Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 6 (2023)

Enhancing Edge Computing in Mobile Dependence and Qos Monitoring

[1] Sowmiya Sree C., [2] Rohini C., [3] Monisha S., [4] Dr.D.Usha

[1]Assistant Professor,[4]Professor, [2,3] Final B. Tech CSE,

Dept of Computer Science and Engineering, DR Mgr Educational And Research Institute, Chennai

Abstract A new computing paradigm that operates at the network's edge is known asmobileedgecomputing.Byplacingedgeserversclosetomobiledevices,itoffersservices to users. Due to shifting edge environments, services may not functionproperlyor do not meet user requirements.The concept of "quality of service"(QoS) is widely recognized as a vital measure of a service's excellence. Mobile edge computing becomes even more crucial to promptly and effectively monitor the QoS of services. However, due to user mobility and interdependencies in QoS values, monitoring resultsgenerally departfrom the actual outcomes in the mobile edge environment. Unfortunately, the current QoS monitoring strategies fail to consider these challenges. To address this, we introduceghBSRM-MEC (Gaussian hidden Bayesian Runtime Monitoring for Mobile Edge Computing), an innovative strategy that takes into account mobility and dependence factors in QoS monitoring. This paper presents ghBSRM-MEC as a promising solution to these issues.

Keywords-*BayesianClassifier, QoS Monitoring, Mobile EdgeComputing,Cloud Computing.*

1. Introduction

A developing technology known as Mobile Edge Computing (MEC) implements services by placingan edge server(suchasarouter, firewall, or otherdeviceofasimilartype)closetomobileclients(suchassensors,smartphones,orotheredgeends) and between cloud servers andmobile clients. It has a quick processingspeed and a quick response time. Web services are commonly used in various of our lives, including business, manufacturing,healthcare, entertainment, and the creation of new technologies.

Various service providers may offer services that have comparable functionalities, yet the execution of these services can differ across different edge servers. QoS incorporatesa range of non-functional characteristicsof services, like availability,throughput,reliability,as well as response time, among others. These attributes can exhibit variations between edge servers and service providers, even for services that possess similar functionalities.Users anticipate choosing mobile edge services that provide secured QoS.

The service provider typically discloses a service's QoS information. Some providers might provide misleading Quality of Service(QoS) information to the users. Relying on inaccurate data is not dependable for evaluating the quality of services.As aresult, timely and efficient monitoring of QoS attributes at runtime is crucial foraccuratelyevaluatingtheoperationofservices.Monitoring is one of the efficient methodsto find out if a service is accurate at runtime. ProbabilisticqualityattributescantypicallyrepresentQoSattributes.

2. System Overview

Cloud computing is the process of storing and retrieving data and applications on servers located remotely over the internet, rather than on local servers or the computer's hard drive. It is also referred to as Internet-based computing, where users can access resources as services through the Internet. Cloud computing allows for the

storage of various types of documents, such as files, photos, and papers. The utility concepts impact the metrics employed for evaluating the performance of cloud computing services. Additionally, cloud computing aims to tackle the quality of service and reliability concerns that are encountered in other grid computing models.Cloud computing strives to enable users to leverage various technologies without requiring an in-depth understanding of each.

