From Static to Smart: Enhancing UI/UX Adaptation with Double Q- learning

Prof. Vishal Nayakwadi[1], Aarya Shendaware[2], Gaurav Tijare[3], Nimisha Tirpude[4], Abhijeet Uphade[5]
VIT Pune

Abstract — Currently, the majority of applications simply have a single interface that cannot be changed regardless of how individuals use the application. The common method, Q-Learning, can give things a strange or annoying fix, which RL can fix, as it will make things smarter and customized, yet it is overly optimistic. To solve this we implemented Double Q-Learning (DQL) to create an interface that actually changes. We have two AI evaluators, one suggests a change, and the other one verifies whether it is possible. This makes the interface continuously develop and prevents poor decisions. With users in real time; when they click on something, the duration of interaction with a feature, which feature to visit, our framework will continue to make minor adjustments to the layout, content, and priorities of features. We realized that this is the best way to make software significantly easier to interact with, to maintain user engagement and increase customer satisfaction. It performs excellently with websites, phones, and also assistive technology.

Keywords: **Double Q-Learning (DQL), Reinforcement Learning (RL), Human-Computer Interaction (HCI), Personality, Adaptive User Interface (AUI), User Experience (UX); Software engineering, intelligent systems, dynamic response of UI, and bias of overestimation.**

I.INTRODUCTION

Do you ever think that apps are meant to be used by someone? That's a pretty common vibe. Most of the software nowadays is one-size-fits all without considering the way everyone uses their gadgets. Its designs do not show any sensitivity to individual preferences, habits, and phone size thus making it irritating and difficult to read. The fix? A user-friendly interface that, in fact, fits you. The ancient, unchanging method also has a giant flaw, even when AI, particularly in reinforcement learning, can be used to assist. The AI that drives Qplus learning is over hyped and generates unrealistic options, thus the application rewires itself similar to a bad makeover, making you even more frustrated.

It is like a second opinion by the AI. The system proposes a modification in one component, such as the positioning of a button or rearranging of menu and verifies the utility of such a procedure in another component. This allows the rapid verification to ensure that the interface does not get cluttered and updates remain consistent and predictable. The system reads through what you tap, the speed at which you scroll and when you stop and thus it learns what actually works with you. We have demonstrated that DQL is a good method to customize software to make it entirely personal, a tremendous success with assistive technology, phone applications, and websites that have been created to look like they were built just to you.

II.LITERATURE REVIEW

Studies on how to make user interfaces (UI) and user experiences (UX) more engaging through numerous testing, fine-tuning, and deep design are enormous. There is a large number of individuals who are considering mobile UI testing. Similar to Salihu and colleagues [1] discussed AMOGA, a combination of static and dynamic tricks to debug UI models on Android applications. There is also another group [2] that created Nighthawk to scan the screen, and automatically identify and locate UI glitches. On the security aspect, Xiao et al. [3] proposed icon Intent which performs a static analysis and icon classification to identify sensitive UI widgets. In [5], Salihu also examined this, closer. A thorough research on the effect of UI implementation on Android applications, particularly on app crawlers, was carried out by Wan and friends [4]. Wang et al. [7] actually designed a CNN-based program named Textout to locate text layout errors in mobile applications.

Other researches borrowed what is already known and included adaptive interfaces. AUI was the first programming language to be introduced by Schneider and Cordy [20] to make interactive software, capable of changing based on different devices. Sboui et al. [23] later demonstrated a software product line strategy, DSPL, through which UIs can be modified during use when the platform, the environment, (or even your own preferences) cause changes. Dominguez et al. [25] came up with a universal adaptation model to assist fire up of UIs to provide a smooth media experience on all types of gadgets.

