QoE Assessment of Enterprise Applications based on Self-motivated Ratings

Kathrin Borchert, Michael Seufert, Kathrin Hildebrand, Tobias Hoßfeld
University of Würzburg, Institute of Computer Science
Würzburg, Germany
{kathrin.borchert | seufert | kathrin.hildebrand | hossfeld}@informatik.uni-wuerzburg.de

***Abstract*—In most companies, enterprise applications, such as office products or databases, are heavily used by employees during work hours. Impairments and performance issues not only slow down business processes, but might also increase the frustration of the workforce. While Quality of Experience (QoE) has been widely studied for personal multimedia applications, such as video streaming, its application to the business usage domain is still in its infancy. Due to several reasons, e.g., the high complexity of IT infrastructure, classical QoE studies can hardly be transferred to business applications. These studies are often independent from the context of usage and actively poll ratings from their participants. This work contrasts the commonly used ”pull” method for collecting user ratings with a self-motivated ”push” approach. This approach is inspired by complaint systems, in which users can directly report problems with a technical system as soon as they notice them. Therefore, performance assessments of a business application from employees of a cooperating company are collected with both rating systems during a time span of 1.5 years. Besides the analysis of the interaction of users with the ”push” system, differences between the two methods are discussed. Further, QoE models for the monitored business application are derived based on the self-motivated ”push” ratings.**

I. Introduction

Quality of Experience (QoE) is a concept, which describes the degree of delight or annoyance of the user of an application or service [1]. While QoE has been widely studied for personal multimedia applications, such as video streaming or VoIP telephony, its application to the business usage domain is still in its infancy [2], [3]. In most companies, enterprise applications, such as office products or databases, are heavily used by employees during work hours. Outages or slow response times of applications or services not only slow down business processes, but might also increase the frustration of the workforce. This becomes even more important as an increasing number of business applications are served remotely from a server, e.g., in a data center or a cloud. This introduces additional network delays depending on the amount of transferred data, the physical location of the server, and the network capacity and load. Thus, the QoE of enterprise applications has to be considered as a major business driver, as it directly influences the motivation and productivity of the employees.

However, classical lab or crowdsourced QoE studies can hardly be transferred to the domain of business applications as the study would influence the daily business of the company [4]. First, the IT infrastructure could be highly complex, making it hard to identify the most important technical parameters of the business application. Second, the company might be reluctant to monitor or influence a production system. Third, as the QoE of enterprise applications can only be meaningfully assessed in the work context, the employees would need to participate in the QoE study during working, which would be time consuming and potentially distracting.

To nevertheless obtain subjective feedback in an enterprise context, in [3], a survey tool was designed, which combined the non-intrusive monitoring of response times in a production SAP enterprise software system with minimal subjective feedback. The tool uses a “pull” approach, meaning that the assessments are pulled once an hour by asking the users to rate the system performance. This approach can capture the workers’ perception over long time periods, however, it is costly and shows a coarse granularity.

Thus, this paper contrasts the “pull” approach with a “push” approach, in which workers themselves can push, i.e., trigger and submit, a subjective rating at any time. This approach is inspired by complaint systems, in which users can directly report problems with a technical system as soon as they notice them. The interactions of users with the “push” system are analyzed and differences between the “pull” and “push” approaches are discussed. Finally, QoE models for the monitored SAP enterprise software system are derived, which are only based on the self-motivated “push” ratings of the employees.

Therefore, the paper is structured as follows. Section II outlines related works on QoE of enterprise applications and the usage of self-triggered feedback. Section III describes the survey tool and the collected data set. The “pull” and “push” approaches are contrasted in Section IV, and the QoE models are presented in Section V. Section VI concludes this work.

II. Background and Related Work

QoE of various applications and services has been subject of numerous studies. Besides the identification and analysis of influence factors, e.g., for web browsing [5] or mobile applications [6], much research focuses on the modeling of QoE [7]. Here, a wide spectrum of approaches is used to assess the user’s QoE based on technical parameters. These methods comprise, for example, machine learning approaches [8] or analytical models [9]. However, little research is done on the analysis and modeling of the perceived performance quality of business software. First results are presented in [10], which investigated the influence of loading delays of a fictive business application on the QoE. The analyzed subjective ratings were collected with the common used "push" approach. Even if this approach is in line with best practices for collecting QoE assessments, there is a lack of contextual factors. More context related approaches derive the perceived quality of business software from existing sources in the enterprise, e.g., support requests [11] or ticketing systems [12]. On the one hand, these approaches are easy to realize and less costly compared to surveys, on the other hand they offer only a limited view on the QoE. For less frequent used applications during working processes it might be feasible to ask the employees to rate the perceived quality after each interaction with the system [13]. This is not applicable for business applications, which are permanently used. Therefore, [3] introduced a non-intrusive survey tool, which pulls ratings via hourly requests from the employees. Further, the authors introduced a first model to derive user satisfaction from technical data collected in the wild [4]. However, the introduced pull rating system leads to coarse-grained performance assessments by only collecting ratings once an hour, which is still time consuming from the employees' point of view.