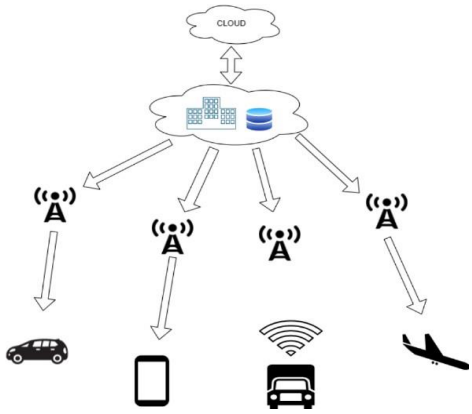

Fig.1.Mobile Edge Computing

3. System Model

3.1CLASSICAL QoS MONITORING

This method involves monitoring techniques to collect data in real-time or from users.To monitor QoS, Zeng et al. [21] implemented a QoS observation model. Their monitoring system is capable of detecting and routing QoS and operationevents of services. Despite it concentrates on metrics andevaluation formulae at the enterprise level, this technique overlooks user needs and scalability. Radovanovic et al.[22]used TR-069, also recognized as CWMP("CPE WAN Management Protocol")[23],remote management to set up a cloud-based monitoring system. The cloud-based access interface of this system gives mobile applications essential information that enables the viewing of obtained QoS parameters.The present monitoring system is tailored to the specific needs of enddevices, making it applicable in a varietyof settings.For monitoring of QoS, Michlmayret al.[24] proposed an architecture that includes both client-side as well as server-side monitoring. Event handling is used in this framework to notify interested users about present QoS levels and breaches of service-level protocols. If the QoS is not satisfactory, adaptive measures may be initiated.The proposed framework, however, has a significant performance overhead.Coppolino et al. [25] blended QoS-MONaaS ("QoSMONitoring as a Service") andSocIoS.The QoS monitoring from the SRT-15 project is integrated into the SocIoSframework, and theSocIoSapplication is monitored using QoS-MONaaS.An online monitoring methodon the basis of SLA ("Service Level Agreement") is employed in a non-intrusive way. This approach analyzes the timestamp and the kind of communication to find violations of agreements. To evaluate the extensibility of this approach, more research is necessary. Furthermore, this method ignores the possibility that SLAs may impose requirements on non-time attributes and only takes time-related attributes into account. These methods mainly depend on the expensive collection of real-time data and prompt user input.

3.2 PROBABILISTIC QoS MONITORING

In recent years, there has been a rise in the use of probabilistic monitoring approaches. These approaches convert users' imprecise requirements into probabilistic descriptions and utilize historical data for statistical analysis to make monitoring decisions. Based on different theoretical approaches, there are presently three main categories of QoS monitoring approaches.

The first category relies on traditional probability approaches, with Chan et al.[11]being the pioneers of this probabilistic monitoring approach.

A probabilistic monitoring approach of the second category relies on the theory of hypothesis testing. In their study, Sammapun et al. [12] computed the likelihood of the successful samples being representative of the entire sample set, and subsequently usinga hypothesis test, ascertained whether the system, with a specified level of certainty, meets the prerequisites for a probabilisticcriterion.

The Bayesian theory serves as the foundation for the third category of approach. What sets Bayesian methods apart is their ability to incorporate historical empirical data into present predictions, utilizing both prior and liable probabilities to articulate volatility. Zhu et al. originally presented BaProMon, a Bayesian probabilistic QoS monitoring method [15]. It assesses runtime data by computing Bayesian factors to ascertain whether the monitoring results corroborate the initial theory or a different one. They then presented an improved method, called lgS-wBSRM, in their later work [17].A sliding window method was applied to progressively eliminate early and inappropriate samples.

A challenge faced by classical QoS monitoring approaches is their limited suitability for mobile edge environments, as they do not account for user mobility and the interdependence of QoS values. To overcome this issue and mitigate the impact of user mobility and QoS value interdependence on monitoring results, a new QoS monitoring approach is proposed in this paper.

4. System Methodology

4.1 MEC

It is a configuration that drives cloud computing and IT resources closer to the mobile networkedge, improving the performance of mobile and Internet of Things applications. Most often, this is acquired by aligning them in proximity to cellular base stations. By processing data and running applications at the network's edge, MEC aims toreduce latency and improve overall efficiency, especially in scenarios where data is generated close to the edge.

MEC infrastructure utilizes edge servers, which can be either physical or virtualized computing resources, in close proximity to the mobile network's access points. These servers can accommodate applications, services, and content, thereby minimizing the time taken for data to travel to and from a centralized data center.

The management and orchestration of MEC resources at each edge location is the responsibility of edge nodes. These nodes efficiently allocate computing and network resources and can optimize the placement of edge applications according to the real-time demands of users and applications.

