case, the weight of input W_t is small, so the error of input $\varepsilon_t * W_t$ in (28) is small in the next prediction. When observation days is large ($obs = 13 \sim 15$), average ANE of our model increases rapidly, proving the large observation days begins to deviate from the real situation which results



(a) Average ANE in GuCheng (b) Average ANE in GuanDu

FIGURE 7. The results of choosing different observation days in GuCheng and GuanDu cities.

TABLE 1. Experiment for correntropy with Guassian kernel function.

Value (δ)	ANE
0.1	0.0644
0.2	0.0774
0.3	0.0780
0.4	0.0622
0.5	0.0595
0.6	0.0646
0.7	0.0669
0.8	**0.0577**
0.9	0.0648

in large initial error ε_t. As the weight of input W_t and the error of input ε_t are coupled, the accumulated error in (28) will become large in this case. We find that when the value of observation days is 12 days, the average number error(**ANE**) is the minimum in Fig. 7. And finally we choose 12 observation days as the input of the model.

D. BANDWIDTH OF GAUSSIAN FUNCTION

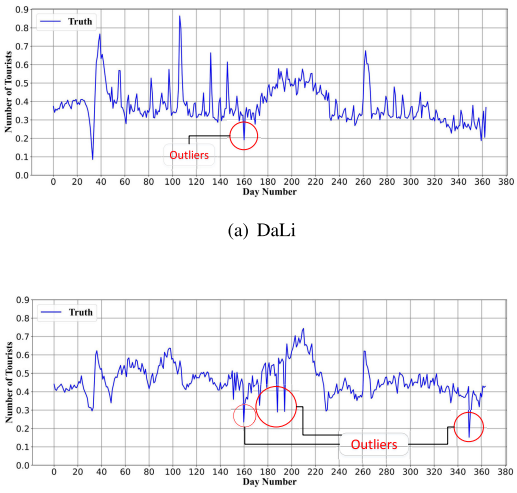
Because Mean Square Error does not deal with outliers well that have a great effect on the tourist prediction. Thus, we introduce Correntropy with Guassian kernel function to train the model. The bandwidth δ of guassian kernel function can control the local range of Gaussian kernel function. Different choice of bandwidth δ will affect the prediction results. Through the experiment, we find that the choice of hyper parameter δ has little difference for the data of all cities. Thus we represent nine different settings of hyper parameter δ in the experiment of GuCheng City as a representative in Table 1. The proper $\delta = 0.8$ achieves the best result in ANE, which is 5.77%. And the maximum ANE of $\delta = 0.3$ is 7.80%. In particular, we found that using a proper $\delta = 0.8$, the average number error is reduced by 2.03% compared with $\delta = 0.3$, we use this value for all experiments.

E. QUANTITATIVE ANALYSIS

1) COMPARISON BETWEEN LSTM, M-LSTM AND C-LSTM

As shown in Table 2, the average ANEs of LSTM and M-LSTM model on the six datasets are 8.18% and 7.84%. C-LSTM model achieves 7.16% for the average ANE. Therefore, on these 6 datasets, the average ANE of the C-LSTM model is 1.02% and 0.68% higher than that of the LSTM model and M-LSTM model.

For DaLi in Table 2, the results of LSTM, M-LSTM and C-LSTM are not much different. In order to find out the reason, the number of tourists in DaLi City is presented in Fig. 8(a). The results show that compared with GuanDu City, shown in Fig. 8(b), there are fewer outliers in the number of tourists in DaLi City. This explains why the performance of LSTM and C-LSTM in DaLi is not much different.



(a) DaLi

(b) GuanDu

FIGURE 8. The number of tourists in DaLi and GuanDu cities.

The experimental results show that the correntropy with Gaussian kernel as the cost function can solve the outlier problem to a certain extent. The outliers will be given a small weight which makes the outliers contribute little to loss back propagation.

2) COMPARISON BETWEEN CF-C-LSTM AND OTHERS

As shown in Table 2, the average ANEs of ARIMA, ELM, LSTM, M-LSTM, C-LSTM and FC-M-LSTM models on the six datasets are 9.38%, 8.84%, 8.18%, 7.84%, 7.16% and 6.49%, respectively. And the average ANE of FC-C-LSTM model is 5.90%. So the FC-C-LSTM model achieve the better performance than ARIMA, ELM, LSTM, M-LSTM, C-LSTM and FC-M-LSTM, increasing 3.48%, 2.94% 2.28%, 1.94%, 1.26%, 0.59% for the average ANE on all datasets.

