QoEXplainer: Mediating Explainable Quality of Experience Models with Large Language Models

Nikolas Wehner*°, Nils Feldhus†°, Michael Seufert‡, Sebastian Möller†§, Tobias Hoßfeld*
*University of Würzburg, Würzburg, Germany
†German Research Center for Artificial Intelligence (DFKI), Germany
‡University of Augsburg, Augsburg, Germany
§Technical University of Berlin, Berlin, Germany

***Abstract*—In this paper, we present QoEXplainer, a QoE dashboard for supporting humans in understanding the internals of an explainable, data-driven Quality of Experience model. This tool leverages Large Language Models and the concept of Mediators to convey relevant explanations to the user in an understandable, chatbot-like fashion. For this purpose, our tool QoEXplainer integrates a data-driven video streaming QoE model and techniques from Explainable Artificial Intelligence. The resulting data-driven model explanations are illustrated in the dashboard and users can interact with the chatbot to ask questions about the data and QoE model and control the dashboard to enhance model understanding. With this hybrid demo, we aim to conduct a live study at QoMEX 2024 to evaluate Mediators in the context of (data-driven) QoE modelling with domain experts.**

***Index Terms*—Quality of Experience; Explainability; Mediators; Large Language Models**

I. Introduction

With the recent surge of advances in Artificial Intelligence (AI), in particular in the fields of Generative AI due to Large Language Models (LLM), AI has become ubiquitous in many domains, e.g., natural language processing, computer vision, speech, networking, drug discovery, finances, or marketing [1]–[3]. Despite these advances, current AI systems have not yet reached a satisfactory performance in tasks related to Quality of Experience (QoE) of services [4] as still required by service and network providers to this day. Nevertheless, these developments in AI also pave the way for novel tools, which may effectively support providers in enhancing this understanding. Combining the best world of both QoE modelling and AI, QoE modelling with Explainable AI (XAI) has been recently proposed as a general concept to model the QoE of arbitrary applications and to adapt the model automatically over time in a data-driven fashion [5].

While the explanations provided by an XAI-based QoE model may help AI experts or people with domain knowledge to better understand the model, these explanations may sometimes still be difficult to grasp for both experts and, specifically, non-experts. One approach to improve the understanding of these explanations might be the use of Generative AI in the form of Mediators [6], [7]. Mediators allow an LLM to explain a model's internals, datasets, or overall behavior to the end-user in a human-friendly way. As the mediation of (explainable) QoE models by an LLM has not been investigated yet, we frame the following research question in this work: *Are Large Language Models capable of supporting humans in interpreting (explainable) QoE models?*

In the light of this research question, we develop QoEXplainer, a dashboard, which offers information on the internals of an XAI-based QoE model, and also integrates a chatbot that acts as the Mediator. With this dashboard, we aim to conduct multiple studies to answer the posed research question and conduct one of these studies as a live study during the demo session of QoMEX 2024. This allows us to obtain valuable feedback from several domain experts simultaneously.

II. Background and Related Work

A. XAI: Explainable Artificial Intelligence

With the recent advances in the field of XAI, it has become a relevant topic for all kinds of research areas. In [2], the authors survey the current state-of-the-art in XAI and introduce various concepts. In QoEXplainer, we include the Neural Additive Model (NAM) [8], an interpretable additive model based on neural networks, and SHAP [9], a widely used post-hoc explainer for black-box models based on game theory. Composing different XAI techniques and models into a single interface has already been investigated in literature. Existing XAI interfaces include Gamut [10], LIT [11], and WebSHAP [12]. While their interactivity is limited to opening and closing of views and charts, the main advantage of QoEXplainer is the natural language interface lowering the level of necessary expertise and allowing for more diverse requests.

B. Mediators

Framing XAI as a conversation between the user and the explanatory system has been gaining more attention recently. Exemplary implementations of explanation dialogue systems are TalkToModel [13], ConvXAI [14], and InterroLang [7]. The concept of a Mediator in this context is the LLM chatbot conveying the relevant explanations of the underlying model's behavior to the user in an understandable way [6].