One of the main, ongoing issues in UI design is the creation of interfaces which can suit all of us, and adapt easily to every type of user interface situation, and user group. Having this information, numerous researchers magnified some individuals. Yun et al. [11] and Lim [12] took older adults as the subjects of their study. Their application of intelligent UI/UX systems that adjust themselves according to cognitive stimuli was meant to make smart phones easier to use by seniors, and it signifies a larger idea: the same cautious localization required by age also contributes to the issues of the region. A good example was provided by Budi et al. [13], who used Design Thinking to develop a mobile flood-info app in rural Indonesia- addressing digital literacy issues more directly on the local level.

These context-based systems must be concrete. Yalla et al. [14] have noted that Context-aware Adaptive Mobile Systems (CAAMS) require deep testing to ensure that they perform well in relation to the location, time and activity of a user. The concept of adaptive experience was also experimented on cars. As stated by Rittger et al. [10], various degrees of adaptiveness were elaborated and the importance of transparency to the drivers.

Reinforcement learning (RL) appears to be a potentially effective means of supporting all such adaptation. As demonstrated by Sun et al. [6] (and other Sun papers), the user experience itself (clickthroughs, retention rates, etc.) may be the primary training indication. The entire intention is to allow the system to learn what the users are in fact doing so that it can design interfaces that actually respond to them.

However, the only challenging task is not adapting, but also, maintaining privacy is enormous. The federated learning became a subject matter. Xu et al. [9] developed a system, FedABR, which applies federated RL, to personalize the video stream without collecting user information. Riahi et al. [22] extended that concept by developing a federated learning model that relies on blockchain to provide additional trust and security. Another major concern is efficiency. The researchers desire to retain privacy and at the same time make the learning process quicker. Wang et al. [21] addressed the issue of transitioning between an offline, pre-trained model to a live online one, which proposes a guidance model to aid in the process of enhancing the policy, which receives new information in real time.

We have Double Q - Learning (DQL), which is a significant way of reinforcement learning, in our own work. It is not mere theory but has already been demonstrated to come in handy in complicated areas, particularly in areas where maximizing resources is of concern. One might observe that it was applied in He et al. [15] who applied to manage the energy of a hybrid vehicle with the help of a DQL model and saved tons of fuel. Another example is the Industrial Internet of Things (IIoT): Qiu et al. [16] optimized the allocation of resources to blockchain systems with the help of a dueling deep Q-learning technique. Hammadi et al. [18] also applied a resource management approach based on the DQL to indoor VLC systems and achieved significant performance improvements.

Lastly, the emergence of new UI design and control tricks is coming up. Liebing et al. [19] provided a viable approach of aligning UI components and web service operations. Duan et al. [24] proposed a model to auto-create high-quality and personalized UI designs with text and sketches using diffusion models. The range of intelligent system design is demonstrated in other papers by Huang and Zheng [8] of the music-teaching system to Tam et al. [17] of the federated learning of image sensing.

III.METHODOLOGY

Given that this is a research that is less about conducting an actual experiment and more about developing a conceptual model, this section describes how we are going to configure our system. This is not the final version of our product but rather a description of the blueprint of a more reliable and intelligent adaptive system.

1. The Dilemma of Adaptation vs. Stability.

The biggest issue with most adaptive UIs is that they are too eager. They modify the interface to small, temporary user habits, thereby confusing the user and making the app unstable. This adds the cognitive load - users are forced to learn the interface again, becoming frustrated and quitting. We have a Cognitive Load Aware Framework (CLAF). It is all designed based on a single philosophy, which is that you should make a change in the UI only when the net benefit in the long term is more than the net confusion in the short-term.

This model is divided into two stages:

- Phase 1: The Heterogeneous Habituation Model (Offline): Trains the default behavior of all the users to develop a smart common sense.
- Phase 2: The Personalization Layer (Online): Gradually and cautiously tweaks that default to a single user, only when such a user exhibits a strong and steady preference.