With the guidance of cascaded feature, FC-C-LSTM model is able to capture the periodic information, which helps to improve the prediction accuracy.

However, in the experiment of PanLong City, the C-LSTM model performs better than the FC-C-LSTM. In order to find out the reason, the predictive results in PanLong City is shown in Fig. 9 and we compares PanLlong city data with any of the other two cities in Fig. 10. There is no particular change in the number of tourists in the holidays in PanLong City. Meanwhile, the number of tourists in two consecutive years does not match well, because the peak data may correspond to the trough on the same date. Thus the performance of the FC-C-LSTM model may be worse than that of the C-LSTM model. Therefore, before forecasting, we need to analyze the data, such as periodicity and holiday impact, and then select the appropriate model for prediction.

TABLE 2. Quantitative results of baselines and our models on all datasets.

Dataset	Performance(ANE)						
	ARIMA	ELM	LSTM	M-LSTM	C-LSTM(OURS)	CF-M-LSTM(OURS)	CF-C-LSTM(OURS)
GuCheng	0.0972	0.0807	0.0865	0.0843	0.0705	0.0665	**0.0577**
GuanDu	0.1023	0.0655	0.0895	0.0748	0.0771	0.0684	**0.0623**
DaLi	0.0913	0.1040	0.0811	0.0801	0.0809	0.0642	**0.0618**
XiShan	0.1282	0.0899	0.0874	0.0884	0.0799	0.0699	**0.0684**
PanLong	0.0620	0.1036	0.0519	0.0494	**0.0488**	0.0552	0.0544
JingHong	0.0819	0.0866	0.0941	0.0936	0.0723	0.0652	**0.0496**
Average	0.0938	0.0884	0.0818	0.0784	0.0716	0.0649	**0.0590**

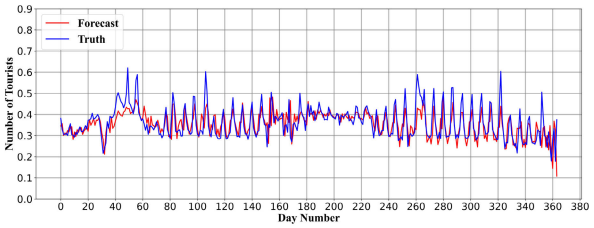
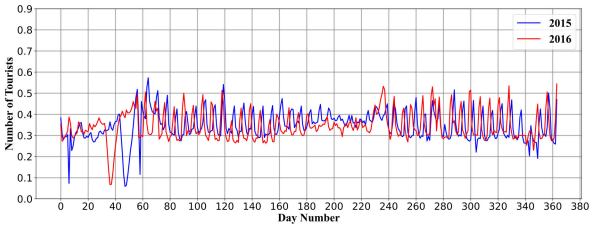
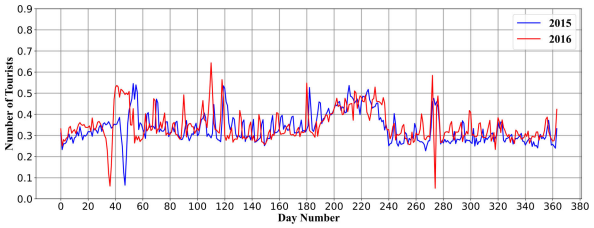


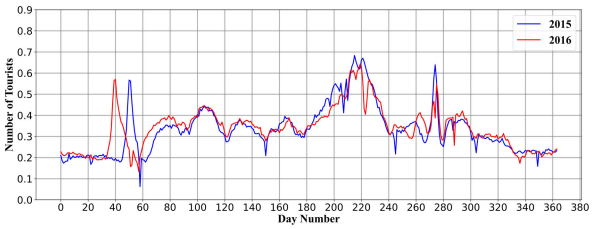
FIGURE 9. The predictive results of FC-C-LSTM for the PanLong city.



(a) PanLong



(b) DaLi

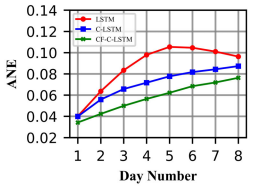


(c) GuCheng

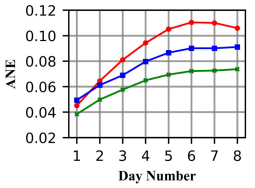
FIGURE 10. The number of tourists in 2015 and 2016 in PanLong, DaLi and GuanDu cities.