A more fine-grained view on the QoE may be achieved by collecting only self-motivated ratings comparable to complaint systems, where users can report bugs and malfunctions of software or services. In a certain way complaints may be related to QoE ratings [14], even if it is difficult to map them to the often used five-point QoE rating scale [15]. However, there is a relationship between user complaints and technical performance parameters as demonstrated for an IPTV service [14]. Besides the correlation of the subjective and technical data, the motivation of providing ratings is important. While other self-motivated feedback systems, e.g., product reviews, lead to a visible outcome to the feedback provider, reporting performance issues often has no direct benefit to the users. Thus, the motivation of the users differs depending on the perceived usefulness of the ratings. The analysis of user behavior with an integrated error reporting system by Microsoft, Inc. showed that the perceived usefulness is affected by the transparency of the data usage and transparency concerning the role of the users [16]. Further, the motivational factor may change over time [17]. Here, the authors investigated the activity and retention of volunteers participating in online studies.

Based on these findings, this work does not only investigates whether self-motivated QoE ratings can be used for QoE modeling, it also analyzes the motivation of employees to provide ratings over a long time period of 1.5 years.

III. Methodology and Data Set

A. Tool Description

The gathering of performance ratings about business applications in enterprise environments leads to several challenges and requirements. Most importantly, the gathering should not interrupt critical business processes. Further, it should not affect the performance of the business application. Therefore, the user study is conducted with a non-intrusive survey tool which fulfills various requirements specified by the cooperating company. With this tool, performance ratings were collected via a short survey in two steps. First, the user is asked to rate if s/he is satisfied or unsatisfied with the application performance on a two-point scale. In the second step, independent of whether the user was satisfied or not, the survey asks to explain the rating by selecting one out of a set of reasons, predefined by experts from the cooperating company, or the option *other*.

Due to the huge number of employees working with the business application, it is not possible to let them all participate in the user study. Further, having the same, few users participating in the study over years is also not feasible. To solve this challenge, the tool automatically selects representative user samples, each invited for a study period of two weeks. The sample members are only able to participate in the study during the specified two weeks, further referred as one study cycle. Hence, every two weeks the group of participants changes. If an employee is not able or willing to participate in the study, the tool offers a sign out functionality. Thus, no further rating requests are sent to that user. A detailed description of the survey tool and the participant's acquisition is described in [3].

The survey tool is configured in two ways to collect ratings with the push and the pull approach. The pull approach represents the poll method commonly used in QoE user studies. The tool actively asks the users to rate the application performance once an hour by automatically opening the survey in a pop-up. As the employees work permanently with the business application, it is not possible to ask for a rating after every interaction with the application. If the employee does not react within a few seconds, the pop-up closes automatically, and these unanswered rating requests are saved as missed requests. The push system is realized by giving the employees the opportunity to open the survey by themselves at any time. Therefore, a tray icon is integrated in the task bar of the operating system to make it as easy as possible to provide ratings. Again, if no rating is submitted within a few seconds, the pop-up survey window closes automatically.

B. Monitoring Technical Parameters

The performance of the business application is monitored with an internal monitoring system. This system records technical parameters about the executed transactions, such as type, total response time including processing time on the server, delay for traversing the network, information on the number and size of the transferred packets for the different transactions, as well as error flags and the number of current users in the system. Due to the huge amount of monitored transactions, the data is aggregated in five minute intervals. Thus, in a data point, the parameters for one user and for one type of transaction within a five minute interval are averaged.

C. Data Set

The data set contains technical data and user ratings collected between mid-November 2017 and mid-August 2019. Two study cycles have been excluded due to technical issues with the survey tool. Additionally, all data from regular holidays and days with incomplete technical monitoring data have been removed from the data set. Thus, overall the data set includes the data from 43 study cycles with a total of 4,433 participants. Of those 4,320 participants provide 389,105 ratings via the pull approach and 28,632 ratings were collected with the push approach from 3,207 users.