4.2SERVICES AND EDGE SERVICES

Classical services typically consist of functional and non-functional features. The functional characteristics include elements such as service inputs (I),service descriptions(D), service outputs(O), and other related aspects. Conversely, non-functional attributes primarily consist of QoS factors, like credibility,reliability,andresponse time.

Edge services, which enable software inter-operations in edge networks, are a novel type of service. The non-functional features of edge services are significantly impacted by user movement, changes to server settings, and changes to network circumstances[20].

4.3HIDDEN NAIVE BAYES

Naive Bayes, a popular probabilistic algorithm in machine learning, is frequently employed for classification and supervised learning purposes. It relies on Bayes' theorem to determine the likelihood of a hypothesis (class label) given the available data. The word naive is derived due to its presumption of conditional independence. While this assumption simplifies probability calculations, it may not always be valid in real-world scenarios.

The following is an expression for the formula:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \frac{P(B|A)P(A)}{P(B)} \quad (1)$$

A Bayesian theorem is based on the combination of class conditional density and prior probability of a Bayesian classifier. It is frequently used in data mining classification problems because of its ease of use and great efficiency. The procedure entails figuring out how likely it is for a given sample to fall into each category,and then assigning the samples to the categories with the highest probability.

Let C = {c0, c1, ...cj} be considered to be a set of predefined categories and X = {x1, x2, ...xn}considered to be a sample vector. The probability that X belongs to cj is presented as P (Cj |X) and is calculated using the Bayesian formula.

$$P(cj\,|X) = \frac{P(cj)P(X)|cj)}{P(X)} P(cj\,|X) = \frac{P(cj)P(X)|cj)}{P(X)} \quad (2)$$

With a given attributes $x_k x_k$ and $x_i x_l$ of X are mutually exclusive and P(X) is uniform acrosseach categoryin which X belongs to cj, the formula of the Bayesian classifier could be expressed as,

$$C(X) = \underset{c_j \in C}{argmax} \{P(c_j) \prod_{i=1}^{n} P(x_i|c_j)\} \quad (3)$$

C(X) represents the category of X in the current inquiry. The Bayesian classifier, which functions on the premise that characteristics are independent of one another, ignores the dependence between attribute values. A technique to improve the performance of the Bayesian classifier is to introduce a hidden parent attribute for every attribute and reduce the degree of independence between them. This parent attribute shows theimpact of other characteristics on a specific attribute. The structure of the enhanced Bayesian classifier is revealed in Figure 2. π($x_l x_l$),the "hidden parent attribute"for $x_i x_i$, denotes the impact of other attributes on $x_i x_i$This could be presented using the following formula.

$$C(X)=\frac{argmax_{c_j \in C}\{P(c_j) \prod_{i=1}^{n} P(x_i|\pi(x_i),c_j)\} argmax_{c_j \in C}\{P(c_j) \prod_{i=1}^{n} P(x_i|\pi(x_i),c_j)\}}{} \quad (4)$$

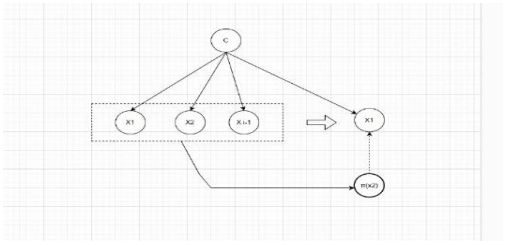

Fig.2. Hidden Naïve Bayes's structural diagram

4.4 KNN:

The KNN ("K-Nearest Neighbors") algorithm is a simple and easy-to-understand ML ("Machine Learning")method used for regression and classification purposes. It is an instance-based and non-parametric learning algorithm that is frequently employed for data-driven decision-making, particularly when the underlying data distribution is uncertain.

