# Mobility-Aware Proactive QoS Monitoring for Mobile Edge Computing

Ting Wei[1], Pengcheng Zhang[1*], Hai Dong[2], Huiying Jin[1], and Athman Bouguettaya[3]

[1] College of Computing and Information, Hohai University, Nanjing, 211100, China
458279713@qq.com; pchzhang@hhu.edu.cn;367046895@qq.com
[2] School of Computing Technologies, RMIT University, Melbourne, Australia
hai.dong@rmit.edu.au
[3] School of Computer Science, The University of Sydney, NSW, Australia
athman.bouguettaya@sydney.edu.au

**Abstract.** This article presents a novel probabilistic QoS (Quality of Service) monitoring approach called LSTM-BSPM (DonLSTM-Den based BayeSian Runtime Proactive Monitoring), which is based on the DouLSTM-Den model and Gaussian Hidden Bayesian Classifier for mobile edge environments. A DouLSTM-Den model is designed to predict a user's trajectory in mobile edge environments. The predicted trajectory is leveraged to obtain the mobility-aware QoS and capture its spatio-temporal dependency. Next, a parent attribute is constructed for each QoS attribute to reduce the influence of dependence between QoS attributes on monitoring accuracy. A Gaussian hidden Bayes classifier is trained for each edge server to proactively monitor the user's mobility-aware QoS. We conduct a set of experiments respectively upon a public data set and a real-world data set demonstrate the feasibility and effectiveness of the proposed approach.

**Keywords:** Mobile/Multi-Access edge computing, Quality of Service, Monitoring, Bayesian classifier, LSTM model.

## 1 Introduction

Mobile (or Multi-Access) edge computing is a new distributed computing paradigm that transfers the computing power from cloud data centers to the edge of a network [1]. Mobile edge services refer to the services provisioned in mobile edge environments [2]. Users' requirements on mobile edge services have gradually shifted from functional requirements to non-functional requirements, i.e. QoS (Quality of Service) [3,4]. There has been a stronger focus recently on selecting a service that meets a user's QoS requirements among many services with similar functions [5]. Monitoring the runtime QoS is a key means to ensure the accurate service selection.

A variety of monitoring methods have been devised for probabilistic quality attributes. These include QoS monitoring methods based on traditional probability statistics [3], hypothesis testing [4,6] and Bayes' theorem [7,8]. Those

* Corresponding author

Wei, T., Zhang, P., Dong, H., Jin, H., Bouguettaya, A. (2022). Mobility-Aware Proactive QoS Monitoring for Mobile Edge Computing. In: Troya, J., Medjahed, B., Piattini, M., Yao, L., Fernández, P., Ruiz-Cortés, A. (eds) Service-Oriented Computing. ICSOC 2022. Lecture Notes in Computer Science, vol 13740. Springer, Cham. https://doi.org/10.1007/978-3-031-20984-0_9

methods aim to perform continuous QoS monitoring based on user-defined standards in addition to computation overhead reduction. However, these methods encounter the following problems in the mobile edge environment:

*Traditional QoS monitoring approaches lack a proactive mechanism.* Service providers usually deploy a large number of services in the network environment. It is impractical for sensors to monitor and record in real-time the QoS generated by different users due to time, financial and resource constraints. In addition, monitoring the current status of a service cannot fully prevent the service from failure. In this regard, the monitoring results received by a user at present can only reflect the service status in the past due to the network latency. Therefore, it is essential to develop proactive service monitoring solutions to detect service failure in advance.

*The current QoS monitoring approaches ignore the temporal and spatial characteristics of QoS.* Our literature survey reveals that existing QoS monitoring approaches overlook the spatio-temporal dependency of QoS. This defect may lead to deviation of monitoring results from the real situation. The QoS of a service (observed from the client side) relies on the state of the service (on the server side) and the network environment. The service state is impacted by the server capacity and workload, the allocated computing resources, etc. The network environment is influenced by users and servers' locations, network bandwidth and traffic, the number of clients, etc. Both of them are highly dynamic over time and space.