The technical data set contains 70,651,831 entries, considering only core working times. For the evaluation, the data was again aggregated per user both for all transaction types as well as only for the top 20 transaction types, which were performed within a five minute interval. These aggregations included the computation of minimum, maximum, mean, and different quantiles, which resulted into 74 descriptive metrics per user per five minute interval. In total, the data set contains the technical parameters of 3,965,516 intervals.

IV. Differences of Approaches

A. Analysis of Users' Motivation

As the participants' motivation may change over time, first, the response rates of the invited groups of employees are investigated. The user groups per study cycle were selected by random, hence, 77.7% of the employees take part in more than one cycle. This may lead to a decreasing motivation over time, especially for the push system. For the analysis, only users with at least two pull requests during a study cycle are considered. This excludes users who sign out from the study after the first rating request. For the remaining users, the response rate of the pull approach is defined as the ratio of given ratings and all requested ratings including the missed ones. The resulting response rate is on an hourly basis and considers only time spans in which the users were logged in the system. The response rate for ratings collected with the push approach is defined as the ratio of working hours containing ratings and those hours, in which the users were logged in the system but did not submit a rating. Again, log-in times are derived from the monitored pull requests.

Starting with a comparison of the response rates per study cycle, a trend toward decreasing rates can be observed over time. Indicated by a simple linear regression the decrease of the response rates is larger for the pull approach than for the push system. To further understand this effect, the response rate of the participants during each study cycle is analyzed. Therefore, the response rate for each day of a cycle is computed. While we found no evidence for a decrease of motivation when using the pull system, there is a decrease in the response rate of the push system during the two weeks of a cycle. This observation is established by a strong, negative correlation ($\rho = -0.964$) between the response rate and the day of the study cycle. Thus, the participation does not only decrease between the study cycles, but also within the cycles. To evaluate the influence of multiple participations, the change of the response rate between the first and consecutive participations is evaluated. The analysis is limited to 678 employees who participate at least in five study cycles.

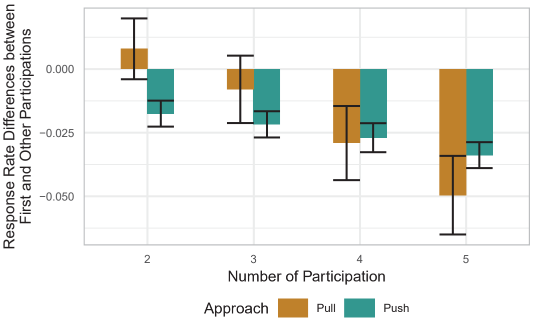


Fig. 1: Mean differences between the response rate of the first participation cycle and the response rates of the second up to fifth participation with 95% confidence intervals.

Figure 1 shows the average difference between the response rate of the first and the following four participations with 95% confidence intervals. As assumed, the response rates decrease for both approaches over time. While the rate is stable for the second participation in case of pull requests and then starts to decrease, the response rate continuously decreases after the first participation for the push approach. The significance of the differences in the response rates is established by Friedman's test for repeated measurements for the pull approach ($\chi^2(4) = 93.223$, $p < 0.001$) and the push system ($\chi^2(4) = 526.25$, $p < 0.001$). It explains the effect that the response rate decreases during the study period of 1.5 year. Based on these results, it is still unclear if the motivation to participate will be continuously on the decline or if it will level out at a certain response rate.

B. Temporal Assessment of Rating Behavior

To investigate temporal rating characteristics, e.g., differences in the rating frequency, the inter-arrival times of the ratings from both systems are analyzed. Figure 2 shows the CDFs of the inter-arrival times of ratings arriving in each system, both independent and dependent of individual users.

The user-independent inter-arrival time of pull ratings is short with a mean of 0.79 minute. 99% of the ratings have an inter-arrival time of less than 6 minutes. In contrast, the mean inter-arrival time of user-independent push ratings is higher with 8.35 minutes and the 99% quantile is about 69.58 minutes. This shows, that pull ratings arrive more frequent in the system than push ratings. The observation is in line with the higher response rate discussed in Section IV-A.

