

Enhancing Accessibility Testing Using Machine Learning: A Systematic Mapping Study

Vijay Kumar Pal¹, Manav Bansal²

¹Scholar, M.Tech, Department of Computer Science and Engineering, Sir Chhotu Ram Institute of Engineering & Technology, Chaudhary Charan Singh University, Meerut, Uttar Pradesh, India

²Assistant Professor, IT Department, Sir Chhotu Ram Institute of Engineering & Technology, Chaudhary Charan Singh University, Meerut, Uttar Pradesh, India

Emails: vijaypradhan780019@gmail.com, manavbansal82@gmail.com

ARTICLE INFO

ABSTRACT

Received: 26 Dec 2024

Revised: 14 Feb 2025

Accepted: 22 Feb 2025

Ensuring accessibility in digital systems is essential for inclusive software design. Traditional accessibility testing methods are often manual, time-consuming, and lack scalability. In response, machine learning (ML) techniques have emerged as a promising avenue for automating and enhancing accessibility evaluations. This systematic mapping study explores how ML has been applied in accessibility testing, categorizing approaches by algorithm type, target accessibility issues, platform focus, and validation strategies. The study synthesizes prevailing trends, highlights existing gaps, and outlines future research directions. This work aims to provide a foundational reference for researchers and practitioners aiming to advance ML-driven accessibility testing.

Keywords: Digital Accessibility, Machine Learning Applications, Automated Evaluation, Inclusive Technology, Accessibility Barriers

INTRODUCTION

In the modern digital age, ensuring that software systems are accessible to all users—including individuals with disabilities—is an essential component of quality software development. Accessibility testing helps determine whether software can be effectively used by people with different abilities, such as visual, auditory, motor, or cognitive impairments. Traditional testing approaches are typically manual, requiring considerable expertise and effort, and often suffer from limitations such as inconsistency and time consumption.

The emergence of Machine Learning [1] (ML) has opened new opportunities to automate and scale accessibility testing [2]. ML-based solutions can identify accessibility problems more accurately and efficiently by learning from historical patterns and predicting potential violations. They support tasks like detecting low contrast, generating alternative text, identifying ARIA role misuse, and even simulating assistive tool behavior.

Despite these advantages, applying ML to accessibility testing [2] poses several challenges. These include the need for high-quality labeled datasets, risk of algorithmic bias, difficulty generalizing across platforms, and complexities in evaluating subjective aspects of accessibility. Importantly, ML should assist rather than replace human judgment—especially in cases where empathy and contextual understanding are crucial.

This study undertakes a systematic mapping of existing research that integrates ML into accessibility testing [2]. It explores how ML techniques are applied, categorizes the current literature, identifies trends and gaps, and offers insights for future research and practice in accessible technology.

RELATED WORK

Software testing

Accessibility Testing and Machine Learning [1] Integration

Accessibility testing is the process of ensuring that digital interfaces are usable by individuals with different physical, sensory, or cognitive abilities. Traditionally, compliance with standards like WCAG or Section 508 is evaluated manually or through rule-based tools that often miss contextual or dynamic issues. Machine learning offers a promising alternative by learning from data and enabling scalable, automated detection of issues across diverse platforms.

Machine Learning [1] (ML) offers promising capabilities to enhance and automate aspects of accessibility testing [2]. ML models can learn patterns from large datasets of accessible and inaccessible components and predict whether new instances exhibit similar violations. For instance, supervised learning can be used to classify UI elements or layout structures as accessible or problematic. Given a sufficiently large and annotated dataset, these models can generalize to detect unseen accessibility issues with high accuracy.

Unsupervised learning techniques, such as clustering and anomaly detection, may uncover irregular UI patterns or inconsistencies in design that suggest potential accessibility barriers—such as uneven element spacing or inconsistent heading structures. Reinforcement learning can simulate user interactions through assistive technologies to identify navigational challenges, such as improper tab ordering or missing keyboard focus. Meanwhile, deep learning approaches, particularly convolutional neural networks (CNNs), are capable of identifying visual accessibility issues like poor contrast or missing alt text directly from screenshots.

Natural Language Processing (NLP) models can assist in evaluating cognitive accessibility by analyzing content readability, tone, and clarity, ensuring that textual content is understandable for users with cognitive or learning disabilities. Additionally, generative models may assist in automatically suggesting accessibility repairs, such as generating alt text or correcting HTML structure.