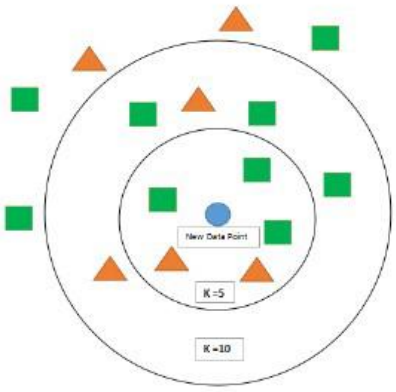

Fig.3.The sample graph of KNN

One way to classify data is to use a voting mechanism. A prediction is made using the labeled category that appears consistently among the k nearest neighbors.For instance, consider the sample data shown in Figure 3, consisting of two categories: squares and triangles. The unclassified sample, represented by a circle, is to be categorized. If k=3, there are 1 square and 2 triangles in close proximity to the unlabeled sample. The unclassified sample is thus classified as a triangle based on the results of the voting of the KNN.However, when k=9, the unclassified sample is classified as a square. Regression, on the other hand, involves averaging the values of the KNN to predict numerical values for unlabeled samples.These techniques can be improved with weighted voting and averaging, in which the weight is based on the between a neighbor and the unlabeled sample. Theweight of a neighbor increases with proximity.

4.5 *THE GHBSRM-MEC APPROACH*

Relying exclusively on preceding data from previous servers may lead to inaccurate monitoring results for services in new edge servers because of the dynamic changes in user locations within MEC. The problem of creating parent attributes among the QoS attributes can be lessened and the monitoring errors brought on by the dependencies between these attributes can be minimized. By using the KNN algorithm and data from peripheral edge servers, the system can also monitor edge servers without historical data and switch between them dynamically.The primary structure of ghBSRM-MEC is represented in Figure 4. It mainly consists of three sequential steps:

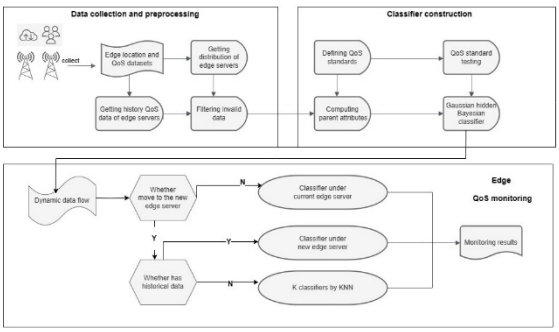

Fig 4.Structure of ghBSRM-MEC

Step 1: Data collection and preprocessing

Data collectionencompasses the acquisition of both the geographical dispersion of edge servers and theQoS information from each server. The positions of the edge servers are crucial in determiningtheir spatial distribution across different regions. On the other hand, the QoS data obtained from these servers plays a vital role in the development of tailored classifiers for each specific edge server.Data preprocessing primarily focuses on eliminating erroneous samples from edge servers. Furthermore, in our methodology, synthetic sample data is generated based on QoS requirements to serve as validation samples in the experiment.

Step 2: Classifier construction

The probabilistic QoS requisite is established based on the user's specific need, which aims to guarantee that the response time of the service has a probability greater than 80% of being less than 3.6s. Subsequently, the preceding data obtained from each edge server is utilized to evaluate whether a new QoS sample satisfies this probabilistic requirement.The model's parameters can be determined by computing the parent attribute of the QoS attribute. Ultimately, a classifier is created using these calculated values.

Step 3: Edge QoS monitoring

The evaluation of Quality of Service (QoS) in the ghBSRM-MEC method encompasses the analysis of the user's transition to an alternative edge server as well as the condition of the server itself. When the user remains within the service coverage area of the present server, the monitoring outcomes are derived from the classifier within that particular server. Conversely, if the user relocates to a different edge server that possesses preceding data, the monitoring findings are derived from that specific dataset. The proposed approach chooses the k closest edge servers when the new edge server lacks access to preceding data. The posterior probability of these chosen servers is then computed and further weighted according to their separation from the new edge server. This approach allows for obtaining accurate monitoring results.