By having a closer look at the user-dependent inter-arrival times, the individual rating behavior of users who submit at least two ratings at the same day is analyzed. This includes 4,103 employees using the pull system multiple times a day, resulting in 239,983 inter-arrival times. The amount of users submitting multiple push ratings a day is lower with 1,401 employees and 10,435 inter-arrival times.

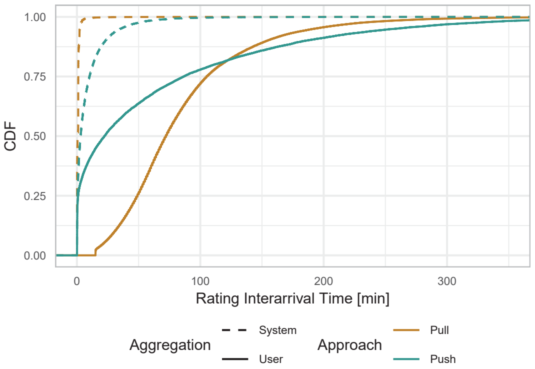


Fig. 2: Inter-arrival times of ratings arriving in the system independent and dependent from the users.

Self-motivated ratings occur more often with a shorter time interval than pull ratings, indicated by the solid, green curve which is located to the left of the solid, brown curve of pull inter-arrival times. Further, about 13% of the push ratings occur within 1 minute after another rating. In contrast, the shape of the CDF of pull rating arrivals is nearly linear within a time span of 10 to 120 minutes. This behavior is caused by the configuration of the pull system to collect ratings uniformly random once per hour. Overall, the data confirms that the pull ratings provide a continuous view on the perceived quality of a user, while push ratings give a more concentrated, detailed view on short time scales.

C. Analysis of Rating Opinions

As the users' opinion derived from the ratings may differ between the approaches, the share of negative ratings is analyzed. While only 17.9% of the gathered pull ratings are negative, the share of unsatisfied push ratings is higher with about 88.7%. Other than expected, the employees use the rating tool not only to report performance issues. However, the trend toward a complaint tool cannot be denied. Reasons for that may be that users are more motivated to report performance issues than stating that everything works fine. An indicator for this hypothesis is the decreasing response rate during the study cycles observed for the push system. To establish the hypothesis, the changes in the push rating behavior are evaluated with respect to the share of unsatisfied responses during each study cycle. The evaluation is based on the activity of the participants, meaning that each day with a push or pull rating is classified as an active day for that user. This allows to compute the share of negative push ratings per active day for all users.

Figure 3 shows the mean deviation of the unsatisfied share from the mean share of unsatisfied ratings per cycle including 95% confidence intervals. On the first two active days, the mean deviation is negative, meaning that more satisfied ratings arrive than on average. On the third day, the share is nearly equal to the average share of negative ratings of the cycles. Day four to ten deviates positively with a higher share of negative responses than on average. The significance of the differences in the deviation is revealed by Friedman's test for repeated measurements ($\chi^2(9) = 92.827$, $p < 0.001$). A pairwise comparison using Nemenyi's multiple comparison test reveals that the share of unsatisfied ratings is significant lower on the first and second active day than on all other active days ($p < 0.01$). There are also significant effects when comparing day 3 with day 7 to 10 ($p < 0.05$), while the average behavior on the other days does not differ significantly ($p > 0.05$). This results corroborate the assumption that the motivation to provide satisfied push ratings decreases over time. Running cycles with a duration of more than two weeks would converge the push approach to a pure complaint system. Further, we observe that the user gave more specific feedback about the affected system components in the second step of the survey. A chi-squared test establishes that the distributions of the selected reasons differ significantly ($\chi^2(6) = 3,266.2$, $p < 0.001$). With the push approach, the user less often select the option *others* (push: 9%, pull: 28%).

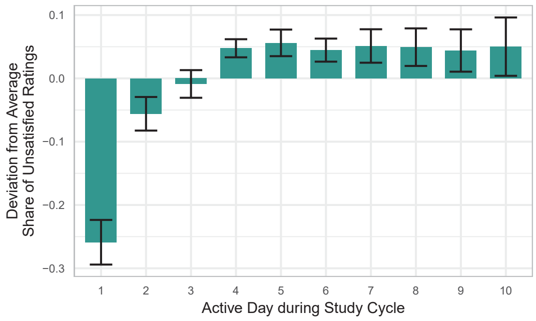


Fig. 3: Deviation from average share of unsatisfied ratings

D. Discussion

To sum up, with a decreasing motivation to provide positive push ratings, the self-motivated ratings converge towards a complaint system. However, the push approach provides a more specific understanding of negative ratings than the pull approach. If employees are motivated to rate the performance, their ratings occur often in short time spans. In contrast, the view on the users' QoE with pull ratings is more steady, but also coarser-grained per user.

