Phoenix: International Multidisciplinary Research Journal
Vol 4, No. 1.1, March, 2026
ISSN: 2583-6897

A COMPARATIVE STUDY OF PROGRESSIVE WEB APPS (PWAS) VS NATIVE APPS: USER EXPERIENCE (UX)[1]

Ms. Dhanashree Satish Wagh
PDEA's Baburaoji Gholap College, Savitribai Phule Pune University, Pune
Mr. Rushikesh Y Hiwale
PDEA's Waghire College of Arts, Commerce and Science, Savitribai Phule Pune University, Pune
Dr. Seema. S. Chowhan
PDEA's Baburaoji Gholap College, Savitribai Phule Pune University, Pune

Abstract - Mobile applications are central to communication, commerce, and learning, making the choice of development approach critical. Two dominant models are native apps, built in platform-specific languages (e.g., Swift, Kotlin), and Progressive Web Apps (PWAs), which use web technologies but offer native-like features. Native apps integrate deeply with the OS, enabling advanced hardware access, smooth animations, and robust offline capability, at the expense of multiple codebases and higher maintenance costs. PWAs run from a single codebase, can be installed from the browser, consume minimal storage, and work offline through service workers — offering lower cost and wider reach. This study evaluates PWAs and native apps across five UX dimensions (speed, navigation, offline performance, storage efficiency, overall satisfaction) using a mixed-methods approach: quantitative benchmarking (Lighthouse, Chrome Dev.-Tools, and Android Profiler) and qualitative survey analysis (50–100 respondents). Additionally, the study incorporates trend data (2015–2029) to situate results in the context of rising mobile adoption and increasing mobile traffic share. The report also outlines a Random Forest-based classification pipeline to model retention and experience outcomes from survey features.

Keywords- *Native Mobile Applications, Progressive Web Apps (PWAs), User Experience (UX) Evaluation, Mobile App Performance.*

I. INTRODUCTION

The worldwide digital economy is driven by mobile applications, which support a range of activities from social networking and online shopping to healthcare services and educational tools. In light of over 7.5 billion mobile users and billions of annual downloads (Statist, 2024), businesses must emphasize user-friendly strategies to stay competitive. The key choice lies between Progressive Web Apps (PWAs), which are web-based solutions that utilize HTML, CSS, JavaScript, service workers, and manifests, and native applications, which are designed for specific platforms using languages such as Swift or Kotlin that offer superior access to hardware, offline capabilities, and seamless performance. PWAs help lower expenses and speed up deployment by allowing cross-platform launches from a singular codebase, installation through browsers, minimal storage requirements (usually under 10 MB), automatic updates, and limited offline functionality. Despite the growing popularity of PWAs, many organizations still rely on native applications due to their optimized performance, deeper integration with device hardware, and better responsiveness. This creates a need to understand how users perceive the experience of both approaches and which factors influence their long-term engagement. The aim of this study is to compare the user experience (UX) of Progressive Web Apps (PWAs) and native mobile applications using survey data in order to identify the major factors that influence user satisfaction and retention. The study focuses on examining important usability aspects such as perceived load time, responsiveness, and offline behaviour while also analysing how storage requirements and installation processes affect user preference and continued usage. In addition, the research evaluates user perceptions, usability, and satisfaction levels when interacting with both

application types. To gain deeper insights, machine learning techniques, particularly the Random Forest algorithm, are applied to predict user retention and overall experience while identifying the most influential features affecting satisfaction. Based on these insights, the study aims to provide practical recommendations for improving the user experience of both PWAs and native applications, helping developers and organizations make informed decisions regarding mobile application design and deployment strategies

II. BACKGROUNF AND RELATED WORK

Nonetheless, this choice directly influences user experience (UX), encompassing satisfaction, speed, reliability, and ease of use. PWAs provide equal access, especially in low bandwidth regions, whereas native apps shine in smooth integration. Subjective elements, like perceived simplicity, confidence, and enjoyment, lack the in-depth study found in technical metrics, such as loading durations. To assist developers in attaining optimal UX-cost ratios, this research assesses speed, offline availability, storage, and navigation via performance evaluations, user feedback, and sentiment analysis

A. Native apps and PWAs offer distinct paths for user engagement among the swift growth of mobile, yet their UX impacts necessitate thoughtful reflection by businesses, developers, users, and the broader community.