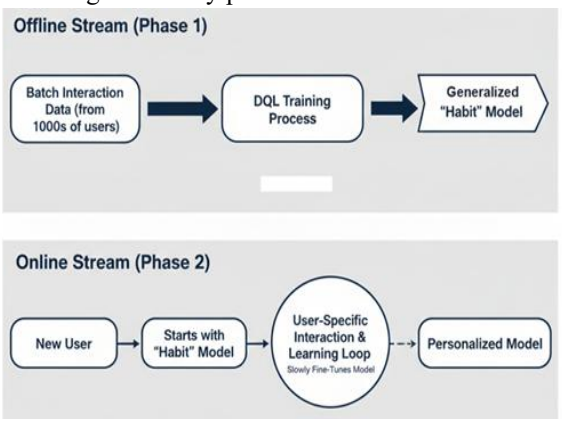

Fig. 1. The so-called Proposed Cognitive-Load-Aware Framework (CLAF) Workflow.

2. A.1 Proposed System Architecture

We have made the CLAF architecture to be modular. The technology stack that we would implement would be:

- Client (Frontend): The front end, a React application, which will display all of the UI elements (buttons, menus) based on a JSON object sent by the server, so that the interface can be changed without loading the page again.
- Backend (API): A node.js (express) server which serves 2 tasks:
  1. Ingestion: Events of interaction (clicks, scrolls, hovers) are sent to the React client.
  2. Adaptation: Keeps a WebSocket connection to send real time changes of the UI.
- Data Pipeline: A Redis stream is a high-performance message queue that is used to handle incoming interaction events without straining the server.
- Database and AI: A PostgreSQL storage contains logs of interaction between the user. The main DQL agent, which is a Python (PyTorch) program, is read from this database when training offline. The decisions of the trained model are served to the Node.js backend by a FastAPI server.

3. A.2 World (The MDP) of the AI

An AI needs to be taught in a Markov Decision Process (MDP), and this process requires defining its environment in a manner comprehensible to the AI.

- State (S): So what is the state of things? It is not only the page to which the user is on. We also suggest a rich state with: the existing UI design, the history of the user (e.g. just added to cart), and the type of a user (e.g. new user versus power user).
- Action (A): What can the AI do? The actions of the AI are small, discrete, they include: moveButtonXtoPosY, changeColorOfButtonZ, reorderMenu. The action of uttering nothing is the most significant one and it is central to our structure. The AI should be trained that when nothing is done, it can be the best option.
- Reward (R): What is the way to inform the AI that it has done a good job? This is the crux. We do not merely reward the completion of the task but rather, we have a Cognitive Load Penalty (CLP). The reward function is:

$$R_t = (\text{Task Success Bonus}) - (\text{Cognitive Load Penalty})$$

  - Task Success Bonus (e.g., +100 points): The AI receives a large reward upon achievement of the goal by the user (e.g. makes a purchase, finds information).
  - Penalty on Cognitive load (e.g., -5 points): The AI will be penalized a little whenever it makes any decision other than doNothing.

And the penny falls on the head. Consider it: the AI is aware of the fact that it will lose 5 points regardless of whether it tries or not. That's a fixed cost. It must be very sure that the change will earn the +100 bonus to

make the risk worth it. This causes an alarm bell: do not make random adjustments, but high-value adjustments that are stable.

4. A.3 DQL Engine: What is the Rational behind "Double" Q-Learning?

The last one is the AI algorithm. The only major weakness that we discovered about standard Q-Learning is that it is an overexcited student, it overvalues its actions. That one mistake produces panic-stricken volatile UI alterations.

We suggest Double Q-Learning (DQL) due to its solution of that. Fig. 2 shows that DQL separates action selection and evaluation.

- It also relies on a single AI network (Q 1 Network) to select the optimal action.
- It has a second AI network (Q-Network 2) that measures the quality of that action that has been selected.

This second opinion prevents the overexcitement of the AI. It makes the agent less reckless and volatile and this is ideal in our framework. Our Cognitive Load Penalty, together with the reliability of DQL, produces a system that is designed to be attentive to the user, as well as prevent frustration.