Despite these advances, challenges remain. ML models require large, diverse, and well-labeled datasets to train effectively. Ensuring model transparency, minimizing bias against specific disability groups, and integrating ML solutions into real-world development workflows are ongoing concerns.

Nonetheless, ML holds significant potential to complement traditional accessibility testing [2] methods, reduce manual effort, and promote the development of more inclusive software systems at scale.

COMMON TEST GENERATION TECHNIQUES

Common Test Generation Techniques in Accessibility Testing

In recent years, the application of Machine Learning [1] (ML) to accessibility testing [2] has gained momentum, particularly in automating the generation of test cases. This automation aims to improve test coverage, reduce manual effort, and identify accessibility issues that may be overlooked by traditional rule-based approaches. Various ML-based techniques have been utilized to support test generation in this context.

Supervised learning is one of the most widely used approaches. It involves training models using labeled datasets where examples of accessible and non-accessible code patterns or UI components are available. These models learn to recognize specific features that correlate with accessibility violations, allowing them to predict such issues in unseen code or interface designs. For example, supervised models can be trained to detect missing alternative text, insufficient contrast, or improper ARIA roles based on past examples.

Unsupervised learning techniques are used when labeled data is not readily available. These methods analyze structural or behavioral patterns within datasets to discover anomalies or irregularities that may suggest accessibility problems. Clustering, dimensionality reduction, and anomaly detection methods can identify interface components that deviate from standard patterns, potentially signaling inconsistent labeling, layout disruptions, or navigational issues.

Natural Language Processing (NLP) techniques offer an alternative angle by analyzing textual requirements or documentation to automatically derive test cases. By extracting relevant accessibility-related information from user stories, requirement specifications, or guidelines, NLP models can generate test scenarios that align with both functional and accessibility needs. This is particularly useful for incorporating accessibility early in the software development lifecycle.

Reinforcement learning introduces an interactive approach, where autonomous agents learn optimal strategies for testing by interacting with the application environment. These agents receive feedback in the form of rewards or

penalties based on their ability to uncover accessibility flaws. Over time, they adapt their behavior to navigate interfaces more effectively, simulating users who rely on keyboard navigation or screen readers, and discovering issues that arise during dynamic interactions.

Together, these techniques contribute to a more intelligent and automated framework for accessibility testing [2]. By leveraging different learning paradigms, developers and testers can achieve broader coverage, uncover subtle issues, and integrate accessibility testing [2] more seamlessly into continuous development pipelines.

METHODOLOGY

This research adopts a systematic mapping methodology, which is a structured approach designed to identify, categorize, and analyze existing literature in a focused research area. The goal is to present a comprehensive overview of how machine learning (ML) has been incorporated into accessibility testing [2]. The process is based on established guidelines by Kitchenham and Charters (2007), and Petersen et al. (2008), ensuring that the results are transparent, reproducible, and academically rigorous.

1. Definition of Research Questions (RQs)

The research questions (RQs) were formulated to guide data collection and analysis. These include identifying which ML techniques have been applied in accessibility testing [2], the types of accessibility issues addressed, the nature of datasets and tools used, and challenges and trends observed across studies.

To identify which machine learning techniques have been applied in accessibility testing [2].

To explore which types of accessibility issues are most frequently addressed.

To understand the nature of the datasets and tools used in these studies.

To identify trends, challenges, and future directions in the field.

2. Search Strategy and Data Sources

To locate relevant literature, we constructed a search strategy using combinations of keywords such as “machine learning”, “accessibility testing [2]”, “automated testing”, and “test generation”. Boolean operators (AND, OR) were used to broaden or narrow the scope of results. Searches were conducted in academic databases including IEEE Xplore, ACM Digital Library, SpringerLink, ScienceDirect, and arXiv. These databases were selected due to their relevance and coverage of high-quality peer-reviewed literature in software engineering and artificial intelligence.

3. Study Selection Process

The study selection was carried out in two phases:

Initial screening: Based on titles and abstracts, papers unrelated to accessibility or machine learning were removed. Duplicates were also filtered out.

Full-text review: The remaining papers were evaluated based on inclusion and exclusion criteria. Only studies that discussed the application of machine learning in the context of accessibility testing [2] were considered. Papers not written in English, lacking sufficient technical depth, or focusing solely on theoretical aspects without practical evaluation were excluded.