3) ANE OF EACH FRAME

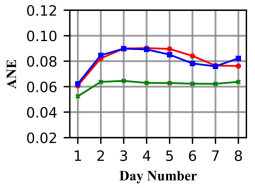
Our task is to predict the number of people in the next 8 days based on the previous 12 days data. Fig. 11 depicts the average numerical error with respect to the number of days predicted on the six datasets, respectively. The red, blue, and green lines represent the results of the LSTM, C-LSTM, and



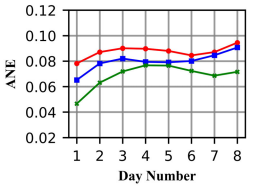
(a) GuCheng



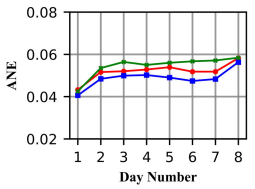
(b) GuanDu



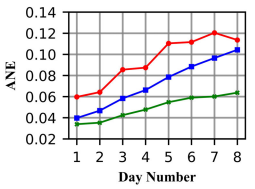
(c) DaLi



(d) XiShan



(e) PanLong



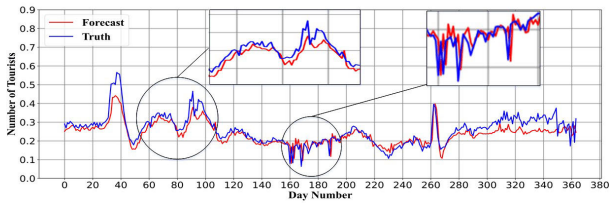
(f) JingHong

FIGURE 11. Variations of the error of each frame. The x-axis represents the day number to be predicted, ranking from 1 to 8, and the y-axis represents the average number error.

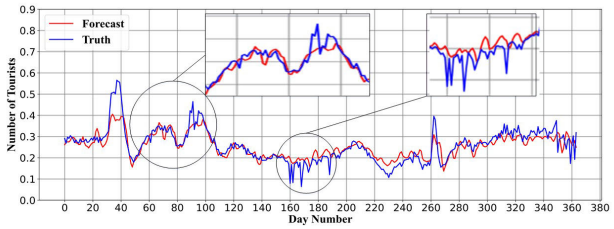
FC-C-LSTM models. In general, the ANEs increase with the number of prediction days in Fig. 11, and our proposed C-LSTM and FC-C-LSTM models achieve the better performance than that of LSTM. Furthermore, under the guidance of cascade features, the FC-C-LSTM model is better than the C-LSTM model in most cases, that is, the green line is at the bottom.

In Fig. 11(a), 11(b) and 11(f), the ANE increases when day number increases, it is congenial with reason and common sense. However, in Fig. 11(c), 11(d) and 11(e), the change is dramatic, the ANEs increase at the beginning and then plunges.

In order to find out the reason, we refer to the prediction results in Fig. 9 and Fig. 12. We found that the forecast results perform worse than usual during holidays. This may result in the increase of ANE in the intermediate dates. Another possibility is that the data from the real world obtains outliers in some values, which leads to a large deviation in the prediction results. Since there is no periodicity in the data of PanLlong



(a) C-LSTM



(b) FC-C-LSTM

FIGURE 12. The prediction results of JingHong city.

city, as shown in Fig. 9, the periodicity introduced by the feature cascaded module affects the accuracy of prediction. In Fig. 11(e), the proposed C-LSTM model achieves the best performance, that is, the blue line is at the bottom.

F. QUALITATIVE ANALYSIS

In Fig. 12, we visualize the number of tourists predicted by the C-LSTM model and the FC-C-LSTM model in a period.

The experimental results present that the C-LSTM model without feature cascaded module predicts the results several days later than the actual value. Specifically, there is a one to three day delay in the prediction results of the number of days from 50 to 110 in Fig. 12(a). In Fig. 12(b), we can observe that the FC-C-LSTM model with feature cascade module has greatly improved the delay phenomenon in the same time interval.

In Fig. 12(a), there are also some data outliers, which abnormally rise or fall in 80 to 100 days interval. FC-C-LSTM can make stable predictions by ignoring the abnormal rise or fall of data. In addition, the C-LSTM model has an over-fitting in the interval between 160-190 days. But for FC-C-LSTM, this over-fitting problem has been solved. The FC-C-LSTM model not only has higher numerical accuracy, but also solves the time delay in the prediction results, as shown in Fig. 12.

In conclusion, our proposed FC-C-LSTM model is able to understand the temporal dependence of the number of the tourists and reduce the impact on the prediction process of the outliers on the dataset. In addition, with the guidance of the information from the previous years, our model solves the time lag problem in the prediction task.