Our version of a Mediator is specifically designed to explain QoE models like NAMs [5], [8] using feature attribution and dataset analyses. Our QoEXplainer is different from the

°Equal contribution

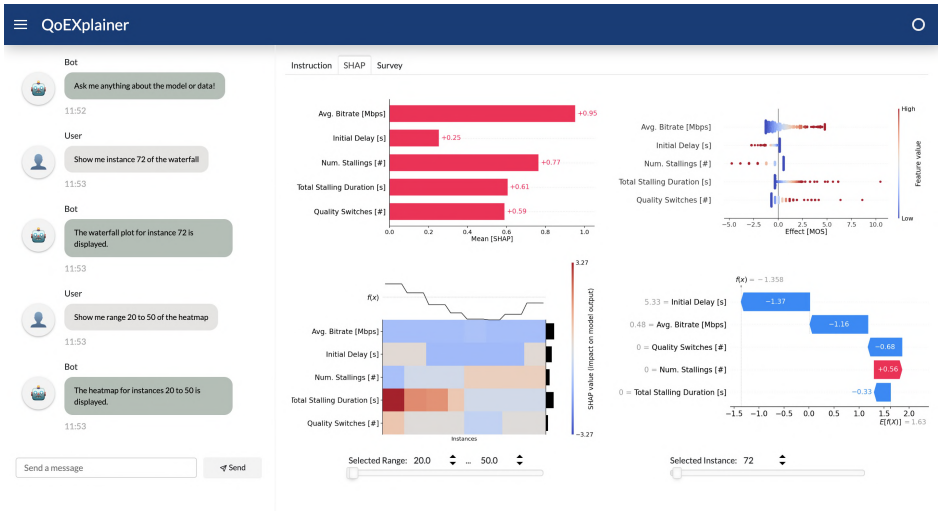

Fig. 1: QoEXplainer dashboard showing an example dialogue of user requests (left) and SHAP explanation diagrams (right).

existing systems in that all explanations are already pre-computed and stored in a Pandas DataFrame instead of having to compute explanations on-the-fly and relying on dedicated Python functions to retrieve such explanations.

Translating natural language questions into executable commands is an active area of research. While most works have focused on Text-to-SQL tasks, other works started to analyze the capabilities of LLMs handling Pandas query prompts that include information about the DataFrame in question [15].

III. QoEXplainer

A. Use Case

As exemplary use case for QoEXplainer, we consider the same video streaming QoE setting as in [5], as video streaming (QoE) is usually well-understood by domain experts. The training data for the data-driven QoE models thus consists of five different databases [16]–[20], resulting in a highly heterogeneous training dataset. We consider the five expert features average bitrate, initial delay, number of stalling events, total stalling duration, and the number of quality switches for training the QoE model. We then integrate this training data, the trained NAM from [5], and the computed SHAP values for this model into the dashboard.

B. Dashboard

For building the frontend of QoEXplainer, we use the Python library Panel from the HoloViz ecosystem [21]. A screenshot of the dashboard is depicted in Figure 1. The dashboard consists of a main view with different tabs and a side bar for the LLM-based chatbot. In total, the main view offers five views with different functionalities:

The *Instruction* tab provides instructions and explanations to the user in customizable Markdown. The *Model* tab is composed of five figures, which display the learnt functions of the NAM-based QoE model, along with the current MOS prediction in the form of an equation on top of the figures. Users can interact with the figures to observe how the MOS changes with different feature values. The *Training Data* tab allows the user to interact with the training data used for learning the QoE model. Users can inspect the training samples by sorting the samples based on the features or the MOS. In the *SHAP* tab, four figures display different aspects of the computed SHAP values. There are two non-interactive figures on top, which comprise a global feature importance plot and a sample-based beeswarm summary plot. On the bottom there are two interactive figures, which show a heatmap explaining how SHAP values and feature values are distributed across the selected samples and what the model outputs, and a sample summary plot explaining how SHAP derives the model's output for a single sample. Below there are widgets to configure the currently inspected (range of) sample(s). Finally, the *Survey* tab provides the questionnaires.

