

# Research on GRU Neural Network Satellite Traffic Prediction Based on Transfer Learning

Ning Li<sup>1</sup> · Lang Hu<sup>1</sup> · Zhong-Liang Deng<sup>1</sup> · Tong Su<sup>1</sup> · Jiang-Wang Liu<sup>1</sup>

Accepted: 23 December 2020 / Published online: 3 January 2021 © The Author(s) 2021

## Abstract

In this paper, we propose a Gated Recurrent Unit(GRU) neural network traffic prediction algorithm based on transfer learning. By introducing two gate structures, such as reset gate and update gate, the GRU neural network avoids the problems of gradient disappearance and gradient explosion. It can effectively represent the characteristics of long correlation traffic, and can realize the expression of nonlinear, self-similar, long correlation and other characteristics of satellite network traffic. The paper combines the transfer learning method to solve the problem of insufficient online traffic data and uses the particle filter online training algorithm to reduce the training time complexity and achieve accurate prediction of satellite network traffic. The simulation results show that the average relative error of the proposed traffic prediction algorithm is 35.80% and 8.13% lower than FARIMA and SVR, and the particle filter algorithm is 40% faster than the gradient descent algorithm.

Keywords Low-earth orbit satellite network  $\cdot$  Traffic prediction  $\cdot$  GRU neural network  $\cdot$  Transfer learning  $\cdot$  Particle filter

# **1** Introduction

The satellite network traffic is affected by the periodic changes of the satellite network topology, the frequent switching of the satellite inter-satellite links, and the dynamic change of the inter-satellite link on-off relationship with time. The load of the satellite network traffic is adjacent to the geographical location of the satellite. Satellite network traffic has more complex and nonlinear characteristics [1]. To prevent network congestion and improve the utilization of network resources, reasonable network traffic management is especially important. The prediction of network traffic can grasp the changing characteristics and trends of network traffic in advance, to specify a reasonable and effective traffic management strategy to meet the requirements of users for quality of service (QoS) [2]. Therefore, it is of great practical significance to establish a high-precision traffic prediction model for satellite network.

⊠ Ning Li Inmmdsy@sina.com

<sup>&</sup>lt;sup>1</sup> School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing, China

The traditional autoregressive model (AR), moving average model (MA), autoregresssive moving average model (ARMA), and autoregressive Integrated moving average model (ARIMA) can only express short correlation traffic characteristics [3]. With the continuous introduction of technologies such as neural networks and support vector machines, prediction models based on machine learning algorithms have emerged, such as artificial neural networks, least squares support vector machines (LSSVM), extreme learning machines (ELM), etc. [4]. The problem with the above algorithm is the lack of consideration of the temporal correlation of time series data, the limited prediction accuracy, and the satellite network traffic cannot be predicted effectively [5]. The recurrent neural network (RNN) is a deep neural network that introduces cyclic feedback [6]. Long short-term memory (LSTM) network is a special model of RNN. It can learn the long-term dependence between time series data and can effectively solve the gradient disappearance and gradient explosion problem in traditional RNN training process. However, the LSTM network introduces three types of gate structures and state space, resulting in greater time complexity [7]. To alleviate computing resources of satellite, reduce computational complexity, this paper proposes a GRU neural network, which simplifies the three gate structures in LSTM into two kinds of gate structures, the update gate and the reset gate, and combines the cell state and output into one state [8]. In this simplified way, it not only retains the LSTM's ability to store long-term state, but also greatly reduce the computational complexity. GRU can greatly improve the training efficiency of the model and retain the effect like LSTM [9].

To further reduce the consumption of satellite computing resources, and to solve the problem of insufficient real-time data on the star and sufficient historical data, a transfer learning method is introduced. By learning accumulated knowledge from data from similar domains, the transfer learning approach facilitates the formation of predictive models from data in the target domain. [10]. At the same time, in order to reduce the complexity of online update parameters on the satellite, we abandon the traditional Stochastic Gradient Descent (SGD)based training method and study the low- computational complexity of particle filtering (PF) online training. The method further determines the optimal parameters of the model, improves the accuracy of the model prediction, and reduces the training time of the model.

## 2 GRU Neural Network

The GRU neural network retains the ability to remember long-term states by using update gates and reset gate structures, and greatly reduces computational complexity [11]. The GRU neural network diagram is below (Fig. 1).