VI. CONCLUSION

In this paper, we propose a novel feature-cascaded framework for LSTM using a correntropy as cost function to solve the problem of tourists number prediction. In our work, we extract the feature information from the previous span and corresponding period and integrate them as the cascaded feature which can express the temporal dependence of the number of tourist and impact of external factors. Meanwhile, the FC-C-LSTM using a correntropy as cost function can reduce the impact of outliers on the prediction results, so that we do not need to manually remove outliers. Compared with ARIMA, ELM and initial LSTM, higher accuracy of our framework (FC-C-LSTM) is verified on six datasets. And, the proposed model is a practical application in the forecast modeling analysis the number of tourists, it can accurately predict the number of visitors to a city every day, provide the basis for the government to control traffic and avoid some accidents.

Due to insufficient data collection, the FC-C-LSTM does not take into account the influence of some factors, such as weather, specific policies and emergencies *et al.* Therefore, we will improve model by introducing the weather, news, specific policies and other external factors, such as multimodal hybrid model for the number of tourist prediction which has higher accuracy and stronger generalization.

REFERENCES

[1] J. Franklin, "A time series model for the stochastic process associated with acoustic measurement systems," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Hartford, CT, USA, May 1977, pp. 303–306, doi: 10.1109/ICASSP.1977.1170326.

[2] G. Brown, "Application of the integral equation method of smoothing to random surface scattering," *IEEE Trans. Antennas Propag.*, vol. AP-32, no. 12, pp. 1308–1312, Dec. 1984, doi: 10.1109/TAP.1984.1143253.

[3] C. N. Babu and B. E. Reddy, "Predictive data mining on average global temperature using variants of ARIMA models," *IEEE Int. Conf. Adv. In Eng., Sci. And Manage. (ICAESM)*, Nagapattinam, India, 2012, pp. 256–260.

[4] J. A. Putra, F. Basbeth, and S. Bukhori, "Sugar production forecasting system in PTPN XI semboro jember using autoregressive integrated moving average (ARIMA) method," in *Proc. 6th Int. Conf. Electr. Eng., Comput. Sci. Informat. (EECSI)*, Bandung, Indonesia, Sep. 2019, pp. 448–453, doi: 10.23919/EECSI48112.2019.8977010.

[5] M. Sikalubya, S. Xu, W. Yu, and P. Moonga, "Study on forecasting soybean production: An application of ARIMA model," in *Proc. Int. Conf. Intell. Comput., Autom. Syst. (ICICAS)*, Chongqing, China, Dec. 2019, pp. 447–452, doi: 10.1109/ICICAS48597.2019.00100.

[6] H.-Y. Zhu, "N days average volume based ARIMA forecasting model for Shanghai metro passenger flow," in *Proc. Int. Conf. Artif. Intell. Educ. (ICAIE)*, Hangzhou, China, Oct. 2010, pp. 631–635, doi: 10.1109/ICAIE.2010.5641088.

[7] A. Kharista, A. E. Permanasari, and I. Hidayah, "The performance of GM (1,1) and ARIMA for forecasting of foreign tourists visit to Indonesia," in *Proc. Int. Seminar Intell. Technol. Appl. (ISITIA)*, Surabaya, Indonesia, May 2015, pp. 33–38, doi: 10.1109/ISITIA.2015.7219949.

[8] S. Hansun, M. B. Kristanda, C. R. Indrati, and T. Aryono, "Forecasting domestic tourist arrivals to bali: H-WEMA approach," in *Proc. 5th Int. Conf. New Media Stud. (CONMEDIA)*, Bali, Indonesia, Oct. 2019, pp. 121–124, doi: 10.1109/CONMEDIA46929.2019.8981825.

[9] X. Wang, "The short-term passenger flow forecasting of urban rail transit based on Holt-Winters' seasonal method," in *Proc. 4th Int. Conf. Electromech. Control Technol. Transp. (ICECTT)*, Guilin, China, Apr. 2019, pp. 265–268, doi: 10.1109/ICECTT.2019.00067.

[10] A. A. Rizal and S. Hartati, "Recurrent neural network with extended Kalman filter for prediction of the number of tourist arrival in Lombok," in *Proc. Int. Conf. Informat. Comput. (ICIC)*, Mataram, Indonesia, 2016, pp. 180–185, doi: 10.1109/IAC.2016.7905712.

[11] Y. Hu, W. Yang, S. Xia, W.-H. Cheng, and J. Liu, "Enhanced intra prediction with recurrent neural network in video coding," in *Proc. Data Compress. Conf.*, Snowbird, UT, USA, Mar. 2018, p. 413, doi: 10.1109/DCC.2018.00066.