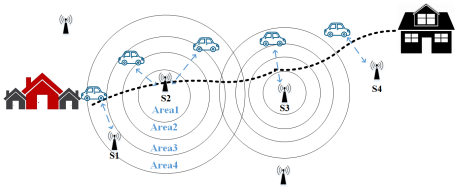



Fig. 1: Motivation scenario

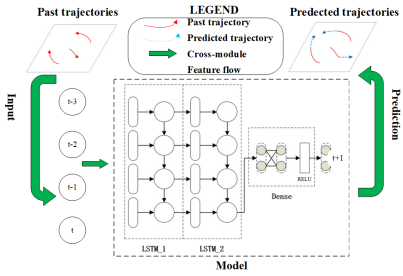



Fig. 2: Architecture of model

## 2 Related Work

Many probabilistic QoS monitoring techniques based on Bayesian classifiers were proposed to address the limitation of the aforementioned methods on variable user requirements. A new mobility and dependency-based QoS monitoring method named ghBSRM-MEC was presented in [9]. This method assumes that the QoS attribute value of an edge server obeys Gaussian distribution. A parent attribute is constructed for each attribute, thereby reducing the dependence between attributes. A Gaussian implicit Bayes classifier is constructed for each edge server to realize QoS monitoring in the mobile edge environment.

Proactive monitoring techniques have also been applied into other fields. A QoS monitoring algorithm that can quickly detect broken or congested links was

depicted in [10]. This algorithm takes advantage of a multithreaded design based on lock-free data structures. It improve the performance by avoiding synchronization among threads. Their work specifically focuses on real-time streaming. It does not realize proactive QoS monitoring. A proactive solution was introduced in [11]. It migrates the virtual machines before violating the actual delay threshold. The authors proposed a delay-aware resource allocation method that considers an adaptive delay warning threshold for various users. Their work focuses on dynamic resource allocation for hosting delay-sensitive vehicular services in a federated cloud. It cannot realize proactive QoS monitoring.

All the above monitoring methods do not take into account the proactive selection of servers by capturing the mobility of users in mobile edge environments. They also ignore the temporal and spatial dependency of QoS monitoring. These defects would lead to their failure to address the problems of lagging monitoring and long monitoring delay. This inspires us to devise a context-dependent proactive QoS monitoring method to fully cater to mobile edge environments.

## 3 The LSTM-BSPM Approach

As shown in Fig. 1, we use a mobile edge service scenario to illustrate our motivation. And its main framework is shown in Fig.3. It mainly includes three steps.

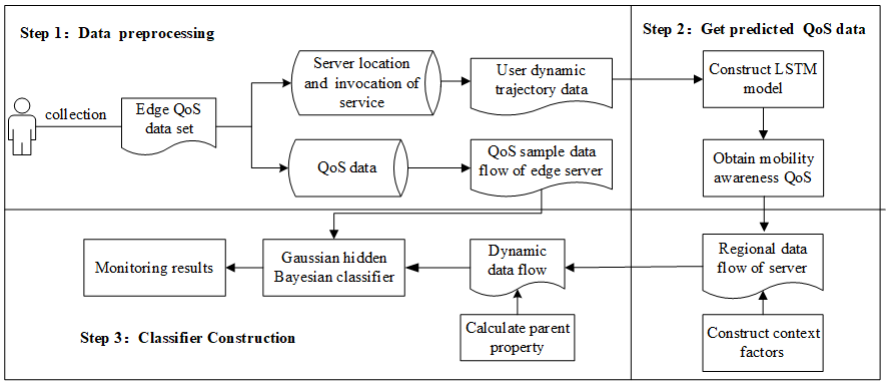



Fig. 3: Structure of proactive QoS monitoring

### 3.1 Data preprocessing.

First, we partition the spatial QoS data according to the locations of their belonged edge servers. The monitoring process in a mobile edge environment needs to consider user's historical trajectory data and information of service calls. The existing data sets do not meet such requirements. Hence, we need to construct a data set for mobile edge servers and users. The second major mission of the data preprocessing is to filter invalid data, such as the sample data with response time of -1 and 0. It makes the experimental data more in line with the real situation.