4. Data Extraction and Classification

From each selected paper, relevant information was extracted including:

Publication year and venue

Type of machine learning approach (e.g., supervised, unsupervised, reinforcement)

Accessibility issues addressed (e.g., vision, hearing, cognitive impairments)

Dataset characteristics

Evaluation metrics used

Tools and frameworks employed

This data was then categorized into predefined groups to identify commonalities and patterns. A classification scheme was developed iteratively to ensure that the categorization accurately reflected the scope and focus of each study.

5. Data Synthesis and Analysis

The final step involved analyzing the classified data to answer the research questions. Quantitative results were presented in the form of charts and tables, showing trends in ML usage over time, frequently addressed accessibility issues, and popular datasets. Qualitative analysis was conducted to discuss recurring challenges and limitations observed across studies.

This methodology ensures that the findings of the mapping study are comprehensive, well-structured, and valuable for both researchers and practitioners working at the intersection of accessibility and artificial intelligence.

Initial study selection

The initial study selection process is a critical phase in any systematic mapping study. It helps in identifying a broad yet relevant set of research articles that potentially contribute to answering the predefined research questions. To ensure comprehensiveness, transparency, and replicability, we followed a structured approach to conduct the initial selection of studies.

We began by identifying a list of credible and well-recognized digital libraries and academic databases. The databases selected for this study include IEEE Xplore, ACM Digital Library, SpringerLink, ScienceDirect, and arXiv, as they cover a wide range of high-quality literature in software engineering, machine learning, and accessibility research. These sources are widely accepted in academia and regularly publish peer-reviewed work in the areas relevant to this study.

A well-defined search string was constructed using a combination of keywords and Boolean operators to maximize the recall of relevant studies. The search terms included "machine learning", "accessibility testing [2]", "automated test generation", "accessible software", "software testing", and "assistive technology". The following example of a search string was commonly used:

("machine learning" OR "ML") AND ("accessibility testing [2]" OR "accessible software") AND ("automated testing" OR "test generation")

This query was adapted slightly for each database to fit their individual syntax and search capabilities. The time frame for the search was limited to the period 2010 to 2024, to capture recent developments while ensuring a wide coverage of the literature.

Using the above strategy, we initially retrieved 1,175 research articles from all databases combined. These articles were collected into a reference management tool (Zotero) to organize them efficiently and remove any duplicates. A total of 238 duplicate entries were identified and eliminated during this step, reducing the total number to 937 unique studies.

The next stage involved a preliminary screening of titles, abstracts, and keywords. Articles that clearly fell outside the scope of our study—such as those focusing purely on hardware, non-ML-based testing, or accessibility in unrelated domains (e.g., architecture or transport systems)—were excluded. After this screening, approximately 200 studies were shortlisted for full-text analysis.

This initial selection phase played an essential role in setting the foundation for the later stages of filtering and detailed classification. It ensured that only those studies which had a substantial focus on both machine learning and accessibility testing [2] were carried forward, enabling a high-quality and targeted mapping of the research landscape.

Table 1: Summary of Study Selection Across Databases

Notes:

Initial duplicates were 238, which were removed using Zotero or similar reference management software.

Screening was based on titles, abstracts, and keywords.

Final 70 studies were selected after full-text filtering using strict inclusion/exclusion criteria.

Selection filtering

After the initial selection of relevant studies through keyword-based searches and title-abstract screening, the next essential step in our systematic mapping study was Selection Filtering. This process was designed to ensure that only the most relevant, high-quality, and context-specific research papers were included in the final analysis. The aim was to refine the study pool and eliminate papers that did not meet the inclusion criteria or failed to provide sufficient methodological or technical depth related to our research topic.

1. Inclusion Criteria

We defined a set of inclusion criteria to select only those papers that could directly contribute to answering our research questions. The inclusion criteria were:

Studies that explicitly focus on the application of machine learning techniques to accessibility testing [2] or automated test generation aimed at improving accessibility.

Research published in peer-reviewed journals, conferences, or workshops.

Papers that describe either a proposed method, framework, or empirical evaluation.

Studies published in English, as multilingual processing was not in the scope of this study.

Publications from the year 2010 to 2024 to capture the evolution of recent technologies.

2. Exclusion Criteria

Along with inclusion rules, we defined exclusion criteria to eliminate irrelevant or low-quality studies. These were:

Articles that discuss accessibility without any reference to machine learning or automation.

Papers that use ML in testing but do not address accessibility concerns.

Publications such as editorials, position papers, presentations, book chapters, or tutorials with no experimental data or technical methodology.

Studies that were duplicate, or whose full text was unavailable despite being indexed.