[12] R. Lotfidereshgi and P. Gournay, "Speech prediction using an adaptive recurrent neural network with application to packet loss concealment," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Calgary, AB, Canada, Apr. 2018, pp. 5394–5398, doi: 10.1109/ICASSP.2018.8462185.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[14] Q. Wang, R.-Q. Peng, J.-Q. Wang, Z. Li, and H.-B. Qu, "NEWLSTM: An optimized long short-term memory language model for sequence prediction," *IEEE Access*, vol. 8, pp. 65395–65401, 2020, doi: 10.1109/ACCESS.2020.2985418.

[15] P. Poonia and V. K. Jain, "Short-term traffic flow prediction: Using LSTM," in *Proc. Int. Conf. Emerg. Trends Commun., Control Comput. (ICONC)*, Lakshmangarh, India, Feb. 2020, pp. 1–4, doi: 10.1109/ICONC345789.2020.9117329.

[16] W. Yu, W. Zhifei, W. Hongye, Z. Junfeng, and F. Ruilong, "Prediction of passenger flow based on CNN-LSTM hybrid model," in *Proc. 12th Int. Symp. Comput. Intell. Design (ISCID)*, Hangzhou, China, Dec. 2019, pp. 132–135, doi: 10.1109/ISCID.2019.10113.

[17] D. Chen, C. Xiong, and M. Zhong, "Improved LSTM based on attention mechanism for short-term traffic flow prediction," in *Proc. 10th Int. Conf. Inf. Sci. Technol. (ICIST)*, London, U.K., Sep. 2020, pp. 71–76, doi: 10.1109/ICIST49303.2020.9202045.

[18] K. Peng, W. Bai, and L. Wu, "Passenger flow forecast of railway station based on improved LSTM," in *Proc. 2nd Int. Conf. Adv. Comput. Technol., Inf. Sci. Commun. (CTISC)*, Suzhou, China, Mar. 2020, pp. 166–170, doi: 10.1109/CTISC49998.2020.00033.

[19] Z. Abbas, A. Al-Shishtawy, S. Girdzijauskas, and V. Vlassov, "Short-term traffic prediction using long short-term memory neural networks," in *Proc. IEEE Int. Congr. Big Data (BigData Congr.)*, San Francisco, CA, USA, Jul. 2018, pp. 57–65, doi: 10.1109/BigDataCongress.2018.00015.

[20] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *Proc. 31st Youth Acad. Annu. Conf. Chin. Assoc. Autom. (YAC)*, Wuhan, China, Nov. 2016, pp. 324–328, doi: 10.1109/YAC.2016.7804912.

[21] S. Dai, L. Li, and Z. Li, "Modeling vehicle interactions via modified LSTM models for trajectory prediction," *IEEE Access*, vol. 7, pp. 38287–38296, 2019, doi: 10.1109/ACCESS.2019.2907000.

[22] W. Yao, P. Huang, and Z. Jia, "Multidimensional LSTM networks to predict wind speed," in *Proc. 37th Chin. Control Conf. (CCC)*, Wuhan, China, Jul. 2018, pp. 7493–7497, doi: 10.23919/ChiCC.2018.8484017.

[23] Q. Ye, X. Yang, C. Chen, and J. Wang, "River water quality parameters prediction method based on LSTM-RNN model," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Nanchang, China, Jun. 2019, pp. 3024–3028, doi: 10.1109/CCDC.2019.8832885.

[24] Z. Zhang, R. Yang, and Y. Fang, "LSTM network based on on antlion optimization and its application in flight trajectory prediction," in *Proc. 2nd IEEE Adv. Inf. Manage., Communicates, Electron. Autom. Control Conf. (IMCEC)*, Xi'an, China, May 2018, pp. 1658–1662, doi: 10.1109/IMCEC.2018.8469476.

[25] X. Song, J. Huang, and D. Song, "Air quality prediction based on LSTM-Kalman model," in *Proc. IEEE 8th Joint Int. Inf. Technol. Artif. Intell. Conf. (ITAIC)*, Chongqing, China, May 2019, pp. 695–699, doi: 10.1109/ITAIC.2019.8785751.

[26] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long-and short-term temporal patterns with deep neural networks," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, New York, NY, USA, Jun. 2018, pp. 95–104.

[27] X. Hu, G. Dai, Y. Ge, Z. Ning, and Y. Liu, "A simplified deep residual network for citywide crowd flows prediction," in *Proc. 14th Int. Conf. Semantics, Knowl. Grids (SKG)*, Guangzhou, China, Sep. 2018, pp. 60–67, doi: 10.1109/SKG.2018.00016.