### 3.2 Mobility-aware QoS acquisition based on DouLSTM-Den

The primary purposes of this step is to construct and train the DouLSTM-Den model to obtain the user's mobility-aware QoS. Here we propose a model named DouLSTM-Den to predict a user's future location. As shown in Fig. 2, DouLSTM-Den comprises an LSTM layer with 3 units, a hidden LSTM layer with 2 units, and a normal dense layer with 2 hidden outputs for 2 columns. The details how this structure is determined is explained in the evaluation part.

The original trajectory data of the moving user is converted into a sequence of $h$ positions $H_i = \{Y_1, Y_2, ..., Y_h\}$, where $H_i$ represents the movement trajectory of $user_i$, $Y_i = \{lng_m, lat_m\}$ represents the $m$th longitude and latitude of $user_i$ based on time series. The current location is $Y' = \{lat_t, lng_t\}$. In practice, we continuously update the trajectory by combining the current location of the user for trajectory prediction.

We predict the $t+1$th location $Y_{t+1}$ of the $user_i$ through the DouLSTM-Den model. A high-level definition of the DouLSTM-Den model can be expressed as:

$$Y_{t+1} = f(\{Y_1, Y_2, ..., Y_h, Y_t\}) \tag{1}$$

Its technical details can be referenced from Section 3.2.

The network conditions in different coverage areas of an edge server are odd. In this regard, the network loads in different locations are diverse. This would cause distinct QoS values among different coverage areas of an edge server. The coverage area of a sever is usually circular. We accordingly divide the coverage area of a server into several circular rings and monitor QoS in each circular ring.

We set the coverage of each edge sever to 2 kilometers by analyzing the users' locations under each server's coverage. The coverage of each edge server is divided into 5 circular areas through the analysis of user distributions. The circular areas are $[1, 400)$, $[400, 800)$, $[800, 1200)$, $[1200, 1600)$, and $[1600, 2000]$ based on their distance to an edge server.

We choose the server closest to a user as the edge server that the user is most likely to access. We then determine the exact circular area of the server. The historical QoS data of the service to be invoked by the user is extracted from all the users in the same circular area of the predicted edge server. It is denoted by $T_{area_{t+1}} = \{T_{u_1}, T_{u_2}, ..., T_{u_n}\}$, where $T_{u_i}$ represents the QoS of the service invoked by the user $i$. The average value of the historical QoS data is calculated to obtain the mobility-aware QoS of the service. It is denoted by $QoS_{t+1} = \sum_n^1 T_{area_{t+1}}/n$, where $n$ is the number of the users in this area.

### 3.3 QoS monitoring based on Gaussian Hidden Bayesian classifier

The main purpose of this step is to train a Gaussian Hidden Bayes classifier based on historical data. The classifier will proactively monitor the mobility-aware QoS acquired from the last step. A Naive Bayes classifier assumes that the attribute values are independent of each other. However, tt ignores the fact that there might be dependence between QoS attribute values, leading to inaccurate classification results. Here we define a parent attribute $\pi(x_i)$ to reduce the dependence between QoS attributes. Each parent attribute represents the influence of the other attributes to each independent attribute. The value of the

parent attribute $\pi(x_i)$ is the mean value of $x_1 \sim x_{k-1}$. The improved Bayesian classifier formula can be expressed as:

$$C(X) = \arg\max_{c_j \in C} \{P(c_j) \prod_{i=1}^{n} P(x_i|\pi(x_i), c_j)\} \tag{2}$$