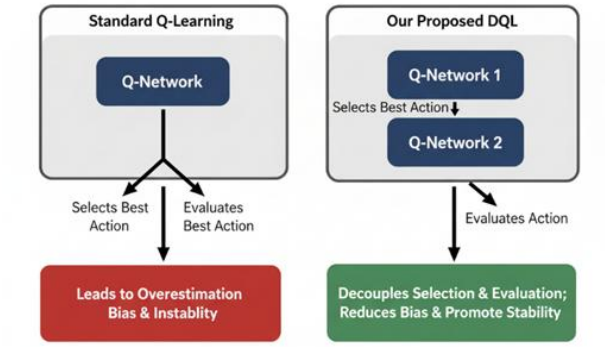

Fig. 2.Theoretical Figure of Standard vs. Double Q-Learning.

IV.RESULTS AND DISCUSSIONS

The paper is a discussion of a new concept known as the Cognitive-Load-Aware Framework (CLAF). It is not a literal experiment that we are going to do, but we are simply magnifying the way we imagine it is going to go and why we chose this design as opposed to the typical ones.

The two big problems with the existing systems are that they cause you to have to decide between two bad options: a Static UI (which is stable yet pretty much idiotic) or a Standard Q-Learning UI (which is smart but a real maniac and crashes occasionally).

Our CLAF intends to introduce a third and improved alternative: a steady and intelligent interface.

- Framework Comparison on Qualitative Basis.

And what exactly are these trade-offs regarding the performance of the app? A brief overview of the performance of how we believe our framework will fare against the two baselines is presented below. It was not a case of wild guesses but rather we are relying on the glaring weaknesses of the baselines and the new stuff we added to CLAF.

Metric	Static UI (Baseline)	Standard Q-Learning (Baseline)	Our Proposed CLAF (Expected)
Adaptability	None	High, but erratic.	High & Meaningful
Policy Stability	Perfect (Never changes)	Very Low (Prone to over-adaptation)	Very High
User Cognitive Load	High (not an expert users)	Very High (User needs re-learning)	Low (Stable and predictable)
Expected Task Success	Low to Medium	Low to Medium (due to frustration)	High
Expected User Trust	Neutral	Very Low	High

- Discussion of Anticipated Outcomes

This is not merely a wish to think so. We will be addressing the difference that will be made by two important concepts that we have introduced which are the Cognitive Load Penalty (CLP) and the DQL algorithm.

1. Failure of Standard Q-Learning: The "Trust-Breaker"

The actual issue with the common Q-Learning model is not that it is stupid. It's that it's "trigger-happy".

It overestimates, which makes every insignificant random click a new discovery. Therefore, when one user accidentally presses the profile button once, the AI will immediately assume, "Whoa! This they love! and presses that button in front.

That makes people lose faith. The UI is disorganized and untrustworthy and users simply abandon it, not

because the app is not clever, but because it is not acting clever.

2. The Solution of Our Framework to This: Patience and Prudence

The entire argument of CLAF is stability. We constructed that up to the foundation.

First is the Cognitive Load Penalty (CLP) which is a kind of patience training on the part of the AI. It is a mere -5 point tax on each and every change. This will compel the AI to stop and enquire: Is this one of the UI shifts worth baffling the user? Is it really going to pay me that +100 bonus?

Usually the answer is "no." That penalty assists the model to disregard the occurrence of random noise and respond solely to robust and patterned trends.

Then Double Q-Learning (DQL) is the second opinion. When CLP suggests the hesitation, DQL is the voice of reason that prevents the AI to become overly excited. It ensures that by the time a decision is arrived at, it will be at more realistic value, which is less optimistic.

Conclusions of This Study.

Ultimately, we are demanding a paradigm change in the way we conceptualize the idea of successful adaptive UIs. It does not concern rapid adaptation, it concerns significant change.

When we put the need to build user trust and maintain a low cognitive load, we will be able to prevent creating interfaces that feel like experimental arousal. We are able to make something useful, patient and slow learning such as a helpful buddy who really gets you over time.