The chatbot in the side bar is implemented in the style of other chatbot services like OpenAI's ChatGPT[1]. In a chat box, users can enter their prompts, and on submit an answer by the chatbot follows. The messages of user and chatbot are clearly distinguishable due to different colors, icons, and names. Additionally, to indicate "thinking" by the chatbot, the chatbot's answers are streamed asynchronously.

Holoviz's Panel runs server-sided, i.e., multiple users can simultaneously use the dashboard. To be able to distinguish between users, we assign each session a random ID (the current timestamp plus forty random bits) during startup, thus also ensuring anonymity. Using this random ID we create a log file on server-side, which is updated continuously, whenever a user interacts with the dashboard in any form.

C. LLM-based Exploration of Data and Explanations

We adapt the Pandas DataFrame toolkit[2] in LangChain for repeated language model inference related to data stored in a Pandas DataFrame[3]. It is a wrapper around a Python agent,

[1] https://chat.openai.com/
[2] https://python.langchain.com/docs/integrations/toolkits/pandas
[3] https://pandas.pydata.org/docs/reference/frame.html

a Pandas DataFrame and an LLM. The DataFrame contains the original values of the data (average bitrate, initial delay, number of stallings, total stalling duration, quality switches) and SHAP explanations [9] for each of them. The info about the columns and data types is provided to the LLM via the prompt. The Python agent takes care of calls to either an open-source LLM (e.g., hosted on Hugging Face[4]; both locally stored models and API-accessible models are possible) or a closed-source LLM such as ChatGPT via its API[5] and executes the Python code synthesized by the LLM on the provided DataFrame. The result of the Pandas command is then returned to the user interface as the final response.

D. LLM-based Interaction with Dashboard

We also allow the agent to call custom tools[6] which directly modify dashboard views, e.g. showing specific samples in the SHAP waterfall plot or a range of samples in the SHAP heatmap plot. Additionally, the user's natural language commands can control the LLM to call tools connected to the *Training Data* tab, e.g., showing specific samples in the table or sorting the table according to one of the given features (or column names), or the *Model* tab, e.g., modifying feature values and changing the visualization accordingly.

E. Study Design

We design the study in a fashion that users are guided through the tasks consecutively. A task always ends with filling out the questionnaires belonging to the current task. In detail, after the dashboard has been loaded, the user is first shown the *Instruction* tab which provides information on the current task. To limit the affordance, we explicitly show only the tabs required to solve the task. A user can finish the current task by changing to the *Survey* tab, which asks the user questions tailored towards the current task. After the survey for the current task has been submitted, the user is again redirected to the *Instruction* tab, where a new set of instructions is displayed. This also means that the *Instruction* tab and the *Survey* tab are always shown for every task. After the survey to the final task has been completed, the user has to fill a final survey on the overall experience and on demographic data.

1) Tasks: In total, our study consists of three tasks and the total study takes around ten minutes. For each task, users are supposed to use the help of the chatbot. The first task evolves around general data understanding. Here, users are shown the *Training Data* tab to learn about the dataset and are supposed to answer specific questions, e.g., "How many samples does the training data consist of?", "How many samples contain more than five stalling events?", or "How many samples have more than 12 quality switches?". Additionally, users are supposed to compute specific statistics for features, e.g., the mean or the 25th percentile of a feature. In the second task, participants use the *Model* tab to get a general model-based understanding. Here, we ask what-if questions, e.g., "How does the MOS change if the average bitrate drops below 1 Mbps?", or "How does the MOS change if we increase the number of stallings from one to six stalling events?". The last task makes use of the *SHAP* tab and considers general attribution-based understanding. For this purpose, users are supposed to answer which feature influences the MOS prediction the most and how the number of quality changes affects the MOS when we consider one specific sample.