Use the formula to express:

$$r_t = \sigma \left( W_r \cdot \left[ h_{t-1}, x_t \right] \right) \tag{1}$$

$$z_t = \sigma \left( W_z \cdot \left[ h_{t-1}, x_t \right] \right) \tag{2}$$

$$\tilde{h}_{t} = \tanh\left(W_{\tilde{h}} \cdot \left[r_{t} \otimes h_{t-1}, x_{t}\right]\right)$$
(3)

$$h_t = (1 - z_t) \otimes h_{t-1} + z_t \otimes h_t \tag{4}$$



$$y_t = \sigma \left( W_o \cdot h_t \right) \tag{5}$$

The square brackets indicate that two vectors are connected, and  $\bigotimes$  is matrix elements multiplication.  $\sigma$  is a sigmoid function whose output is between 0 and 1, indicating the update and forgetting degree of information.  $r_t$  indicates the update gate, which is used to determine how much information is saved to the next moment at the previous moment. The larger the value of the update gate, the more information from the previous moment is retained.  $z_t$  indicates the reset gate, which is used to determine the status information of the previous moment. Among them, the parameters we need to learn training are  $W_r$ ,  $W_z$ ,  $W_{\tilde{h}}$  and  $W_o$ , the input of the output layer is  $y_t^i = W_o h$ , and the output is  $y_o^r = \sigma(y_t^i)$ .

# 3 Traffic Prediction Framework of GRU Neural Network Based on Transfer Learning

To solve the problem of satellite network traffic prediction, this paper proposes a GRU neural prediction framework of network traffic based on transfer learning. The framework is mainly composed of three parts: data processing module, model training module, and model transfer module. The data processing module is mainly responsible for pre-processing the data, converting the continuous flow data into discrete flow data to meet the input requirements of the model. The model building module is the core of the traffic prediction framework. This paper proposes a model tuning method such as batch normalization and dropout. A low complexity training method of particle filter model is proposed. The model transfer module is another important model. It transfers a training model with large number of offline traffic data into online model of satellite to avoid the problem of insufficient online traffic data. Finally, the GRU neural network traffic prediction is constructed.

### 3.1 Data Processing Module

The data processing module samples the flow data at a fixed time interval t to obtain input discrete flow data. Time window is used to convert discrete flow data into a supervised model input data format

In the manner of port sliding, discrete data is divided according to a fixed time window size m sliding window, and finally the traffic data is obtained as  $X = [x_1, x_2, x_3, ..., x_{m-1}, x_m]$ . Taking the data  $x_m$  of the last m time as the predicted target output Y of the model, that is, the label with the supervised data. After that, the supervised data sequence is divided into training set test sets according to a certain proportion, and finally the data set for model training test is obtained.

#### 3.2 Model Building Module

As the core of the traffic prediction framework, the model building module considers the timeliness of satellite network traffic data and limited satellite computing resources and designs a single-layer GRU network structure. This can not only ensure the prediction effect of the model, but also reduce the time for the model to optimize parameters. The overall model structure is a three-layer network model, the first layer is the input layer, and the number of neurons in the input layer is equal to the input traffic data dimension. The second layer is a hidden layer, and the number of neurons in the hidden layer is determined according to the experimental results. The third layer is the output layer, because the model finally predicts that the output is a single flow value, and the number of neurons in the output layer is set to 1.

Model training module: Model training refers to the optimization of the square loss function. The model training reduces the loss function value by constantly adjusting the weight matrix of the network. Usually, the gradient weight reduction method is used to optimize the model weight matrix. However, the gradient descent optimization process may suffer from over-fitting or falling into a local optimal solution. Section 4 details how the ion filter algorithm solves this problem.

Model tuning module: network structure tuning and network parameter tuning. Network structure tuning increases the model's generalization ability, reduces the training time of the model, reduces the possibility of model overfitting, and adds a Dropout layer before the hidden layer [12]. In order to solve the problem of inconsistent data distribution of each batch, batch normalization processing is performed before the activation function [13].

The Dropout layer is an indirect discard, and the output of each neuron is still calculated, and then selecting some neurons with a random probability and their outputs are set to zero. This random discarding method is simple in design, but still needs to calculate discarding neurons, which increases the computational cost of some satellites. This paper designs a pre-drop mode to set the output of neurons, which need to be set to zero. Although the problem of inconsistent data distribution in each batch is solved in the literature [13], some characteristics of the original data itself are lost. This paper introduces the learning parameters  $\beta$  and  $\gamma$  to overcome this problem.