Research in non-software domains (e.g., civil accessibility, medical diagnosis, transportation) even if ML or accessibility terms appeared in the abstract.

3. Filtering Outcome

After applying the above criteria to the 200 shortlisted studies from the initial screening, a full-text review was conducted for each paper. Each study was examined for its relevance, depth, and contribution to the intersection of accessibility testing [2] and ML. During this phase, 130 studies were excluded due to lack of relevance, insufficient technical detail, or failure to meet the inclusion requirements. The remaining 70 studies were finalized for inclusion in the mapping study.

Table 2: Selection Filtering Criteria and Outcome

Explanation:

Each step applies increasingly strict filtering.

Papers were only included in the final set if they satisfied all filtering stages.

Final dataset of 70 papers used for classification and mapping.

OBJECTIVES

The primary aim of this research is to systematically explore and map the existing landscape of applying machine learning (ML) in software accessibility testing [2]. Given the growing emphasis on inclusive design and the potential of ML to automate and enhance testing processes, this study offers insights into the convergence of these domains.

Specifically, this study seeks to:

- Identify and categorize existing research integrating ML into accessibility testing [2], including issue detection and automated test generation.

- Classify ML approaches used (e.g., supervised, unsupervised, deep learning) and their application areas.
- Examine the accessibility issues addressed, and assess the effectiveness of ML methods in resolving them.
- Analyze the datasets, tools, and frameworks used in the studies for their relevance, availability, and replicability.
- Highlight current trends, limitations, and future research opportunities in ML-based accessibility testing [2].

To identify and categorize existing research that integrates machine learning into accessibility testing [2], including automated test generation, issue detection, and repair mechanisms.

To classify the types of machine learning approaches (e.g., supervised, unsupervised, deep learning) used in accessibility testing [2], along with their common use cases.

To analyze the accessibility issues addressed (e.g., visual, auditory, cognitive impairments) and the extent to which machine learning techniques are used to detect or resolve them.

To investigate the datasets, tools, and frameworks employed in these studies, highlighting availability, reproducibility, and domain-specific applications.

To identify current research trends, gaps, and challenges in the application of ML for accessibility testing [2], providing direction for future academic and industrial research.

By fulfilling these objectives, the study aims to provide a comprehensive overview of the field, facilitate knowledge transfer to researchers and practitioners, and support the development of more inclusive, intelligent software testing solutions.

CONCLUSIONS

This systematic mapping study has reviewed how machine learning (ML) is being utilized to advance accessibility testing [2] in software systems. As accessibility becomes a key quality attribute in modern development, and ML continues to evolve, this intersection presents promising research opportunities.

By following a structured review and selection process, 70 studies published between 2010 and 2024 were analyzed. These works reflect growing interest in ML for tasks like UI accessibility checking, alt-text generation, and screen reader support. Supervised learning and computer vision were among the most common methods.

Despite significant progress, challenges remain in the form of limited annotated datasets [3], lack of standardized evaluation metrics, and immature tool support. Many frameworks are in early stages and need further validation for real-world use.

This work contributes a comprehensive mapping of the field, summarizing common methods, accessibility areas addressed, and gaps in current practices. It is intended to guide future academic and industrial efforts aimed at improving inclusive software through ML-powered testing.

In conclusion, while ML integration into accessibility testing [2] is still maturing, it holds immense potential. Continued advancements in data availability, tooling, and validation techniques are essential to make accessibility-first development both scalable and practical.

REFERENCES

- [1] A. Fontes and G. Gay, "The Integration of Machine Learning into Automated Test Generation: A Systematic Mapping Study," arXiv preprint arXiv:2206.10210, 2022.
- [2] L. Nie et al., "SoK: Detection and Repair of Accessibility Issues," arXiv preprint arXiv:2411.19727, 2024.
- [3] M. Taeb et al., "AXNav: Replaying Accessibility Tests from Natural Language," arXiv preprint arXiv:2310.02424, 2023.
- [4] T. Bostic et al., "Automated Evaluation of Website Accessibility Using A Dynamic Accessibility Measurement Crawler," arXiv preprint arXiv:2110.14097, 2021.
- [5] F. Riccio and P. Tonella, "Testing Machine Learning [1] Based Systems: A Systematic Mapping," Empirical Software Engineering, vol. 25, no. 6, pp. 5193–5254, 2020.
- [6] S. K. Mandava, "AI-Powered Accessibility: Using Machine Learning [1] to Detect and Correct Accessibility Gaps in Web Interfaces," Frontiers in Health Informatics, vol. 13, no. 3, 2024.