2.1 Benefits for Enterprises

Because of two codebases and app store holdups, native development increases expenses twofold. PWAs streamline this through cohesive code, instant updates, and proven advantages: Twitter Lite lowered data consumption by 70% and boosted engagement by 65%, whereas Starbucks' PWA shrank to 1 MB, halved load times to 2.5 seconds, and doubled

I. INTRODUCTION

The worldwide digital economy is driven by mobile applications, which support a range of activities from social networking and online shopping to healthcare services and educational tools. In light of over 7.5 billion mobile users and billions of annual downloads (Statist, 2024), businesses must emphasize user-friendly strategies to stay competitive. The key choice lies between Progressive Web Apps (PWAs), which are web-based solutions that utilize HTML, CSS, JavaScript, service workers, and manifests, and native applications, which are designed for specific platforms using languages such as Swift or Kotlin that offer superior access to hardware, offline capabilities, and seamless performance. PWAs help lower expenses and speed up deployment by allowing cross-platform launches from a singular codebase, installation through browsers, minimal storage requirements (usually under 10 MB), automatic updates, and limited offline functionality. Despite the growing popularity of PWAs, many organizations still rely on native applications due to their optimized performance, deeper integration with device hardware, and better responsiveness. This creates a need to understand how users perceive the experience of both approaches and which factors influence their long-term engagement. The aim of this study is to compare the user experience (UX) of Progressive Web Apps (PWAs) and native mobile applications using survey data in order to identify the major factors that influence user satisfaction and retention. The study focuses on examining important usability aspects such as perceived load time, responsiveness, and offline behaviour while also analysing how storage requirements and installation processes affect user preference and continued usage. In addition, the research evaluates user perceptions, usability, and satisfaction levels when interacting with both application types. To gain deeper insights, machine learning techniques, particularly the Random Forest algorithm, are applied to predict user

retention and overall experience while identifying the most influential features affecting satisfaction. Based on these insights, the study aims to provide practical recommendations for improving the user experience of both PWAs and native applications, helping developers and organizations make informed decisions regarding mobile application design and deployment strategies

II. BACKGROUNF AND RELATED WORK

Nonetheless, this choice directly influences user experience (UX), encompassing satisfaction, speed, reliability, and ease of use. PWAs provide equal access, especially in low bandwidth regions, whereas native apps shine in smooth integration. Subjective elements, like perceived simplicity, confidence, and enjoyment, lack the in-depth study found in technical metrics, such as loading durations. To assist developers in attaining optimal UX-cost ratios, this research assesses speed, offline availability, storage, and navigation via performance evaluations, user feedback, and sentiment analysis

A. Native apps and PWAs offer distinct paths for user engagement among the swift growth of mobile, yet their UX impacts necessitate thoughtful reflection by businesses, developers, users, and the broader community.

2.1 Benefits for Enterprises

Because of two codebases and app store holdups, native development increases expenses twofold. PWAs streamline this through cohesive code, instant updates, and proven advantages: Twitter Lite lowered data consumption by 70% and boosted engagement by 65%, whereas Starbucks' PWA shrank to 1 MB, halved load times to 2.5 seconds, and doubled active users. These instances showcase the benefits of scalability and reach offered by PWAs

2.2 Advantages for Developers

PWAs speed up iteration—absence of app-store barriers allows for quick deployments, enabling teams to focus on UX advancements such as enhanced interfaces and functionalities. This flexibility shortens time-to-market and lowers maintenance expenses by surpassing the disconnected processes of native applications.

2.3 Effects on Users

Users favor PWAs due to their minimal size (under 10 MB) and web-based installations, making them ideal for budget-friendly devices in developing countries with unstable networks. Native apps counterbalance superior speed, animations, and robust hardware connections (e.g., GPS, alerts). Measuring user trade-offs, like storage compared to polish, reveals preferences that impact retention.

2.4 Importance in Academia and Society

The literature is filled with empirical standards, yet it scarcely addresses subjective UX elements like trust or emotional connection. This effort fills that gap. Socially, PWAs enhance inclusion in disadvantaged communities by enabling access to vital services (like education and banking) on low-spec devices

Conceptual Background

This section outlines the key concepts underlying the comparison of PWA and native app UX, emphasizing UX principles and the technological features that affect them.