Finally, the overall process of model training and tuning is described as follows: The training of the GRU neural network model can be described as the optimization of the network parameters  $\Theta$ , so that the difference between the predicted value and the true value of the model is reduced as much as possible:

$$\Theta = \arg\min_{\Theta} \frac{1}{N} \sum_{i=1}^{N} \ell oss(X_i, Y_i, \Theta)$$
(6)

$$loss = \frac{1}{N} \sum_{i=1}^{N} (Y_i - \hat{Y}_i)^2$$
(7)

Among them:  $\{X_1, Y_1, X_2, Y_2, \dots, X_N, Y_N\}$  are training data sets, and  $\Theta$  is the weight parameter in the GRU neural network. The loss function of the model is the mean square error, where  $\hat{Y}_i$  is the predicted output of the model. Add a Dropout layer before the hidden layer:

$$p_i^l \sim \text{Bernonlli}(p)$$
 (8)

$$\tilde{x}^l = p_i^l * x^l \tag{9}$$

where  $p_j^l$  is the Bernoulli probability, designed according to the characteristics of each batch of satellite flow data,  $\tilde{x}^l$  is randomly discarded based on input  $x^l$  with probability  $p_j^l$ , and the output of discarded neurons is set zero. Batch normalization means that we normalize a batch of data for a sample:

$$\hat{x}_i = \frac{x_i - \mu}{\sigma} \tag{10}$$

Where: $x = \{x_1, x_2, \dots, x_d\}$  is a batch of data,  $\mu$  is the expectation of the input flow data x, and  $\sigma$  is the standard deviation of the input flow data x. This batch standardization process can reduce the problem of data inconsistency, but directly inputting the standardized processing  $\hat{x}_i$  into the network ignores the feature distribution of the data itself. Therefore, this paper adds two learning parameters  $\gamma_i$  and  $\beta_i$  to maintain the feature distribution of the original data. After batch normalization, the data input into the activation function is:

$$y_i = \gamma_i \hat{x}_i + \beta_i \tag{11}$$

Where  $\gamma_i$  and  $\beta_i$  are parameters learned for a batch data model,  $\gamma_i$  and  $\beta_i$  parameters can retain part of the data features lost due to the normalization operation. Finally, the data distribution input to the activation function is more consistent and has the original data characteristics, and the convergence speed of the model can be improved.

#### 3.3 Model Transfer Module

The model transfer module is to realize the migration of the source data model to the destination data model and train the network to learn the neural network feature representation based on the historical large amount of traffic data, and then migrate the model to the online traffic data for training model.

Firstly, the offline flow data is transformed with the data to obtain the input data format of the model, and then the model building module is used to obtain the offline traffic prediction model. Based on the same processing and online traffic prediction model, online traffic data is added, and the model building module is used to retrain and obtain the online traffic prediction model.

## 4 Efficient Online Training Method Based on Particle Filter

The key of particle filter algorithm is to determine the state transition equation and observation equation of the system [14]. For the traffic prediction model of GRU neural network, the discrete time value is the number of iterations of the model, and the state of each system is the optimal solution of the model. Equations (1)-(5) as the state transition equation of the system, the mean square error loss function (7) as the system Observe the equation. The training process of GRU neural network model based on particle filter algorithm is as follows:

First, a discrete system dynamic model is established. The mathematical model is expressed as follows:

$$X_t = f(X_{t-1}, v_t) \tag{12}$$

$$Z_t = h(X_t, e_t) \tag{13}$$

Where  $X_t$  is the system state variable,  $Z_t$  is the true observation of the system,  $v_t$  is the system noise, and  $e_t$  is the measurement noise of the system.

Particle initialization: Each particle is considered that having equal weight if the system state is unknown. The initial particle set is generated by sampling with probability density  $p(x_0)$ :  $\left\{x_0^i, \frac{1}{N}; i = 1, 2, ..., N\right\}$ .

Initialize system state: Calculate the network output value y according to the parameters of the GRU neural network and Eqs. (1)-(5). Set the minimum threshold of the system,  $N_{thr}$ . Let the total number of particles in the particle filter algorithm be N, the total number of iterations is tf, set the end loss value l, and randomly generate N particles according to the prior probability density  $p(x_0)$ .