That is what CLAF is all about. It provides the tradeoff between dumb-but-stable and smart-but-chaotic between the Q-Learning and the dumb-but-stable static UIs. It is the inaugural system that is witty and credible.

V.FUTURE SCOPE

This paper provides us with a sort of roadmap of this new Cognitive-Load-Aware Framework, or CLAF as we will call it. We have figured out the reason why the idea is cool on paper but that is not all. The next step is to make it into a real, functioning app that works with people and what opens up is a host of interesting questions in research.

1. Empirical validation and Prototyping: Now is actually the hardest part to get it built. We should transform the blueprint into a prototype application with the help of React, Node.js, and PyTorch. When we have something running, we will then carry out controlled experiments of real users. It is planned to conduct A/B tests, which compare CLAF to two other UIs, one being a completely static design and another being a simple Q-Learning UI without limitations. That is the only method to have real-world data. We must test the hypothesis: Does the so-called Cognitive Load Penalty actually have any effect, i.e., does it reduce task performance, eliminate frustration, and enable users to learn to trust the system in the long-term?

2. Implementing Deep Q -Networks (DQN): At this point we consider a standard DQL in the initial implementation. But we must admit it- in real life it is sloppier. A Deep Q-Network (DQN) is much more powerful, in theory, when there are thousands of different combinations of components, or when the input to a UI is an image in itself. Future research must consider replacing the simple DQL agent with a DNN-based agent. In that manner the system will be able to learn more subtle and complex patterns of users and work with high dimensional data.

3. Explainable Adaptation (XAI): It is an application that does a magical change on everything and is incredibly sketchy. The mystery destroys trust even when the changes are beneficial. And this is the reason why we should have an Explainable AI layer. The system cannot simply transform things, it must tell why it did. A small popup message stating, I decided to put my menu here because you clicked on it the most would make people feel better. Such transparency is not only nice, but also necessary in order to create user confidence.

4. Dynamic Reward Modeling: The current Cognitive Load Penalty (CLP) is a simple one: a constant fee on any change. This penalty would be varied in a smarter model. Why is a little color adjustment as expensive as a positioning a complete navigation bar? They shouldn't. A minor adjustment may be -1 and a major -20. In such a way the AI will learn how to be more accurate and adapt to the place where it actually is needed.

5. Ethical Guardrails and Adaptive Dark Patterns: This is a two-sided sword, Ethical Guardrails. It can be misused. When a system is aimed at engagement only, it may establish adaptive dark patterns. As an illustration it may pulse the Buy button on users that it believes are impulsive. This is the reason why effective ethical guardrails have to be developed in future research. We must have the tough limits that prevent the AI to pursue business goals at the expense of user welfare, fair play, and availability.

VI.CONCLUSION

In this study, the researcher presents the Cognitive-Load-Aware Framework (CLAF). Our new conceptualized way of coming up with adaptive user interfaces (UI) that are not only intelligent, but at last stable.

The issue is that we face a two-sided problem: the inflexible and restrictive nature of the static UIs and the unstable and chaotic nature of the traditional Q-Learning models. Our model provides a two-stage learning approach, which, most importantly, puts the trust of users in the first place and strives to reduce the cognitive load.

The design of the CLAF is based on two major innovations.

- At first, we have the Cognitive Load Penalty (CLP). It is the process that actively prevents the AI to make arbitrary and random changes to the UI. Second, we apply Double Q-Learning (DQL) to address the over-excitement problem that is observed in classic reinforcement learning- based on overestimation bias.
- It is this combination that is the main part of our approach. It allows the system to train a generalized "habit" model on a offline system and switch to an individualized fine-tuned interface online. but--this is the most important part--it will only switch to that in a case when the persistence of behavior is shown by the user that an adaptation is actually meaningful.