2) Questionnaires: After each task, we ask the participants similar, but task-oriented questions, i.e., rating the helpfulness of the individual figures and the quality of the answers generated by the chatbot on a five-level Likert scale. After all tasks have been completed, we ask participants for general demographic information, e.g., age, gender, and education. We then use the NASA-TLX [22] to assess the task load, which we will use later in another study to evaluate the helpfulness of the chatbot. Finally, participants are asked about their overall dashboard and chatbot experience on a five-level Likert scale.

IV. Demo

The goal of the demo is to acquire as much feedback from domain experts as possible, such that we can later compare the QoE of experts and non-experts with respect to overall and chatbot experience, and usefulness of XAI techniques.

We design our demo as a hybrid experience demo. We make the dashboard publicly available, such that remote users can simply start the demo themselves by browsing to the offered URL. On-site participants are also free to use their own device or they can use the single device in the presentation space. There is no other requirement for remote users or local users to access the demo except for having access to an Internet browser. The initial loading time of the dashboard may vary depending on the available Internet connection, though. To the best of our knowledge, there is no limitation on how many users can simultaneously participate. During the study there is, however, no social interaction between local and remote users and bystanders. Ideally, bystanders are also not able to glimpse into the on-site device to avoid biases, while another expert participates in the study. Otherwise, there are no additional requirements for the presentation space.

V. Conclusion

In this work, we introduced QoEXplainer, a state-of-the-art QoE dashboard, which aims to help humans understand XAI-based QoE models by incorporating Large Language Models (LLM). The LLM in QoEXplainer uses the concept of Mediators to convey different aspects of the QoE model's internals in an easy-to-understand fashion to the user. Since this topic has not been researched yet in the context of QoE modelling, we framed a single research question which we would like to answer in future studies with QoEXplainer. With this dashboard, we plan to conduct a hybrid experience demo at QoMEX 2024 as this session would allow us to collect expert feedback on various topics, e.g., if LLMs are already useful for domain experts and which kind of data is especially helpful in understanding a data-driven QoE model.

[4]https://huggingface.co/models

[5]https://platform.openai.com/docs/api-reference/introduction

[6]https://python.langchain.com/docs/modules/agents/tools/custom_tools

References

[1] R. Gozalo-Brizuela and E. C. Garrido-Merchán, "A survey of generative ai applications," *arXiv preprint arXiv:2306.02781*, 2023.

[2] A. B. Arrieta, N. Díaz-Rodríguez *et al.*, "Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges Toward Responsible AI," *Information fusion*, 2020.

[3] H. Afifi, S. Pochaba, A. Boltres, D. Laniewski, J. Haberer, L. Paeleke, R. Poorzare, D. Stolpmann, N. Wehner, A. Redder, E. Samikwa, and M. Seufert, "Machine Learning with Computer Networks: Techniques, Datasets and Models," *IEEE Access*, 2024.

[4] K. Brunnström, S. A. Beker, K. De Moor, A. Dooms, S. Egger, M.-N. Garcia, T. Hossfeld, S. Jumisko-Pyykkö, C. Keimel, M.-C. Larabi *et al.*, "Qualinet White Paper on Definitions of Quality of Experience," 2013.

[5] N. Wehner, A. Seufert, T. Hoßfeld, and M. Seufert, "Explainable data-driven qoe modelling with xai," in *2023 15th international conference on quality of multimedia experience (QoMEX)*. IEEE, 2023, pp. 7–12.

[6] N. Feldhus, A. M. Ravichandran, and S. Möller, "Mediators: Conversational agents explaining NLP model behavior," in *IJCAI 2022 - Workshop on Explainable Artificial Intelligence (XAI), Vienna, Austria*, R. Weber, O. Amir, and T. Miller, Eds. International Joint Conferences on Artificial Intelligence Organization, 7 2022. [Online]. Available: https://arxiv.org/abs/2206.06029

[7] N. Feldhus, Q. Wang, T. Anikina, S. Chopra, C. Oguz, and S. Möller, "InterroLang: Exploring NLP models and datasets through dialogue-based explanations," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 5399–5421. [Online]. Available: https://aclanthology.org/2023.findings-emnlp.359

[8] R. Agarwal, L. Melnick, N. Frosst, X. Zhang, B. Lengerich, R. Caruana, and G. E. Hinton, "Neural additive models: Interpretable machine learning with neural nets," *Advances in neural information processing systems*, vol. 34, pp. 4699–4711, 2021.