A limitation of the study may result from using both systems in parallel leading to an unintended influence of the pull on the push system. For example, a missed, but noticed pull request might result in a push rating later on. However, only 4% of the push ratings occur within 5 minutes after a missed pull request. As it is unclear if this is caused by chance, this effect is neglected in the further evaluations. Nevertheless, the analysis of influence effects brings another phenomenon to light. 4% of the push ratings occur within a distance of one minute after an answered pull rating. Reasons may be the intention to correct a given rating or to add an additional reason for the pull rating. Hence, these ratings are excluded from the further evaluation.

V. QoE Model

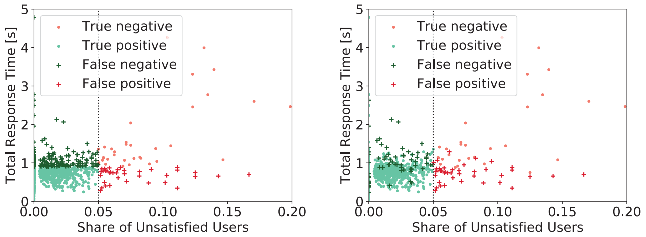
First of all, we investigate the correlations between ratings collected with pull requests and push approach aggregated per day. Note that we limit this analysis to days with at least 10 ratings gathered with each approach. Considering Spearman's rank correlation ρ between the daily share of unsatisfied ratings from both approaches, a significant positive correlation $\rho = 0.433$ can be observed. Supporting the hypothesis that the employees use the push approach similar to a complaint system, we can increase the correlation to $\rho = 0.592$ by interpreting hours without any push rating as time slots where the users are satisfied. This suggests that, in contrast to the pull approach, the evaluation of the push approach should focus on the negative ratings, which might point to annoying or unacceptable performance of the enterprise application.

Therefore, we investigate the correlation between the technical data of the enterprise application and the self-motivated negative push ratings. We use the point-biserial correlation coefficient between the Boolean indicators whether a negative push rating was given in a five minute interval and the performance metrics. All correlation coefficients for the technical parameters are close to 0. As the correlations are so low, which was expected, we aggregate the technical data and rating data into intervals of one hour. When considering these 5,433 aggregated intervals, the highest correlations can be observed for the mean of the minimum server processing time. It has a significant positive correlation to the share of unsatisfied users ($\rho = 0.384$), while other typical technical parameters, such as the mean of the total response time ($\rho = 0.282$), show a lower correlation. Due to the characteristics of the push-based approach, which includes short inter-arrival times (cf. Figure 2), we will not aggregate the data further to not lose or average out the temporal proximity of system performance and submitted push ratings.

A. Threshold-based QoE Model

As [4] successfully applied a threshold-based model to estimate the share of satisfied users, a similar model is developed for the push approach focusing on the relative number of unsatisfied users within one hour. The target threshold to distinguish an interval with a good application performance from intervals with a bad performance was set to 5% of the users. This means, the performance of the enterprise application is considered to be bad if more than 5% of the currently active users submit a negative push rating.

Similar to [4], two thresholds are fitted to the mean total response time based on two criteria. Namely, the balanced accuracy for both classes shall be maximized, and the number of intervals that cannot be classified, i.e., intervals whose technical parameter lies in between the thresholds, shall be minimized. When fitting the model to the data and optimizing for balanced accuracy in the first place, the two thresholds fall to the same value, which obviously also optimizes the second criterion. The resulting threshold resides at a mean total response time of 900ms with a balanced accuracy of 0.67. The overall accuracy is 0.81, and the corresponding per-class accuracy values lie at 0.84 (good QoE) and 0.51 (bad QoE). In the following, machine learning (ML) is applied to develop a ML-based model, which outperforms the threshold model.



(a) Threshold-based model.

(b) ML-based Model.

Fig. 4: Performance of push-based QoE models on test set.