The Gaussian distribution is generally used to represent the class conditional probability distribution of continuous attributes. We apply Gaussian distribution to the probability distribution of continuous variables in Bayesian classifier. The assumption of the Gaussian distribution is expressed as follows:

$$P(x_i|\pi(x_i), c_j) = N_{c_j} \begin{pmatrix} u_{x_i} + \rho \frac{\sigma_{x_i}}{\sigma_{\pi(x_i)}}(\pi(x_i) - u_{\pi(x_i)}), \\ \sigma_{x_i}^2(1-\rho^2) \end{pmatrix} \tag{3}$$

where $N_{c_j}$ represents the Gaussian distribution of the corresponding category $c_j$, $u_{x_i}$ and $\sigma_{x_i}^2$ are the mean and variance of the sample attributes, and $u_{\pi(x_i)}$ and $\sigma_{\pi(x_i)}$ are the mean and variance of the parent attributes corresponding to the sample. The correlation coefficient between $x_i$ and $\pi(x_i)$ is denoted by $\rho = \frac{conv(x_i, \pi(x_i))}{\sigma_{x_i}\sigma_{\pi(x_i)}}$ .

In the training phase, a Gaussian hidden Bayesian classifier is constructed upon its parent attributes for each sample, i.e., the mobility-aware QoS value of the user. The classifier is trained based on the historical data of each edge server. The spatio-temporal QoS data (i.e., the QoS data in the same circular area of a sever within the same time period) is used as the input in the classifier. Every time a new QoS value is obtained, whether or not the QoS value satisfies with the pre-defined probabilistic requirements can be determined. We assume that the QoS attribute value follows the Gaussian distribution. Therefore, the determination can be implemented by the probability density integral formula:

$$P(X < Qos\_Value) = \int_{-\infty}^{Qos\_Value} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-u)^2}{2\sigma^2}} \tag{4}$$

where $\mu$ and $\sigma$ represent the mean and standard deviation of the QoS value. For example, if a QoS requirement is that the probability that the service response time is less than 2s is greater than 85%, the value of $QoS\_Value$ is 2.

In the QoS monitoring process, users pre-define a set of QoS requirement vectors as $T_{QoS} = [X_1, X_2, \cdots, X_n]$, where $X_n = [x_1, x_2, \cdots, x_n]^T$ refers to the set of required QoS values of all the services called by the user $n$ when accessing a server. The category set is $C = \{c_0, c_1\}$, where $c_0$ refers to a satisfactory grade and $c_1$ refers to a unsatisfactory grade. The posterior probabilities of $c_0$ and $c_1$ are calculated via the aforementioned process. The category with a higher posterior probability is regarded as the final monitoring result.

## 4 Experiment

### 4.1 Experimental Environment Configuration

**Experiment setup.** The TensorFlow 2.4.0 deep learning framework[4] is used to implement the proposed DouLSTM-Den model. The model is trained with a

[4] *https://github.com/tensorflow/tensorflow/tree/v2.4.0*

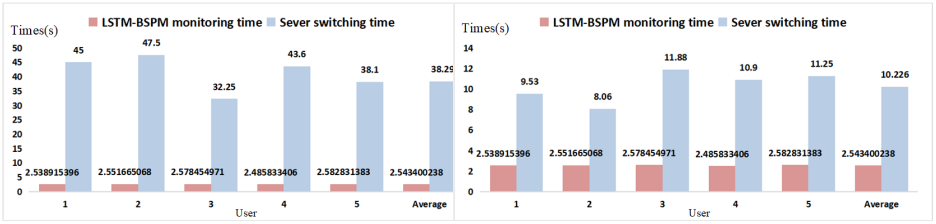
computer with Nvidia GTX1080Ti GPU. The model is trained 30 epochs with a batch size of 128. The initial learning rate is set to 0.001. All these parameters are optimal settings according to our experimental observation.