[9] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4765–4774. [Online]. Available: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

[10] F. Hohman, A. Head, R. Caruana, R. DeLine, and S. M. Drucker, "Gamut: A design probe to understand how data scientists understand machine learning models," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1–13. [Online]. Available: https://doi.org/10.1145/3290605.3300809

[11] I. Tenney, J. Wexler, J. Bastings, T. Bolukbasi, A. Coenen, S. Gehrmann, E. Jiang, M. Pushkarna, C. Radebaugh, E. Reif, and A. Yuan, "The language interpretability tool: Extensible, interactive visualizations and analysis for NLP models," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 107–118. [Online]. Available: https://aclanthology.org/2020.emnlp-demos.15

[12] Z. J. Wang and D. H. Chau, "Webshap: Towards explaining any machine learning models anywhere," in *Companion Proceedings of the ACM Web Conference 2023*, ser. WWW '23 Companion. New York, NY, USA: Association for Computing Machinery, 2023, p. 262–266. [Online]. Available: https://doi.org/10.1145/3543873.3587362

[13] D. Slack, S. Krishna, H. Lakkaraju, and S. Singh, "Explaining machine learning models with interactive natural language conversations using TalkToModel," *Nature Machine Intelligence*, Jul 2023. [Online]. Available: https://doi.org/10.1038/s42256-023-00692-8

[14] H. Shen, C.-Y. Huang, T. Wu, and T.-H. K. Huang, "Convxai: Delivering heterogeneous ai explanations via conversations to support human-ai scientific writing," in *Companion Publication of the 2023 Conference on Computer Supported Cooperative Work and Social Computing*, 2023, pp. 384–387.

[15] J. Ye, M. Du, and G. Wang, "DataFrame QA: A universal LLM framework on dataframe question answering without data exposure," *arXiv*, vol. abs/2401.15463, 2024. [Online]. Available: https://arxiv.org/abs/2401.15463

[16] W. Robitza, S. Göring, A. Raake, D. Lindegren, G. Heikkilä, J. Gustafsson, P. List, B. Feiten, U. Wüstenhagen, M.-N. Garcia *et al.*, "HTTP Adaptive Streaming QoE Estimation with ITU-T Rec. P. 1203: Open Databases and Software," in *ACM MMSys*, 2018.

[17] Z. Duanmu, A. Rehman, and Z. Wang, "A Quality-of-Experience Database for Adaptive Video Streaming," *IEEE Trans. Broadcast*, 2018.

[18] Z. Duanmu, W. Liu, Z. Li, D. Chen, Z. Wang, Y. Wang, and W. Gao, "Assessing the Quality-of-experience of Adaptive Bitrate Video Streaming," *arXiv preprint arXiv:2008.08804*, 2020.

[19] C. G. Bampis, Z. Li, A. K. Moorthy, I. Katsavounidis, A. Aaron, and A. C. Bovik, "Study of Temporal Effects on Subjective Video Quality of Experience," *IEEE TIP*, 2017.

[20] C. G. Bampis, Z. Li, I. Katsavounidis, T.-Y. Huang, C. Ekanadham, and A. C. Bovik, "Towards Perceptually Optimized End-to-End Adaptive Video Streaming," *arXiv preprint arXiv:1808.03898*, 2018.

[21] S. Yang, M. S. Madsen, and J. A. Bednar, "HoloViz: Visualization and interactive dashboards in python," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 4846–4847. [Online]. Available: https://doi.org/10.1145/3534678.3542621

[22] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," in *Advances in psychology*. Elsevier, 1988, vol. 52, pp. 139–183.