B. ML-based QoE Model

To develop ML models for the data of the enterprise application, a Python-based Scikit-learn pipeline was used. The features were comprised of the 74 technical parameters, and the number of actively working participants in an interval, which gives an indication of the overall system load. First, the 5,433 one hour intervals are randomly split into a training set of 80% of the data, and a test set of 20% of the data. As the class distribution (good/bad QoE) is very imbalanced, the training data were upsampled to reach an equal number of instances per class. Several feature subsets, ML algorithms, and hyperparameters were tested with a 3-fold cross-validation on the training set to select the best features, model, and model parameters. The best performing model was a Gradient Boosting Classifier with 200 regression trees using 50 of the 75 features. Its performance was then tested on the test set of 1087 intervals. Here, the prediction accuracy of the good QoE class (1014 intervals), which is also the recall, is 0.92. The precision is 0.96, which gives an F_1-score of 0.94. For the minority class (bad QoE, 73 intervals), the precision is 0.31, the accuracy/recall is 0.53, and the F_1-score is 0.40. Thus, the performance of the ML-based model is better than the threshold-based model, reaching better per-class accuracy values, a better balanced accuracy of 0.73, and a better overall accuracy of 0.89.

Figure 4 visualizes the performance of both the threshold-based and the ML-based model on the test set. The x-axis shows the share of unsatisfied users including the black QoE threshold at 5%, while the y-axis shows the mean total response time. Light green and light red colored dots indicate correct estimations of good or bad QoE, respectively. However, the pluses indicate wrong estimation, namely, false positives (green plus) and false negatives (red plus). In Figure 4a, it can be seen that the threshold-based model separates the data horizontally. All intervals, which lie in the top-left sector are false negatives, erroneously classified as intervals with bad QoE. Diagonally opposite are the false positives, which are classified as intervals with good QoE although they have more than 5% of users, which submit negative push ratings.

Next to it, in Figure 4b, it can be seen that the ML-based model overall performs better, which is expected as it is not limited to a single technical parameter. This is especially evident in the good QoE case, where almost all intervals are correctly classified. Moreover, it can be seen that the performance is also good for high mean total response times. Only in case this technical parameter is low, the model loses a lot of its discriminative power, and misclassifies several intervals into false positives. An important observation is that many of these intervals have very low values of the technical parameters, i.e., an objectively fast performance of the enterprise system. This suggests that, in these cases, possibly the technical system was not (only) responsible for the negative push rating, but maybe other work-related issues triggered the rating.

To sum up, both the threshold- and the ML-based model allow to map the self-motivated push ratings onto QoE. As the threshold-based model only considers a single technical parameter, its decision boundary introduces a lot of false positive and false negative classifications. In contrast, the ML-based QoE model uses all technical parameters, and consequently, can significantly reduce the false negatives. However, due to the class imbalance and possibly other non-technical issues, which are not contained in the data, the performance on the bad QoE class is lower than on the good QoE class. Thus, it was shown that it is also possible for the push-based approach to model the QoE both with simple threshold-based and more complex ML-based models.

VI. Conclusion

In this work, a push approach for collecting self-motivated assessments of the performance of business applications is introduced and it is evaluated in a large, long-term user study. As employees have to trigger a quality rating themselves, the system is relevant for the enterprise environment, reducing the time needed to rate compared to the participation in regularly polled QoE ratings. The analysis of differences in the characteristics between this new approach and the commonly used pull system shows that the rating behavior differs. We found evidence that there is a trend towards a pure complaint tool, based on the much higher share of poor performance ratings from the push system. In addition, the motivation to give satisfied ratings decreases over time within and between study cycles. Nevertheless, in case the employees trigger a performance rating, these push ratings arrive more frequently within short time scales. The temporal proximity of system performance and submitted push ratings leads to a finer-grained view on the performance of the business application compared to the more steady, but coarse-grained pull system.

Moreover, the push approach is equally well suited to derive QoE models for the performance of the business application. Both simple threshold-based and complex ML-models could be successfully applied to the data. Here, it could be observed that the ML-based model, which could consider more technical parameters, outperformed the simple threshold-based model. An interesting observation can be made that intervals with an objectively fast performance of the enterprise system had high ratios of unsatisfied users and were misclassified. This suggests that, in these cases, possibly the technical system was not (only) responsible for the negative push rating, but maybe other work-related issues triggered the rating.

Acknowledgment

This work is supported by kubus IT. The authors express their gratitude to André Rentsch and Peter Karg for their support. The authors alone are responsible for the content.