[28] Y.-H. Chang and H.-C. Jang, "Traffic flow forecast for traffic with forecastable sporadic events," in *Proc. 12th Int. Conf. Ubi-Media Comput. (Ubi-Media)*, Bali, Indonesia, Aug. 2019, pp. 145–150, doi: 10.1109/Ubi-Media.2019.00036.

[29] M. Xu, Q. Wang, and Q. Lin, "Hybrid holiday traffic predictions in cellular networks," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS)*, Taipei, Taiwan, Apr. 2018, pp. 1–6, doi: 10.1109/NOMS.2018.8406291.

[30] L. Bai, "Research on the computer algorithm application in urban rail transit holiday passenger flow prediction," in *Proc. Int. Conf. Netw. Inf. Syst. Comput. (ICNISC)*, Wuhan, China, Apr. 2016, pp. 233–236, doi: 10.1109/ICNISC.2016.058.

[31] G. Zhou and J. Tang, "Forecast of urban rail transit passenger flow in holidays based on support vector machine model," in *Proc. 5th Int. Conf. Electromechan. Control Technol. Transp. (ICECTT)*, Nanchang, China, May 2020, pp. 585–589, doi: 10.1109/ICECTT50890.2020.00133.

[32] Y. Zhou, H. Deng, Y. Zeng, H. Chen, and Z. Lian, "Holiday travel pattern forecast based on machine learning algorithm," in *Proc. IEEE 4th Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, Chengdu, China, Dec. 2019, pp. 2288–2291, doi: 10.1109/IAEAC47372.2019.8997605.

[33] V. Podgorelec, M. Hericko, and I. Rozman, "Improving mining of medical data by outliers prediction," in *Proc. 18th IEEE Symp. Comput.-Based Med. Syst. (CBMS)*, Dublin, Republic of Ireland, 2005, pp. 91–96, doi: 10.1109/CBMS.2005.68.

[34] S. Prykhodko, L. Makarova, K. Prykhodko, and A. Pukhalevych, "Application of transformed prediction ellipsoids for outlier detection in multivariate non-Gaussian data," in *Proc. IEEE 15th Int. Conf. Adv. Trends Radioelectronics, Telecommun. Comput. Eng. (TCSET)*, Lviv-Slavske, Ukraine, Feb. 2020, pp. 359–362, doi: 10.1109/TCSET49122.2020.235454.

[35] T. Liu, H. Gao, and J. Wu, "Review of outlier detection algorithms based on grain storage temperature data," in *Proc. IEEE Int. Conf. Artif. Intell. Comput. Appl. (ICAICA)*, Dalian, China, Jun. 2020, pp. 1045–1048, doi: 10.1109/ICAICA50127.2020.9182588.

[36] F. Angiulli, S. Basta, and C. Pizzuti, "Distance-based detection and prediction of outliers," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 2, pp. 145–160, Feb. 2006, doi: 10.1109/TKDE.2006.29.

[37] L. Zheng, W. Hu, and Y. Min, "Raw wind data preprocessing: A data-mining approach," *IEEE Trans. Sustain. Energy*, vol. 6, no. 1, pp. 11–19, Jan. 2015, doi: 10.1109/TSTE.2014.2355837.

[38] Z. Wang, X. Huang, Y. Song, and J. Xiao, "An outlier detection algorithm based on the degree of sharpness and its applications on traffic big data preprocessing," in *Proc. IEEE 2nd Int. Conf. Big Data Anal. (ICBDA)*, Beijing, China, Mar. 2017, pp. 478–482, doi: 10.1109/ICBDA.2017.8078867.

[39] P. Nair and I. Kashyap, "Hybrid pre-processing technique for handling imbalanced data and detecting outliers for KNN classifier," in *Proc. Int. Conf. Mach. Learn., Big Data, Cloud Parallel Comput. (COMITCon)*, Faridabad, India, Feb. 2019, pp. 460–464, doi: 10.1109/COMITCon.2019.8862250.

[40] C. Sun and P. Guo, "Data preprocessing of wind turbine based on least squares support vector machine and neighbor model," in *Proc. 29th Chin. Control Decis. Conf. (CCDC)*, Chongqing, China, May 2017, pp. 1441–1446, doi: 10.1109/CCDC.2017.7978744.

[41] W. Liu, P. P. Pokharel, and J. C. Principe, "Correntropy: A localized similarity measure," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Vancouver, BC, Canada, Jul. 2006, pp. 4919–4924, doi: 10.1109/IJCNN.2006.247192.