User Experience (UX): In digital engagements, UX includes the overall feelings, emotions, and sense of satisfaction experienced by users. It encompasses velocity, reliability, functionality, aesthetics, and emotional reactions. Seamless navigation, fast loading speeds, flexible designs, and a feeling of safety and assurance are essential for an excellent user experience (UX), which directly enhances engagement, loyalty, and commitment.

Indigenous Software

Native applications, developed in languages such as Swift, Kotlin, or Java and tailored for a specific operating system (like iOS or Android), are distributed via official stores. They possess a profound integration with hardware, encompassing cameras, GPS, sensors, and safe local storage

Scope of the Study

This defines the scope of the research to ensure its concentration, precision, and consistency with the UX comparison aims between PWAs and native applications, integrating user viewpoints and empirical data.

Components Included

Performance Metrics: Utilize instruments such as Google Lighthouse, Chrome DevTools, Android Profiler, and Pingdom to assess speed. For accurate assessments, monitor loading times, initial paint, responsiveness, battery usage, and offline capability.

User Perspectives on Various Configurations: Employ surveys across multiple devices (smartphones, tablets, laptops) and connection types (broadband, 4G/5G, poor signals) to assess subjective impressions (e.g., perceived speed, ease of navigation, satisfaction), mirroring everyday usage.

Dual-Format Applications: To ensure a balanced evaluation of user experience and technology, target public applications that have PWA/native counterparts (like Twitter, Pinterest, and Starbucks)

Literature review

This chapter reviews existing research related to **Progressive Web Applications (PWAs)** and **Native Mobile Applications**, focusing on user experience, performance, and practical challenges. The literature provides theoretical understanding, empirical comparisons, and insights into the benefits and limitations of both technologies. By examining previous studies, this section highlights the current state of knowledge and identifies research gaps in the comparison of PWAs and native applications from a user experience perspective.

User Experience in Mobile Web, Native, and Progressive Web Applications

User experience (UX) plays an important role in determining how users interact with mobile applications. Research conducted by A. Cardieri, G. de Andrade, and L. A. M. Zaina (2018) examined the user experience of mobile web applications, native apps, and Progressive Web Apps from both user and Human–Computer Interaction (HCI) expert perspectives. Their study found that native applications generally provide smoother interactions, faster response time, and better integration with device features. However, Progressive Web Apps offer advantages such as easier accessibility through browsers, reduced installation effort, and lower storage requirements. The research highlights that while native apps deliver a richer experience in terms of performance and design flexibility, PWAs can still provide satisfactory usability for many users, especially when accessibility and quick access are priorities. This study emphasizes that user expectations and device capabilities significantly influence the perceived quality of mobile application experiences.

Practices and Challenges of Progressive Web Applications

A systematic literature review conducted by Reza Fauzan, Ice Krisnahati, Bima Dinda Nurwibowo, and Della Aulia Wibowo (2022) analyzed research on Progressive Web Applications published between 2015 and 2021. The study identified common practices in PWA development, including the use of service workers for offline capabilities, web app manifests for installation, and push notifications to improve user engagement. The review also discussed several benefits of PWAs, such as improved accessibility, reduced development cost, and the ability to work across multiple platforms. However, the researchers noted challenges related to browser compatibility, inconsistent support for certain device features, and limitations in offline functionality compared to native applications. Their findings suggest that although PWAs are becoming increasingly

popular, there are still technical and usability challenges that developers must address when implementing them in real-world applications.

Performance and Energy Consumption Comparison

Another important aspect in evaluating mobile applications is performance efficiency. Research by Ruben Horn, Abdellah Lahnaoui, Edgardo Reinoso, Sicheng Peng, Vadim Isakov, Tanjina Islam, and Ivano Malavolta (2023) compared native Android applications with their web-based counterparts, including Progressive Web Apps. The study focused on resource usage such as CPU consumption, memory utilization, energy usage, and network traffic. Results indicated that native applications generally perform more efficiently in terms of energy consumption and system resource management. Because native apps are directly integrated with the operating system, they can optimize hardware usage more effectively. This efficiency contributes to longer battery life and better performance, which can positively influence the user experience. However, web-based applications, including PWAs, still offer advantages in terms of development simplicity and cross-platform accessibility.