References

[1] P. Le Callet, S. Möller, and A. Perkis, Eds., *Qualinet White Paper on Definitions of Quality of Experience*. European Network on Quality of Experience in Multimedia Systems and Services, 2012.

[2] T. Hoßfeld, R. Schatz, M. Varela, and C. Timmerer, "Challenges of QoE Management for Cloud Applications," *IEEE Communications Magazine*, vol. 50, no. 4, pp. 28–36, 2012.

[3] K. Borchert, M. Hirth, T. Zinner, and A. Göritz, "Designing a Survey Tool for Monitoring Enterprise QoE," in *Workshop on QoE-based Analysis and Management of Data Communication Networks*. ACM, 2017.

[4] K. Borchert, S. Lange, T. Zinner, and M. Hirth, "Identification of Delay Thresholds Representing the Perceived Quality of Enterprise Applications," in *10th Intl. Conf. on Quality of Multimedia Experience (QoMEX)*. IEEE, 2018.

[5] A. Sackl, P. Casas, R. Schatz, L. Janowski, and R. Irmer, "Quantifying the Impact of Network Bandwidth Fluctuations and Outages on Web QoE," in *7th Intl. Workshop on Quality of Multimedia Experience (QoMEX)*. IEEE, 2015.

[6] S. Ickin, K. Wac, M. Fiedler, L. Janowski, J.-H. Hong, and A. K. Dey, "Factors Influencing Quality of Experience of Commonly Used Mobile Applications," *Communications Magazine*, vol. 50, no. 4, pp. 48–56, 2012.

[7] S. Baraković and L. Skorin-Kapov, "Survey of Research on Quality of Experience Modelling for Web Browsing," *Quality and User Experience*, vol. 2, no. 1, p. 6, 2017.

[8] I. Orsolic, M. Suznjevic, and L. Skorin-Kapov, "Youtube QoE Estimation from Encrypted Traffic: Comparison of Test Methodologies and Machine Learning based Models," in *10th Intl. Conf. on Quality of Multimedia Experience (QoMEX)*. IEEE, 2018.

[9] T. Hoßfeld, M. Varela, P. E. Heegaard, and L. Skorin-Kapov, "QoE Analysis of the Setup of Different Internet Services for FIFO Server Systems," in *Intl. Conf. on Measurement, Modelling and Evaluation of Computing Systems*. Springer, 2018.

[10] W. Bonhag, D. Feindt, S. Olschner, and U. Schubert, "Wie schnell ist "schnell" bei Business-Software? Analyse zur Performance bei der Nutzung von Business-Software," *Mensch und Computer – Usability Professionals*, 2015.

[11] A. Mockus, P. Zhang, and P. L. Li, "Predictors of Customer Perceived Software Quality," in *27th Intl. Conference on Software Engineering (ICSE)*. IEEE, 2005.

[12] T. Zinner, F. Lemmerich, S. Schwarzmann, M. Hirth, P. Karg, and A. Hotho, "Text Categorization for Deriving the Application Quality in Enterprises Using Ticketing Systems," in *Intl. Conf. on Big Data Analytics and Knowledge Discovery*. Springer, 2015.

[13] R. Smith and L. A. Kilty, "Crowdsourcing and Gamification of Enterprise Meeting Software Quality," in *7th Intl. Conf. on Utility and Cloud Computing*. IEEE, 2014.

[14] X. Wei, Z. Li, R. Liu, and L. Zhou, "IPTV User's Complaint Prediction based on the Gaussian Mixture Model for Imbalanced Dataset," *Journal of Computers*, vol. 28, no. 6, pp. 216–224, 2017.

[15] L. Wang, J. Jin, R. Huang, X. Wei, and J. Chen, "Unbiased Decision Tree Model for User's QoE in Imbalanced Dataset," in *Intl. Conf. on Cloud Computing Research and Innovations (ICCCRI)*. IEEE, 2016.

[16] K. A. Saeed and A. Muthitacharoen, "To Send or not to Send: An Empirical Assessment of Error Reporting Behavior," *Transactions on Engineering Management*, vol. 55, no. 3, pp. 455–467, 2008.

[17] J. Cox, E. Y. Oh, B. Simmons, G. Graham, A. Greenhill, C. Lintott, K. Masters, and J. Woodcock, "Doing Good Online: The Changing Relationships between Motivations, Activity, and Retention among Online Volunteers," *Nonprofit and Voluntary Sector Quarterly*, vol. 47, no. 5, pp. 1031–1056, 2018.