**Data sets.** This experiment involves three data sets in the experiment.

- Data Set 1 bases on the Shanghai Telecom data set[5]. This data set includes the geographic location information of 3,233 base stations and 611,507 service calling records.
- Data Set 2 bases on a real-world Web service quality data set released by Chinese University of Hong Kong[6]. This data set includes the response time of 4,500 Web services called by 142 users in 64 different time slices.
- Data Set 3 is a simulated verification data set. The verification data set is generated according to users' QoS requirements in the experiment. The verification data is used to verify the effectiveness of the proposed method. For example, if the QoS requirement is that the probability that the response time of the service is less than 3.6s is greater than 80%, we inject more than 20% exceptional response time (i.e. greater than 3.6s) samples in a certain range of the original samples as the verification data.

**Comparison method.** We compare LSTM-BSPM with the following state-of-the-art service quality monitoring methods to verify the superiority of LSTM-BSPM. These include ghBSRM [9], wBSRMM [8] and IgS-wBSRM [12].

### 4.2 Feasibility verification of proactive monitoring




(a) When driving cars (b) When taking high-speed trains

Fig. 4: Time consumption comparison between proactive service monitoring ($t_{LSTM-BSPM}$) and server switching ($t_{tra}$)

We set up an experiment to assess the feasibility of the proposed method. We verify whether our approach can detect abnormal service states before users access new edge servers. The experiment assumes that a group of 160 users call services when driving a car and taking a high-speed train respectively. We assume that the speed of the vehicle is $72km/h$ and the speed of the train is $300km/h$. The monitoring time $t_{LSTM-BSPM}$ mainly contains two parts: the time $t_{LSTM}$ to obtain the mobility-aware QoS attribute value based on the DouLSTM-Den

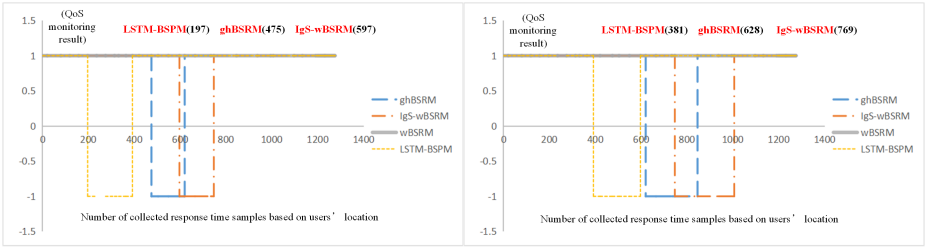
[5] $http://sguangwang.com/TelecomDataset.html$
[6] $http://wsdream.github.io/dataset/wsdream_dataset2.html$

model, and the time $t_{mon}$ to monitor the QoS using the Bayesian classifier. The estimated time $t_{tra}$ required for a user to access a new edge server is obtained by calculating the distance between two edge servers divided by the speed.

Fig. 4a and Fig. 4b respectively show the time needed for proactive monitoring and connecting to a new edge server for 5 randomly selected users and all the users when driving and taking high-speed trains respectively. We can draw a conclusion that our approach can efficiently complete the proactive service monitoring before users access new edge servers. This would provide more time for servers to make decisions if service anomalies occur.

### 4.3 Effectiveness verification of positive monitoring

We establish an experiment to verify whether the proposed proactive monitoring method can more quickly and accurately detect service exceptions before users calling the services. The proposed method is compared with the three aforementioned baseline methods. Data Set 3 is used for the experiment. First, we extract the QoS values of 2000 services to train a Gaussian hidden Bayes classifier. We then inject 200 exceptional samples with response time of 3s in the ranges of [200,400] and [400,600] of 1000 test samples (i.e. services).