[42] B. Chen and J. C. Principe, "Maximum correntropy estimation is a smoothed MAP estimation," *IEEE Signal Process. Lett.*, vol. 19, no. 8, pp. 491–494, Jun. 2012.

[43] B. Chen, X. Liu, H. Zhao, and J. C. Principe, "Maximum correntropy Kalman filter," *Automatica*, vol. 76, pp. 70–77, Feb. 2017.

[44] A. R. Heravi and G. A. Hodtani, "A new correntropy-based conjugate gradient backpropagation algorithm for improving training in neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6252–6263, Dec. 2018, doi: 10.1109/TNNLS.2018.2827778.

[45] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 7132–7141, doi: 10.1109/CVPR.2018.00745.

[46] Y. Zhou, H. Chen, J. Li, Y. Wu, J. Wu, and L. Chen, "ST-attn: Spatial-temporal attention mechanism for multi-step citywide crowd flow prediction," in *Proc. Int. Conf. Data Mining Workshops (ICDMW)*, Beijing, China, Nov. 2019, pp. 609–614, doi: 10.1109/ICDMW.2019.00092.

[47] S. Gong, E.-B. Bourennane, and J. Gao, "Multi-feature counting of dense crowd image based on multi-column convolutional neural network," in *Proc. 5th Int. Conf. Comput. Commun. Syst. (ICCCS)*, Shanghai, China, May 2020, pp. 215–219, doi: 10.1109/ICCCS49078.2020.9118564.

[48] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*. Hoboken, NJ, USA: Wiley, 2015.

[49] N. Zheng, S. Du, J. Wang, H. Zhang, W. Cui, Z. Kang, T. Yang, B. Lou, Y. Chi, H. Long, M. Ma, Q. Yuan, S. Zhang, D. Zhang, F. Ye, and J. Xin, "Predicting COVID-19 in China using hybrid AI model," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 2891–2904, Jul. 2020, doi: 10.1109/TCYB.2020.2990162.

[50] S. Zhang, S. Liu, and M. Liu, "Natural language inference using LSTM model with sentence fusion," in *Proc. 36th Chin. Control Conf. (CCC)*, Dalian, China, Jul. 2017, pp. 11081–11085, doi: 10.23919/ChiCC.2017.8029126.
[51] N. R. Gafurov, I. A. Bessmertny, A. V. Platonov, E. A. Poleshchuk, and A. V. Vasiliev, "Named entity recognition through bidirectional LSTM in natural language texts obtained through audio interfaces," in *Proc. IEEE 12th Int. Conf. Appl. Inf. Commun. Technol. (AICT)*, Almaty, Kazakhstan, Oct. 2018, pp. 1–4, doi: 10.1109/ICAICT.2018.8747163.
[52] S. A. Selouani and M. S. Yacoub, "Long short-term memory neural networks for artificial dialogue generation," in *Proc. IEEE 42nd Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Tokyo, Japan, Jul. 2018, pp. 761–768, doi: 10.1109/COMPSAC.2018.00113.
[53] S. R. Oota, V. Rowtula, M. Gupta, and R. S. Bapi, "StepEncog: A convolutional LSTM autoencoder for near-perfect fMRI encoding," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Budapest, Hungary, Jul. 2019, pp. 1–8, doi: 10.1109/IJCNN.2019.8852339.
[54] B. Premchand, K. K. Toe, C. Wang, S. Shaikh, C. Libedinsky, K. K. Ang, and R. Q. So, "Decoding movement direction from cortical microelectrode recordings using an LSTM-based neural network," in *Proc. 42nd Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Montreal, QC, Canada, Jul. 2020, pp. 3007–3010, doi: 10.1109/EMBC44109.2020.9175593.
[55] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, Dec. 2006.
[56] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 513–529, Apr. 2012.
[57] Z. Q. Geng, L. Qin, and Y. M. Han, and Q. X. Zhu, "Energy saving and prediction modeling of petrochemical industries: A novel ELM based on FAHP," *Energy*, vol. 122, pp. 350–362, Mar. 2017.
[58] Y. Xu, J. Feng, and Q. X. Zhu, "Research and application of extension theory-based radial basis function neural network," *Control Decis.*, vol. 26, no. 11, 2011, Art. no. 1721e5.
[59] W. Lv, Z. Mao, and M. Jia, "ELM based LF temperature prediction model and its online sequential learning," in *Proc. 24th Chin. Control Decis. Conf. (CCDC)*, Taiyuan, China, May 2012, pp. 2362–2365, doi: 10.1109/CCDC.2012.6244378.
[60] H. Tian and B. Meng, "A new modeling method based on bagging ELM for day-ahead electricity price prediction," in *Proc. IEEE 5th Int. Conf. Bio-Inspired Comput., Theories Appl. (BIC-TA)*, Changsha, China, Sep. 2010, pp. 1076–1079, doi: 10.1109/BICTA.2010.5645111.
[61] Y.-M. Xing, X.-J. Ban, and R. Liu, "A short-term traffic flow prediction method based on kernel extreme learning machine," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Shanghai, China, Jan. 2018, pp. 533–536, doi: 10.1109/BigComp.2018.00089.
[62] Z. Ma, G. Luo, and D. Huang, "Short term traffic flow prediction based on on-line sequential extreme learning machine," in *Proc. 8th Int. Conf. Adv. Comput. Intell. (ICACI)*, Chiang Mai, Thailand, Feb. 2016, pp. 143–149, doi: 10.1109/ICACI.2016.7449818.
[63] D. Wang, J. Xiong, Z. Xiao, and X. Li, "Short-term traffic flow prediction based on ensemble real-time sequential extreme learning machine under non-stationary condition," in *Proc. IEEE 83rd Veh. Technol. Conf. (VTC Spring)*, Nanjing, China, May 2016, pp. 1–5, doi: 10.1109/VTCSpring.2016.7504474.
[64] A. Mosavi, S. Ardabili, and A. R. Varkonyi-Kóczy, "List of deep learning models," in *Proc. Int. Conf. Global Res. Educ.*, 2020, pp. 202–214, doi: 10.1007/978-3-030-36841-8_20.
[65] S. Nosratabadi, A. Mosavi, P. Duan, P. Ghamisi, F. Filip, S. Band, U. Reuter, J. Gama, and A. Gandomi, "Data science in economics: Comprehensive review of advanced machine learning and deep learning methods," *Mathematics*, vol. 8, no. 10, p. 1799, Oct. 2020, doi: 10.3390/math8101799.
[66] A. Mosavi, Y. Faghan, P. Ghamisi, P. Duan, S. F. Ardabili, E. Salwana, and S. S. Band, "Comprehensive review of deep reinforcement learning methods and applications in economics," *Mathematics*, vol. 8, no. 10, p. 1640, Sep. 2020, doi: 10.3390/math8101640.