Figure 5 emphasizes the significance of considering the locations of both the user and the server, as well as the availability of preceding data whenmonitoring a service invoked by a user on aserver.

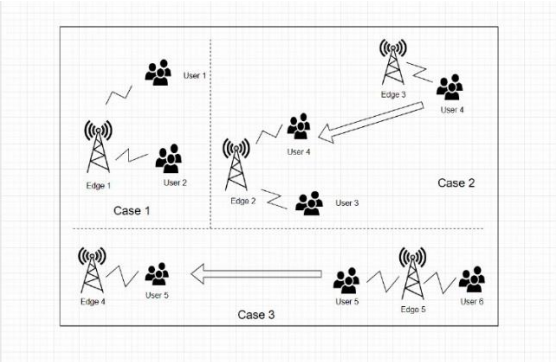

Fig. 5. Mobile Edge Monitoring Services

For formula 4, the determination of $P(^{C_J C_J})$ typically relies on the maximum likeliness of the sample. When monitoring a new service, the classifier must verify if it satisfies the probabilistic QoS requirement. Assuming a Gaussian distribution, the probability measurement can be achieved through the integration formula:

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

5. Results And Discussions

The validation of ghBSRM-MEC in mobile edge computing environments is carried out in this section using a combination of real and assumed QoS data sets. A development platform called Geany 3 is utilized for conducting the experiments, while Python is employed for designing and implementing ghBSRM-MEC.

It utilizes three distinct data sets. The Shanghai Telecom data set [18] is the first data set.The real-world QoS data set, released by the CUHK ("Chinese University of Hong Kong"), is used in the second data set. Finallythird data set, which is merged into the fused "Shanghai Telecom CUHK"data sets, is a collection of randomly created dissatisfied QoS monitoring samples.

The proposed approach is validated and the edge environment is constructed using the concept ofdata fusion.The QoS data sets are geographically diverse sources in mobile edge environments. To satisfyedge characteristics, 2data sets are merged to create a unified data set. Initially, services from the same locations in the QoS data set are grouped based on their similarities, and the collected data is organized.The precise location of edge servers is then determined by selecting a random subset of base stations. Lastly, the edge environment is created to identify the edge locations and assign the relevant edge servers to the consolidated sample QoS data.

Several effective QoS monitoring approaches have been introduced to compare with ghBSRM-MEC. One such approach is iBSRM, which is depending on the Bayesian statistics. It uses the Bayesian factor to make monitoring decisions through hypothesis testing and can achieve continuous monitoring by reusing previous monitoring results. Another approach is wBSRM, which considers the impact of QoSenvironmental factorsand uses TF-IDF to quantify their impact.The samples are weighed using the quantitative value, and preceding data can be used for monitoring, decision-making, and also for creating a weighted Bayesian classifier. Additionally, IgS-wBSRMis an enhanced version of wBSRM that incorporates a sliding window mechanism and calculates information gained to dynamically update the initially assigned weight.

A range of experiments were carried out on all datasets to assess the potential of ghBSRM-MEC in terms of feasibility, effectiveness, efficiency, and user mobility management. Additionally, the impact of k was also examined. As all edge servers possess comparable characteristics, the outcomes of each segment of the experiment were presented on a selection of edge servers chosen at random**.**