We have designed a system that is proactive in thinking twice before executing a change. That, we think, is the avenue to making the long-lost leap to get out of the old dilemma. No longer the option of fixed UIs on one hand and excessively turbulent adaptive ones on the other.

The whole purpose of the CLAF is to provide interfaces, which are not only intelligent, but also intelligent.

What's the difference? It is about making sure that customization increases user experience and usefulness, yes, but it must not come at the price of the vital components of the matter: reliability and trust in the user.

In our opinion, this piece of work preconditions the construction of genuinely intuitive digital experiences. And that base is applicable to all platforms web and mobile apps all the way up to assistive technologies that are specialized.

VII.ACKNOWLEDGMENT

Our faculty mentor owes a hearty debt of gratitude to us. They were the cornerstones of this conceptual framework because of their thought-provoking discussions, priceless advice, and unwavering support. Their profound understanding of human-computer interaction and reinforcement learning significantly determined the magnitude of knowledge and scope of their study.

We shall also need to owe a debt of gratitude to our peers and colleagues, whose helpful criticism and insightful questions allowed us to refine and improve the ideas in this paper.

This became possible due to the academic set up and facilities at the Department of Computer Engineering and Software Engineering at the Vishwakarma Institute of Technology, Pune. We owe them a debt of gratitude that they provided the spirit of collaboration that was very crucial in this project.

REFERENCES

[1] I -A. Salihu, R. Ibrahim, B. S. Ahmed, K. Z. Zamli and A. Usman, "AMOGA: A Static-Dynamic Model Generation Strategy for Mobile Apps Testing," in *IEEE Access*, vol. 7, pp. 17158-17173, 2019.

[2] Z. Liu, C. Chen, J. Wang, Y. Huang, J. Hu and Q. Wang, "Nighthawk: Fully Automated Localizing UI Display Issues via Visual Understanding," in

IEEE Transactions on Software Engineering, vol. 49, no. 1, pp. 403-418, 1 Jan. 2023.

[3] X. Xiao, X. Wang, Z. Cao, H. Wang and P. Gao, "IconIntent: Automatic Identification of Sensitive UI Widgets Based on Icon Classification for Android Apps," *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, Montreal, QC, Canada, 2019, pp. 257-268.

[4] M. Wan, N. Abolhassani, A. Alotaibi and W. G. J. Halfond, "An Empirical Study of UI Implementations in Android Applications," *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, Cleveland, OH, USA, 2019, pp. 65-75.

[5] I. A. Salihu, R. Ibrahim and A. Usman, "A Static-dynamic Approach for UI Model Generation for Mobile Applications," *2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Noida, India, 2018, pp. 96-100.

[6] Q. Sun, Y. Xue and Z. Song, "Adaptive User Interface Generation Through Reinforcement Learning: A Data-Driven Approach to Personalization and Optimization," *2024 6th International Conference on Frontier Technologies of Information and Computer (ICFTIC)*, Qingdao, China, 2024, pp. 1386-1391.

[7] Y. Wang, H. Xu, Y. Zhou, M. R. Lyu and X. Wang, "Textout: Detecting Text-Layout Bugs in Mobile Apps via Visualization-Oriented Learning," *2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE)*, Berlin, Germany, 2019, pp. 239-249.

[8] L. Huang and Y. Zheng, "Design and Application of Experiential Music Teaching System based on Java 2 Enterprise Edition Architecture," *2025 International Conference on Intelligent Systems and Computational Networks (ICISCN)*, Bidar, India, 2025, pp. 1-7.

[9] Y. Xu, X. Li, Y. Yang, Z. Lin, L. Wang and W. Li, "FedABR: A Personalized Federated Reinforcement Learning Approach for Adaptive Video Streaming," *2023 IFIP Networking Conference (IFIP Networking)*, Barcelona, Spain, 2023, pp. 1-9.

[10] L. Rittger, D. Engelhardt and R. Schwartz, "Adaptive User Experience in the Car—Levels of Adaptivity and Adaptive HMI Design," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 5, pp. 4866-4876, May 2022.