Impact of Progressive Web Apps on User Engagement and Accessibility

Further research by Abbin Joseph Thomas and S. Rajesh Kumar (2024) explored how Progressive Web Apps are transforming modern web applications and improving user engagement. Their study highlighted core features of PWAs such as offline access, push notifications, responsive design, and app-like interfaces. These features help increase user engagement by providing reliable and fast interactions even under unstable network conditions. The research also discussed how PWAs improve accessibility for businesses by allowing users to access applications without downloading them from app stores. However, the study noted that some organizations face challenges in adopting PWAs due to limited awareness, technical constraints, and concerns about performance compared to native apps. Despite these limitations, the authors concluded that PWAs have significant potential to enhance user experience when properly designed and optimized.

Research Gaps and Study Motivation

Although previous studies have explored the performance, usability, and technical aspects of Progressive Web Apps and native applications, there is still a need for more comparative research focusing specifically on user experience under different device conditions and network environments. Many studies analyze performance metrics or development practices separately, but fewer studies combine user perception, usability factors, and device-related variables in a single evaluation. Additionally, rapid advancements in mobile technology continue to change how users interact with applications. Therefore, this study aims to contribute to the existing literature by analysing user experience differences between PWAs and native applications through survey-based analysis and performance considerations, providing insights that can guide developers in selecting the most appropriate technology for their applications.

III. RESAERCH METHODOLOGY

This outlines a systematic method for analyzing PWA versus native app user experiences, integrating user feedback and quantitative metrics for a thorough, evidence-based understanding of technological potential and user reactions.

A. Identification of Problems

Context & Motivation: The total of mobile users grew from 3.2 billion in 2015 to 7.6 billion in 2029, traffic rose by 45% to 80%, and daily usage time expanded from 2.5 to 5.6 hours—highlighting the most effective app strategies. Web-based, unified-code PWAs rival platform-specific native applications. To facilitate clear comparisons, the dataset captures user feedback on speed, simplicity, enjoyment, space utilization, and allegiance.

Hypotheses:

The study is guided by several hypotheses related to user experience and application preference. It is hypothesized that Progressive Web Apps (PWAs) can provide load time and responsiveness that are comparable to those of native applications, offering users a smooth and efficient experience. Another assumption is that the lower storage requirements of PWAs positively influence user preference, as many users favour applications that consume minimal device storage. In addition, user satisfaction is expected to be significantly influenced by factors such as application speed, ease of navigation, and the simplicity of the installation process. Finally, it is hypothesized that PWAs are more likely to be preferred in heterogeneous device environments, where users operate across different devices and platforms, due to their cross-platform compatibility and accessibility through web browsers.

Research Design

Design Type: Convergent mixed-methods combining survey analysis with predictive modeling.

Components:

1. Quantitative Survey Analysis: Use ordinal mapping for speed, satisfaction, and influencing factors. Descriptive statistics, cross-tabulations, and correlations describe patterns.
2. Qualitative Analysis: Open-ended Influencing Factors coded into themes (storage, offline, ease of installation, etc.).
3. Predictive Modeling: Random Forest predicts Retention Preference and Overall Experience. Feature importance identifies primary predictors.

Rationale: Mixed methods capture both perceived and measurable user experience. Random Forest handles categorical data effectively and allows interpretability.

Sampling Frame

Population:The population of this study includes mobile and cross-device users who have experience using Progressive Web Apps (PWAs) or native mobile applications. These users may access applications through smartphones, tablets, or computers and have practical experience interacting with different types of apps.

Strata:To make the study more balanced, the population is divided into different groups (strata). These groups are based on age categories (18–25, 26–35, 36–50, and 50+), device type (Android phones, iPhones, MacBook/iPad, Windows laptop/PC, and tablets), how frequently the user uses apps, and whether the user has previous exposure to PWAs. Dividing users into these groups helps analyze how different types of users experience PWAs and native apps.

Justification:Using these groups ensures that the study includes people from different age groups, device types, and usage habits. This helps make the results more reliable because the data represents a wider range of users instead of focusing on only one type of user.