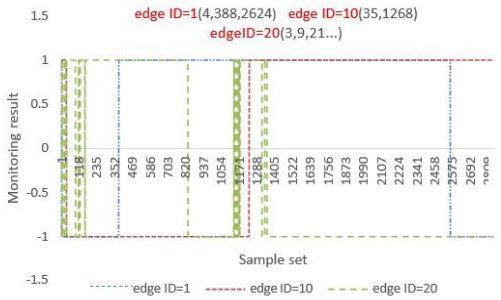

Fig.6.Different Edge Servers monitoring results

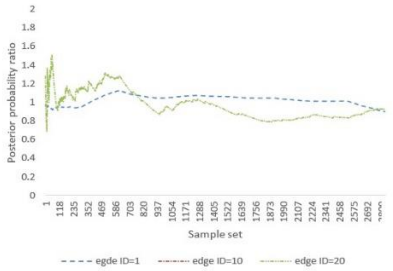

Fig.7.Different Edge Servers posterior probability

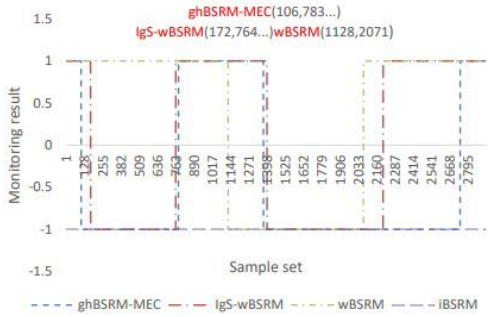

Fig.8.Compared approaches to monitoring results

Fig.9.Compared approaches average monitoring time

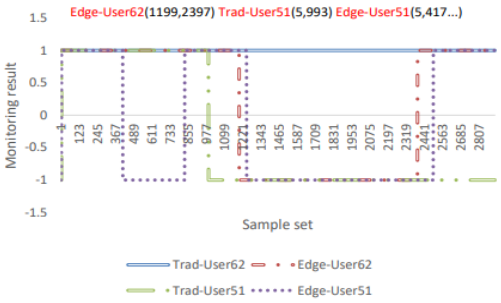

Fig.10.Monitoring results response time

6. Conclusion

The potential for enhancing performance, efficiency, and user experiences in Mobile Edge Computing (MEC) throughdependance-aware Quality of Service (QoS)monitoring is immense. This discussion presents a

comprehensive approach to address the challenges associated with QoS monitoring in MEC, including advanced mobility management techniques, dependence-aware QoS metrics and analysis, adaptive QoS monitoring engine, machine learning-based QoS prediction module, and cross-layer optimization module.

The proposed system optimizes resource utilization, predicts potential QoS degradation, and enables proactive measures to maintain satisfactory QoS levels.Moreover, the integration of security and privacy measures ensures the integrity andconfidentiality of QoS monitoring data while respecting user privacy requirements.

The future of mobility and dependence-aware QoS monitoring in MEC lies in the development of sophisticated techniques, algorithms, and systems that consider the dynamic nature of mobile devices, the interdependencies among services, and the evolving requirements of users. These advancements can significantly improve QoS management, optimize resource allocation, and ultimately enhance user satisfaction in MEC environments. Further research and development in this area will improve the way for a more efficient future.

Reference:

[1] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, “Mobile edge computing—a key technology towards 5G,” ETSI white paper, vol. 11, no. 11, pp. 1–16, 2015.

[2] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: Vision and challenges,” IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637–646, 2016.

[3] M. Urkia Kortabarria, “Web Service Performance on Heterogeneous Systems: A performance comparison between J2EE and. NET on heterogeneous systems,” 2013.

[4] H. Billhardt, R. Hermoso, S. Ossowski, and R. Centeno, “Trust-based service provider selection in open environments,” in Proceedings of the 2007 ACM symposium on Applied computing, pp. 1375– 1380, ACM, 2007.

[5] Raparthi, M., Dodda, S. B., & Maruthi, S. (2023). Predictive Maintenance in IoT Devices using Time Series Analysis and Deep Learning. Dandao Xuebao/Journal of Ballistics, 35(3). https://doi.org/10.52783/dxjb.v35.113

[6] J. El Haddad, M. Manouvrier, and M. Rukoz, “TQoS: Transactional and QoS-aware selection algorithm for automatic Web service composition,” IEEE Transactions on Services Computing, no. 1, pp. 73–85, 2010.

[7] D. Gunter, B. Tierney, K. Jackson, J. Lee, and M. Stoufer, “Dynamicmonitoringofhigh-performancedistributedapplications,”in*Proceedings 11th IEEE International Symposium on High-PerformanceDistributedComputing*,pp.163–170,IEEE,2002.