Importance sampling: When k = 1, 2, ..., N, to avoid particle degradation, it is necessary to copy some of the particles with higher weight and remove the particles with lower weight.

(1) First, randomly extract N particles from a probability distribution function:

$$x_k^i \sim q(x_k^i | x_{k-1}^i, y_k) = p(x_k^i | x_{k-1}^i), \quad i = 1, 2, \dots, N$$
 (14)

(2) Update the weight of the particles and normalize the particle weights:

$$w_{k}^{j} = w_{k-1}^{j} p(y_{k}^{i} | x_{k}^{i})$$
(15)

$$\tilde{w}_k^i = w_k^i / \sum_{i=1}^N w_k^i \tag{16}$$

According to the state transition equation  $p(x_k^i | x_{k-1}^i)$ , N particles are extracted from the initialized particle group. According to the observation Eq. (7) of the system, the matching value of all particles  $x_k^i$  is calculated, and the optimal particle and its corresponding optimal target y value are selected. The weight of the particle that does not satisfy the constraint is reset to zero. When the constraint is satisfied, according to the current observation  $y_k^i$  and the Eqs. (15) and (16) and normalized, updating the weight of the particle.

Resampling: Calculate the number of valid particles:

$$\hat{N}_{eff} = 1 / \sum_{i=1}^{N} \left( \tilde{w}_k^i \right)^2 \tag{17}$$

When  $\hat{N}_{eff} < N_{thr}$ , re-sampling is performed according to  $p(\tilde{x}_k^j = x_k^i) = \tilde{x}_k^j$ . We select the particles with larger weights for copying, and delete the smaller ones. A new set of generated particles is formed:  $\left\{x_k^j, \frac{1}{N}; j = 1, 2, ..., N\right\}$ .

State Estimation: Estimating System Status and Variance

$$\hat{x}_k = E(x_k | y_k) \approx \sum_{i=1}^N x_k^i \tilde{w}_k^i$$
(18)

$$P_{k} = \sum_{i=1}^{N} \tilde{w}_{k}^{i} (\hat{x}_{k} - x_{k}^{i}) (\hat{x}_{k} - x_{k}^{i})^{T}$$
(19)

Let k = k+1, continue the calculation, and judge whether the set loss value termination condition is satisfied.

The setting of  $\hat{N}_{eff}$  in the particle filter will directly determine the prediction accuracy of the model. According to the above steps, iterative iteration, so that the optimal state transition equation estimation can be obtained, and the final flow prediction value can be obtained.

# 5 Experiment Results and Analysis

## 5.1 Evaluation Indicators

In order to measure the prediction results of the model, three error analysis methods are used to verify the prediction results, namely mean absolute error (MAE), root mean square error (RMSE) and mean relative error(MRE), the formula is as follows [15]:

$$MAE = \frac{1}{N} \sum_{i=1}^{N} \left| \hat{Y}(i) - Y(i) \right|$$
(20)

$$RMSE = \sqrt{\frac{1}{N} \sum_{I=1}^{N} (\hat{Y}(i) - Y(i))^2}$$
(21)

$$MRE = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{\hat{Y}(i) - Y(i)}{Y(i)} \right| \times 100\%$$
(22)

Where  $\hat{Y}(i)$  is the true value,  $\hat{Y}(i)$  is the predicted value, and N is the total number of samples.

#### 5.2 Experimental Environment

The GRU neural traffic prediction model of network proposed in this paper is based on the Python2.7 programming language and the tensorflow 1.3 deep learning framework in the Ubuntu 16.04 operating system. The data source used in the experiments in this chapter is "BC-pOct89", which extracts 400,000 data volumes.

First, 400,000 data is divided into two parts: a large amount of data sets of 380,000 and 20,000 online real-time data sets. The 380,000 data was input into the network for training, and the offline pre-trained network model was obtained. The 20,000 data sets on the line are divided into training sets and test sets, of which the training set accounts for 4/5 and the test set accounts for 1/5. The training set is input into the pre-trained GRU neural network model parameters are adjusted to obtain the optimal network parameters. During the experiment, the algorithm was verified by the leave-one method, and the test results of the model were obtained. This chapter experiments and compares with the traditional FARIMA, SVR traffic prediction algorithms.