Limitations:There may still be some limitations in the sampling. For *example*, ***selection bias*** may occur if more responses come from a particular group of users. Also, users who have **limited internet connectivity or less access to digital devices** might not participate in the survey, which means their experiences may not be fully represented in the study.

Sampling Type

Strategy:The study uses **stratified random sampling**, where participants are divided into groups based on **age and device type**. From each group, participants are randomly selected. This method helps ensure that different types of users (different ages and devices) are fairly represented in the study.

Sample Size: A total of **50–100 participants** are selected for the survey. This sample size is considered enough to perform **descriptive analysis (basic data summary), inferential analysis (finding relationships or differences), and predictive analysis (building models to predict outcomes)**.

Non-response Handling: If some survey responses are missing, the missing values for **categorical variables** (such as device type or preference) are replaced using the **mode**, which means the most common answer given by other participants. All missing responses are also **recorded and tracked** to maintain transparency in the data analysis process

Data Collection

Data was collected through a survey dataset containing fields such as age, device, usage, speed, navigation, satisfaction, influencing factors, and retention, while trend data was used for context. The survey instrumentation included both closed and open items, where respondents provided ordinal ratings for speed, navigation, and satisfaction, binary preferences, and free-text Influencing Factors. During data cleaning and preprocessing, missing values were replaced with the mode, and ordinal fields were mapped to numeric scores. Influencing Factors were parsed into binary flags, and derived features such as speed_diff, navigation_diff, and device type indicators were created. Label encoding was applied to categorical variables, and the dataset was divided into training and testing sets using a train/test split for predictive modeling. Ethical considerations were followed by obtaining informed consent, removing personal identifiers, and storing the data securely.

Analytical Workflow

The analytical workflow followed a structured pipeline to ensure reproducibility. In the planning and design phase, the questionnaire was finalized with ordinal scales and piloted ($n \approx 10$). The survey was then launched across channels during data acquisition while monitoring strata and logging metadata. Data was archived with time stamped raw files in a versioned folder structure (/data/raw/, /data/processed/, /notebooks/). During preprocessing, survey data was imported from Excel, quality checks were performed using info and describe methods, missing categorical values were filled with the mode, and speed ratings were converted from Very Slow=1 to Very Fast=5. Binary indicators for influencing factors such as storage, security, offline, installation, and features were created, speed differential was calculated, and label encoding was applied before performing a stratified 80/20 train-test split. Exploratory Data Analysis (EDA) included histograms, bar charts, pie charts, boxplots, cross-tabulations, and Spearman correlations to analyze trends. Hypothesis tests were conducted where H1 (speed) used paired t-test/Wilcoxon test ($p<0.05$, Cohen's d), H2 (storage) used Chi-square/logistic regression (odds ratio, confidence interval), H3 (drivers) used random forest/glm coefficients, and H4 (adoption) used Chi-square test (η^2). Feature engineering included interactions between smartphone and storage and optional PCA. Random Forest modeling was performed using grid search for parameters (n_estimators, max_depth=[None,10,20], min_samples_split/leaf) with 5-fold cross-validation and macro-F1 scoring. Model evaluation used F1 score, confusion matrix, and SHAP/PDP analysis, and the results were mapped to practical recommendations such as caching strategies for improved speed. Finally, reporting integrated visuals such as users_growth.png and feature plots with captions and notes. *Deployment*: Optional Streamlit dashboard for predictions