[8] M. Daniel and T. Menasc, “QoS issues in webservices,” IEEE Internet Computing, vol. 6, no. 6, pp. 72–75, 2002.

[9] L. Baresi and S. Guinea, “Towards dynamic monitoring of WS-BPEL processes,” in International Conference on Service-Oriented Computing, pp. 269–282, Springer, 2005.

[10] M. Leucker and C. Schallhart, “A brief account of runtime verification,” The Journal of Logic and Algebraic Programming, vol. 78, no. 5, pp. 293–303, 2009.

[11] L. Grunske, “Specification patterns for probabilistic quality properties,” in 2008 ACM/IEEE 30th International Conference on Software Engineering, pp. 31–40, IEEE, 2008

[12] K. Chan, I. Poernomo, H. Schmidt, and J. Jayaputera, “A modeloriented framework for runtime monitoring of nonfunctional properties,” in Quality of Software Architectures and Software Quality, pp. 38–52, Springer, 2005.

[13] 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.

[14] L. Grunske and P. Zhang, "Monitoring probabilistic properties," in Proceedings of the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering, pp. 183–192, ACM, 2009

[15] L. Grunske, "An effective sequential statistical test for probabilistic monitoring," Information and Software Technology, vol. 53, no. 3, pp. 190–199, 2011.

[16] 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 2013 13th International Conference on Quality Software, pp. 45–54, IEEE, 2013.

[17] P. Zhang, Y. Zhuang, H. Leung, W. Song, and Y. Zhou, "A novel QoS monitoring approach sensitive to environmental factors," in 2015 IEEE International Conference on Web Services, pp. 145–152, IEEE, 2015

[18] P. Zhang, H. Jin, Z. He, H. Leung, W. Song, and Y. Jiang, "IgSwBSRM: A time-aware Web Service QoS monitoring approach in dynamic environments," Information and software technology, vol. 96, pp. 14–26, 2018.

[19] S. Wang, Y. Zhao, L. Huang, J. Xu, and C.-H. Hsu, "Qos prediction for service recommendations in mobile edge computing," Journal of Parallel and Distributed Computing, vol. 127, pp. 134–144, 2019.

[20] Y. Xu, J. Yin, S. Deng, N. N. Xiong, and J. Huang, "Context-aware QoS prediction for web service recommendation and selection," Expert Systems with Applications, vol. 53, pp. 75–86, 2016.

[21] Z. Liu, Q. Sheng, X. Xu, D. Chu, and W. E. Zhang, "Context-aware and Adaptive QoS Prediction for Mobile Edge Computing Services," IEEE Transactions on Services Computing, 2019, DOI: 10.1109/TSC.2019.2944596.

[22] L. Zeng, H. Lei, and H. Chang, "Monitoring the QoS for web services," in International

Conference on Service-Oriented Computing, pp. 132–144, Springer, 2007.

[23] S. Radovanovic, N. Nemet, M. ´ Cetkovi ´ c, M. Z. Bjelica, and ´ N. Teslic, "Cloud-based framework for QoS monitoring and pro- ´ visioning in consumer devices," in 2013 IEEE Third International Conference on Consumer Electronics Berlin (ICCE-Berlin), pp. 1–3, IEEE, 2013.

[24] H. Rachidi and A. Karmouch, "A framework for self-configuring devices using TR-069," in 2011 International Conference on Multimedia Computing and Systems, pp. 1–6, IEEE, 2011.

[25] A. Michlmayr, F. Rosenberg, P. Leitner, and S. Dustdar, "Comprehensive QoS monitoring ofWeb services and event-based SLA violation detection," in Proceedings of the 4th international workshop on middleware forservice oriented computing, pp. 1–6, ACM, 2009.

[26] L. Coppolino, S. D'Antonio, L. Romano, F. Aisopos, and K. Tserpes, "Effective QoS monitoring in large scale social networks," in Intelligent Distributed Computing VII, pp. 249–259, Springer, 2014