[11] You-Dong Yun, C. Lee and H. -S. Lim, "Designing an intelligent UI/UX system based on the cognitive response for smart senior," *2016 2nd International Conference on Science in Information Technology (ICSITech)*, Balikpapan, Indonesia, 2016, pp. 281-284.

[12] Heuiseok Lim, "Introduction to an intelligent UI/UX for aging people," *2016 2nd International Conference on Science in Information Technology (ICSITech)*, Balikpapan, Indonesia, 2016, pp. 1-1.

[13] D. W. P. Budi, F. A. Anshary and A. A. N. Fajrillah, "UI/UX Design for a Mobile Flood Information Dissemination Application in Citeureup Village Using the Design Thinking Method," *2025 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, Surabaya, Indonesia, 2025, pp. 178-183.

[14] M. Yalla, M. Venkata Raman and M. Fernandes, "Deep Industry Use Cases on Context-Aware Adaptive Mobile Systems Experience Testing," *2023 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, Dublin, Ireland, 2023, pp. 105-106.

[15] D. He, Y. Zou, J. Wu, X. Zhang, Z. Zhang and R. Wang, "Deep Q-Learning Based Energy Management Strategy for a Series Hybrid Electric Tracked Vehicle and Its Adaptability Validation," *2019 IEEE Transportation Electrification Conference and Expo (ITEC)*, Detroit, MI, USA, 2019, pp. 1-6.

[16] C. Qiu, F. R. Yu, H. Yao, C. Jiang, F. Xu and C. Zhao, "Blockchain-Based Software-Defined Industrial Internet of Things: A Dueling Deep Q -Learning Approach," in *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4627-4639, June 2019.

[17] P. Tam, S. Math, C. Nam and S. Kim, "Adaptive Resource Optimized Edge Federated Learning in Real-Time Image Sensing Classifications," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 10929-10940, 2021.

[18] A. A. Hammadi, L. Bariah, S. Muhaidat, M. Al-Qutayri, P. C. Sofotasios and M. Debbah, "Deep Q-Learning-Based Resource Management in IRS-Assisted VLC Systems," in *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 2, pp. 34-48, 2024.

[19] C. Liebing, R. Mennerich and A. Schill, "A Pragmatic Approach for Matching UI Components on Web Service Operations," *2010 6th World Congress on Services*, Miami, FL, USA, 2010, pp. 621-628.

[20] K. A. Schneider and J. R. Cordy, "AUI: a programming language for developing plastic interactive software," *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, Big Island, HI, USA, 2002, pp. 3656-3665.

[21] X. Wang, J. Wang, Z. Qian and B. Zhang, "Online Adaptable Offline RL With Guidance Model," in *IEEE Transactions on Neural Networks and Learning Systems*, 2025 (Early Access).

[22] A. Riahi, A. Mohamed and A. Erbad, "RL-Based Federated Learning Framework Over Blockchain (RL-FL-BC)," in *IEEE Transactions on Network and Service Management*, vol. 20, no. 2, pp. 1587-1599, June 2023.

[23] T. Sboui, M. Ben Ayed and A. M. Alimi, "A UI-DSPL Approach for the Development of Context-Adaptable User Interfaces," in *IEEE Access*, vol. 6, pp. 7066-7081, 2018.

[24] Y. Duan, L. Yang, T. Zhang, Z. Song and F. Shao, "Automated UI Interface Generation via Diffusion Models: Enhancing Personalization and Efficiency," *2025 4th International Symposium on Computer Applications and Information Technology (ISCAIT)*, Xi'an, China, 2025, pp. 780-783.

[25] A. Domínguez, J. Flórez, A. Lafuente, S. Masneri, I. Tamayo and M. Zorrilla, "A Model for User Interface Adaptation of Multi-Device Media Services," in *IEEE Transactions on Broadcasting*, vol. 67, no. 3, pp. 606-618, Sept. 2021.