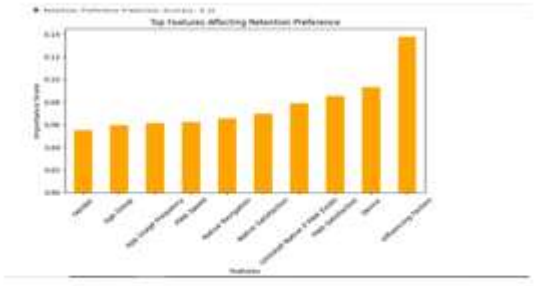

Fig. No.1 Top Features affecting retention preferences

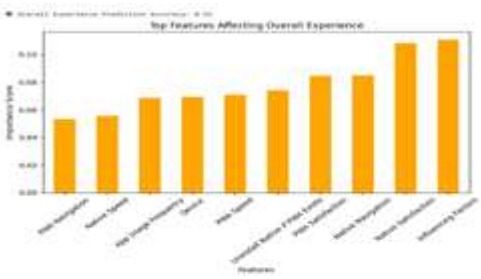

Fig. No.2 Top Features affecting overall experience

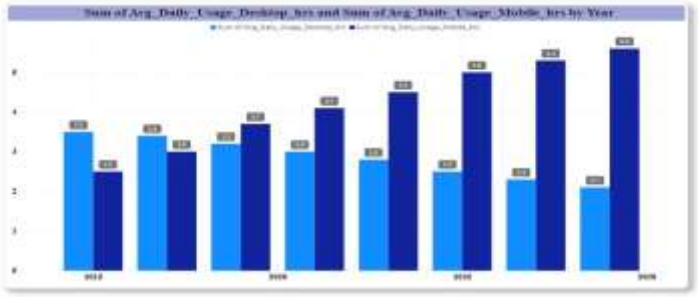

Fig. No.3 Sum of daily desktop and mobile usage per year (hours).

Fig. No.4 Category-wise yearly daily usage totals with comparison of desktop and mobile hours. (hours).

Fig. No.5 Features Contribution to Retention Preference

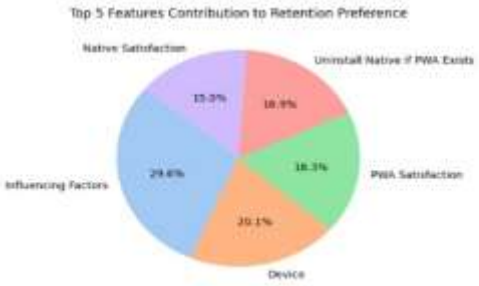

Fig. No.6 Features Contribution to Overall Experience

IV. RESULTS AND DISCUSSION

The study expects to reveal that Progressive Web Apps (PWAs) perform strongly in terms of storage efficiency, ease of installation, and cross-platform accessibility, particularly in low bandwidth conditions where lightweight applications are beneficial. In contrast, native applications are likely to outperform PWAs in areas such as offline functionality, smoother animations, and performance involving heavy graphics. Survey responses are expected to show multi-class sentiment among users, where some users may prefer the convenience and quick access provided by PWAs, while others may favor the richer and more responsive experience offered by native applications. Additionally, factors such as device class, for example high-end versus low-end smartphones, and cultural or regional context may influence user preferences and overall experience.

However, the research also has certain limitations. The sample size of approximately 100 participants may not fully represent the diverse population of global mobile users. Furthermore, variations in device specifications and network conditions cannot be completely controlled during the study, which may influence performance perceptions.

V. CONCLUSION

Conclusion and Recommendation

The study concludes that Progressive Web Apps (PWAs) are a suitable choice for businesses that want to reach a wide audience, especially in areas with limited internet bandwidth. PWAs require less storage space, are easy to install, and can be accessed across different platforms, making them cost-effective and faster to deploy. On the other hand, native applications are more suitable when strong device integration, advanced features, and reliable offline functionality are required. Native apps also provide smoother performance for graphics and complex tasks. Therefore, developers should design user experiences by considering the type of device users have and their expectations from the application. For future development, organizations can also consider hybrid strategies, such as providing both a PWA and a native application in different markets to achieve better reach and performance.

REFERENCES

[1] Yue, L., et al. (2019). A Survey of Sentiment Analysis in Social Media. Springer. Springer Link
[2] Neri, F., et al. (2012). Sentiment Analysis on Social Media. IEEE/ACM. Research Gate
[3] Badr, E.M., et al. (2019). Social Media Sentiment Analysis using Machine Learning and Optimization Techniques. International Journal of Computer Applications. Research Gate
[4] Abdukhamidov, E., et al. (2022). Sentiment Analysis of Users' Reactions on Social Media during the Pandemic. Electronics. MDPI.
[5] Kukkar, A., et al. (2023). Improving Sentiment Analysis in Social Media by Handling Lengthened Words. IEEE Access. Research Gate
[6] Omuya, E.O., et al. (2023). Sentiment Analysis using NLP. Wiley Engineering Reports. Research Gate