### 5.3 Analysis of Experimental Results

In order to reflect the superiority of the migration learning GRU neural network traffic prediction algorithm proposed in this paper, two comparative experiments are set up in this paper. Compared with the FARIMA-based traffic prediction algorithm, FARIMA can only process short-term time series, only considering the sequence. Statistical continuity before and after, and FARIMA does not have nonlinear fitting ability. Compared with the second experiment based on the SVM algorithm, SVM performs well in the classification and prediction of traditional data, but it does not apply to time series data, and cannot handle data of satellite network traffic well. Both the FARIMA algorithm and the SVM algorithm can only fit short-term traffic characteristics and cannot reflect the long-term and complex nonlinear characteristics of satellite traffic. The specific experimental results are shown in Fig. 2.



Fig. 2 Network traffic forecast results graph

To further compare the prediction effects of the GRU model with the SVR and FARIMA models, Table 1 calculates the MAE, MRE and RMSE for the three models. It shows that the table that the prediction results of the GRU model have better MAE, RMSE, and MAE values than the other two models, reflecting the superiority of the GRU model for predicting satellite network traffic.

## 5.4 Online Training Complexity Analysis

This section analyzes the complexity analysis of the particle-based filtering online training method and the traditional training method. The traditional training methods of comparison include stochastic gradient descent (SGD). Using Eq. (7) as the optimization function, the recursive formula for solving the weight of the SGD algorithm is:

$$w_{t+1} = w_t - \mu_t \nabla_{w_t} l(y_t, \hat{y}_t) = w_t + 2u_t (y_t - \hat{y}_t) \Lambda_t^o h(c_t)$$
(23)

Where  $u_t$  represents the learning rate and is a value greater than 0 and less than 1.  $\Lambda_t^0$  is the diagonal matrix of the output. The algorithm complexity of SGD is  $O(m^4 + m^2p^2)$ , where p is the input space dimension and m is the output space dimension. The complexity of the SGD algorithm is related to the input space dimension and the output space dimension.

According to the previous section, the complexity of the PF algorithm is  $O(N(m^2 + mp))$ , where N is the number of particles.

The particle training online training algorithm has the lowest complexity, although it is related to the number of particles N, but it is usually much smaller than the input space dimension p and the output space dimension m, and its algorithm complexity is lower than that of the random gradient.

In order to verify the low complexity and convergence efficiency of the particle filter algorithm compared with the random gradient descent algorithm, the number of iterations is set to be the same, and it is necessary to observe how much data RMSE needs to be stable when training on the same data set. The delay result is shown in Fig. 3.

We can see from the experimental results that the initial relative error of PF-GRU is lower than that of SGD-GRU on the average relative error MRE index, and its convergence speed is fast. After 450 sets of training data, it can converge and optimize. The error value and the relative error of the SGD-GRU after 750 sets of data is required to stabilize. The particle filter algorithm has a faster convergence rate than the random gradient descent algorithm, and the training required the amount of data is less, and the particle filtering algorithm combined with the previous analysis has lower complexity. Therefore, the particle filter algorithm can effectively reduce the computing and storage resources of the satellite.

Table 1 Comparison of prediction results and errors of different models	Model	MAE	RMSE	MRE (%)
	GRU	17.47	26.08	21.05
	FARIMA	33.53	42.20	56.85
	SVM	22.56	32.25	29.18



Fig. 3 Error convergence curve of different training methods on the training data set

# 6 Conclusion

This paper analyzes the characteristics of data of satellite network traffic. We proposes a prediction algorithm for GRU neural network traffic based on migration learning. In this paper, the construction process of the GRU neural network model and the model setting method are described in detail. The algorithm flow of the online training update method based on particle filter is given. What's more, we adopt the transfer learning method to avoid the problem of insufficient online traffic data and reduce the consumption of satellite computing resources. The simulation results show that compared with FARIMA algorithm and SVM algorithm, the proposed algorithm has superior prediction accuracy. We verify that the particle update based online update method has low complexity and fast convergence speed. In short, the proposed traffic prediction algorithm has higher traffic prediction accuracy, lower computational complexity, faster convergence speed, and can effectively reduce satellite computing storage resources. It is a superior prediction algorithm for predicting satellite traffic.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