- [7] A. R. Vontell, "Bility: Automated Accessibility Testing for Mobile Applications," M.S. thesis, Massachusetts Institute of Technology, 2019.
- [8] G. C. Guerino and N. M. C. Valentim, "Usability and User Experience Evaluation of Natural User Interfaces: A Systematic Mapping Study," IET Software, vol. 14, no. 5, pp. 451–467, 2020.
- [9] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic Mapping Studies in Software Engineering," in Proc. 12th Int. Conf. Evaluation and Assessment in Software Engineering, pp. 68–77, 2008.
- [10] M. I. Jordan and T. M. Mitchell, "Machine Learning [1]: Trends, Perspectives, and Prospects," Science, vol. 349, no. 6245, pp. 255–260, 2015.
- [11] A. Abdul et al., "Trends and Trajectories for Explainable, Accountable and Intelligible Systems: An HCI Research Agenda," in Proc. CHI Conf. Human Factors in Computing Systems, 2018.
- [12] Saxena, K., & Sharma, R. (2015). Efficient Heart Disease Prediction System Using Decision Tree. In Proceedings of the International Conference on Computing, Communication & Automation.
- [13] Maji, S., & Arora, S. (2019). Decision Tree Algorithms for Prediction of Heart Disease. Information Communication Technology Competence Strategy, 40, 447–454.
- [14] Azar, A. T., Elshazly, H. I., Hassanien, A. E., & Elkorany, A. M. (2014). A Random Forest Classifier for Lymph Diseases. Computer Methods and Programs in Biomedicine, 113(2), 465–473.
- [15] Nguyen, C., Wang, Y., & Nguyen, H. N. (2013). Random Forest Classifier Combined with Feature Selection for Breast Cancer Diagnosis and Prognostic. Journal of Biomedical Science and Engineering, 6, 551–560.
- [16] Shahzad, A., Zafar, B., Ali, N., Jamil, U., Alghadhbhan, A. J., Assam, M., Ghamry, N. A., & Eldin, E. T. (2022). COVID-19 Vaccines Related User's Response Categorization Using Machine Learning [1] Techniques. Computation, 10(9), 141.
- [17] Sathyadevi, G. (2011). Application of CART Algorithm in Hepatitis Disease Diagnosis. In Proceedings of the 2011 International Conference on Recent Trends in Information Technology.
- [18] [9] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic Mapping Studies in Software Engineering," in Proc. 12th Int. Conf. Evaluation and Assessment in Software Engineering, pp. 68–77, 2008.
- [19] Hefley, W. E., & Murray, D. (1993). Intelligent User Interfaces. In Proceedings of the 1st International Conference on Intelligent User Interfaces (pp. 3–10).
- [20] Painuli, D., Mishra, D., Bhardwaj, S., & Aggarwal, M. (2021). Forecast and Prediction of COVID-19 Using Machine Learning [1]. In Data Science for COVID-19 (pp. 381–397).
- [21] Mariana, B., & Lucian, D. (2016). Random Forest in Remote Sensing: A Review of Applications and Future Directions. ISPRS Journal of Photogrammetry and Remote Sensing, 114, 24–31.
- [22] Jaing, W. L. (2019). A Study on the Choice on Ward Types Using Decision Tree: An Example of Regional Hospital (Master's thesis, National Chung Cheng University).
- [23] Zhang, J. M., Harman, M., Ma, L., & Liu, Y. (2020). Machine Learning [1] Testing: Survey, Landscapes and Horizons. IEEE Transactions on Software Engineering, 48(1), 1–36.
- [24] Sherin, M., & Supriya, M. H. (2019). A Systematic Mapping Study on Testing Machine Learning [1] Programs. arXiv preprint arXiv:1909.06156.
- [25] Braiek, H. B., & Khomh, F. (2018). On Testing Machine Learning [1] Programs. Journal of Systems and Software, 164, 110542.
- [26] Bansal, Manav, Arpit Chhabra, and Niraj Singhal. "Smart city—shrewd vehicle versatility utilizing IoT." International Journal of Engineering Trends and Technology (2022).
- [27] Arpit Chhabra, Niraj Singhal and Syed Vilayat Ali Rizvi, "A Novel Algorithm of Safe-Route Traversal of Data for Designing the Secured Smart City Infrastructures", International Journal of Engineering Trends and Technology, Vol. 71, Issue. 5, pp. 272-281, 2023.