YUEHAI CHEN (Graduate Student Member, IEEE) received the B.S. degree in automation from Xi'an Jiaotong University, Xi'an, Shaanxi, China, in 2020, where he is currently pursuing the master's degree in control science and engineering.

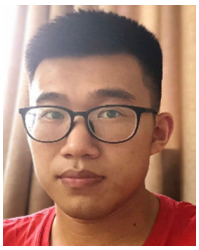
His research interests include machine learning, artificial intelligence, motion detection and tracking, and their applications to intelligent systems.

JING YANG (Member, IEEE) received the B.S. and M.S. degrees in control science and engineering, and the Ph.D. degree in pattern recognition and intelligent systems from Xi'an Jiaotong University, China, in 1999, 2002, and 2010, respectively.

From 1999 to 2003, she was a Research Assistant with the Institute of Automation, Xi'an Jiaotong University. Since 2003, she has been an Assistant Professor with the Department of Automation Science and Technology, Xi'an Jiaotong University. Since 2004, she has been a member of the Intelligent Vehicles Team, Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. Her research interests include machine learning, reinforcement learning, and information theory and their applications to intelligent systems, such as autonomous vehicles.

KUN ZHANG (Graduate Student Member, IEEE) received the B.S. degree in automation from Xi'an Jiaotong University, Xi'an, Shaanxi, China, in 2020, where he is currently pursuing the master's degree in control science and engineering.

His research interests include machine learning, object detection, action recognition and detection, and their applications to intelligent systems.



YI XU (Student Member, IEEE) received the B.S. degree in automation from Xi'an Jiaotong University, Xi'an, Shaanxi, China, in 2017, where he is currently pursuing the master's degree in control science and engineering.

His research interests include machine learning, artificial intelligence, model-based reinforcement learning, and their applications to intelligent systems.

YUEWEN LIU (Member, IEEE) received the Ph.D. degree in information systems from the City University of Hong Kong, in 2009, and the Ph.D. degree in management science and engineering from the University of Science and Technology of China, in 2010.

He is currently an Associate Professor with the School of Management, Xi'an Jiaotong University, China. His current research interests include information systems, e-commerce, and business statistics.