# References

- 1. Tang, F., Zhang, H., Fu, L., et al. (2019). Multipath Cooperative Routing with Efficient Acknowledgement for LEO Satellite Networks. *IEEE Transactions on Mobile Computing*, 18(1), 179–192.
- Chen, C., Yang, W., Gong, Z., et al. (2020). Analysis of the impact of large constellations on the space debris environment and countermeasures. *Aerospace China*, 21(2), 16–22.
- 3. Peilong, L., Yu, H., et al. (2018). Hybrid-traffic-detour based load balancing for onboard routing in LEO satellite networks. *China Communication*, *15*(06), 38–51.
- Moscholios, I. D., Vassilakis, V. G., Sarigiannidis, P. G., et al. (2018). An analytical framework in LEO mobile satellite systems servicing batched Poisson traffic. *IET Communications*, 12(1), 18–25.
- Di, B., Zhang, H., Song, L., et al. (2018). Ultra-dense LEO: integrating terrestrial-satellite networks into 5G and Beyond for Data Offloading. *IEEE Transactions on Wireless Communications*, 18(1), 47–62.
- Ozdemir, E., Topcu, A. E., & Ozdemir, M. K. (2018). A hybrid HMM model for travel path inference with sparse GPS samples. *Transportation*, 45(2), 1–14.
- Shin, D. K., Lee, D. Y., Kim, K. C., et al. (2016). Artificial neural network prediction model for geosynchronous electron fluxes: dependence on satellite position and particle energy. *Space Weather-the International Journal of Research & Applications*, 14(4), 313–321.
- 8. Wu, Y., & Tan, L. (2018). Reducing model complexity for DNN based large-scale audio classification.
- Wang, D., Fan, J., Fu, H., et al. (2018). Research on optimization of big data construction engineering quality management based on RNN-LSTM. *Complexity*, 2018, 1–16.
- Yixing, W., Meiqin, L., Zhejing, B., et al. (2018). Short-term load forecasting with multi-source data using gated recurrent unit neural networks. *Energies*, 11(5), 1138.
- 11. Lu, K., Yan, Z., Xin, W., et al. (2018). Short-term electricity load forecasting method based on multilayered self-normalizing GRU network. IEEE Conference on Energy Internet & Energy System Integration.
- 12. Li, C., Gupta, S., Rana, S., et al. (2018). High Dimensional Bayesian Optimization Using Dropout.
- 13. Li, X., Chen, S., Hu, X., et al. (2018). Understanding the Disharmony between Dropout and Batch Normalization by Variance Shift.
- 14. Tian, Y., Liu, Z., Ge, M., et al. (2019). Multi-dimensional particle filter-based estimation of inter-system phase biases for multi-GNSS real-time integer ambiguity resolution. Journal of Geodesy (2).
- Zhang, G., Yang, J., Wang, W., et al. (2019). Adaptive visual target tracking algorithm based on classified-patch kernel particle filter. *EURASIP Journal on Image and Video Processing*, 2019(1), 20.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Ning Li** is an associate professor from the Beijing University of Posts and Telecommunications. She received the master's degree in automatic control from Beijing Institute of Technology in 1994, and the Ph.D. degree in physical electronics from Beijing University of Posts and Telecommunications in 2003. Her research interests include architecture, routing and resource management in satellite networks. E-mail: lnmmdsy@sina.com.

825



Lang Hu received the B.S. degree in Electronic Engineering from the Beijing University of Chemical Technology, Beijing, China in 2016. He is currently working toward the M.S. degree in the Beijing University of Posts and Telecommunications, Beijing, China. His research interests include satellite network routing and artificial intelligence. E-mail: hlang8160@163.com.



Zhong-Liang Deng is a famous professor from the Beijing University of Posts and Telecommunications. He represents the top-of-the-future indoor navigation in China. He has gained 70 patents from the China Patent Office and contributed 221 articles indexed by SCI and EI. He also mainly took several National High Technology Research and Development Programs ("863" Program) of China. He has gained two National Science and Technology Progress Award of China. E-mail: hlang8160@163.com.



**Tong Su** received the B.S. degree in Electronic Engineering from the Beijing University of Posts and Telecommunications, Beijing, China in 2016. He is currently working toward the M.S. degree in the Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include channel coding and reinforcement learning. E-mail: s\_stone\_mail@163.com.



Jiang-wang Liu received the B.S degree in Electronic Information Science and Technology from Huaihua College, Hunan, China in 2014. He is currently working toward the M.S degree in Beijing University of posts and Telecommunications, Beijing, China. His research interests include Satellite Communication and Network routing. E-mail: 1054062832@qq.com.