(a) Exceptional samples injected in [200,400]

(b) Exceptional samples injected in [400,600]

Fig. 5: Result of response time monitoring

Fig. 5a and Fig. 5b respectively show the monitoring results of the exceptional samples injected in different intervals. The abscissa represents the number of samples that a monitoring method can obtain based on the test set The ordinate represents the monitoring result, where 1 represents normal, and -1 represents abnormal. The number of samples required for each method to monitor the abnormality of the service status is marked on the top of the diagram. It can be seen that the proposed proactive monitoring method (i.e. LSTM-BSPM) needs the lowest numbers of samples to detect the service exceptions. In general, it can be seen that the prediction results of LSTM-BSPM are more consistent with the injected exceptions. The experimental results verify the effectiveness of the proposed proactive monitoring method in the mobile edge environment.

## 5 Conclusion

This paper presents a proactive QoS monitoring method in the mobile edge environment based on DouLSTM-Den model and a Gaussian hidden Bayes classifier.

Experiments are conducted on both simulated and real data sets. The experimental results show the effectiveness and feasibility of the proposed method.

For the future work, the following tasks will be considered: i) we will design solutions to accurately predict users' multi-lag moving paths; ii) we will improve this method to adapt to multivariate QoS monitoring; iii) we will consider user privacy protection when designing future proactive QoS monitoring methods.

## 6 ACKNOWLEDGMENTS

This work is funded by the National Natural Science Foundation of China under Grant (No.62272145, No.U21B2016), the Natural Science Foundation of Jiangsu Province under grant No.BK20191297, the Fundamental Research Funds for the Central Universities under grant No.B210202075. This research was also partially supported by the Australian Government through the Australian Research Council's Discovery Projects funding scheme (project DP220101823).

## References

1. S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: Convergence of computing, caching and communications," *Access*, vol. 5, pp. 6757–6779, 2017.
2. S. Wang, J. Xu, N. Zhang, and Y. Liu, "A survey on service migration in mobile edge computing," *Access*, vol. 6, pp. 23511–23528, 2018.
3. K. Chan, I. Poernomo, H. Schmidt, and J. Jayaputera, "A model-oriented framework for runtime monitoring of nonfunctional properties," in *QoSA/SOQUA*, pp. 38–52, Springer, 2005.
4. I. Lee, O. Sokolsky, J. Regehr, *et al.*, "Statistical runtime checking of probabilistic properties," in *International Workshop on Runtime Verification*, pp. 164–175, Springer, 2007.
5. H. Billhardt, R. Hermoso, S. Ossowski, and R. Centeno, "Trust-based service provider selection in open environments," in *SAC*, pp. 1375–1380, 2007.
6. L. Grunske, "An effective sequential statistical test for probabilistic monitoring," *Information and Software Technology*, vol. 53, no. 3, pp. 190–199, 2011.
7. Y. Zhu, M. Xu, P. Zhang, W. Li, and H. Leung, "Bayesian probabilistic monitor: A new and efficient probabilistic monitoring approach based on bayesian statistics," in *QSIC-13*, pp. 45–54, IEEE, 2013.
8. P. Zhang, Y. Zhuang, H. Leung, W. Song, and Y. Zhou, "A novel QoS monitoring approach sensitive to environmental factors," in *ICWS*, pp. 145–152, IEEE, 2015.
9. P. Zhang, Y. Zhang, H. Dong, and H. Jin, "Mobility and dependence-aware qos monitoring in mobile edge computing," *TCC*, vol. 9, no. 3, pp. 1143–1157, 2021.
10. F. Tommasi, V. De Luca, and C. Melle, "QoS monitoring in real-time streaming overlays based on lock-free data structures," *Multimedia Tools and Applications*, vol. 80, no. 14, pp. 20929–20970, 2021.
11. M. Najm, M. Patra, and V. Tamarapalli, "An Adaptive and Dynamic Allocation of Delay-sensitive Vehicular Services in Federated Cloud," in *2021 COMSNETS*, pp. 97–100, IEEE, 2021.
12. P. Zhang, H. Jin, Z. He, H. Leung, W. Song, and Y. Jiang, "Igs-wbsrm: A time-aware web service qos monitoring approach in dynamic environments," *Information and software technology*, vol. 96, pp. 14–26, 2018.