International Journal of Computer Trends and Technology
ISSN: 2231–2803 / https://doi.org/10.14445/22312803/IJCTT-V72I9P127

Volume 72 Issue 9, 170-175, September 2024

Original Article

AI-Driven Approaches to Improve Accessibility Testing Across IoT Devices

Saurabh Kapoor

Senior Quality Assurance Engineer, Amazon, Virginia, USA.

Corresponding Author : saurabh.kapoor16@gmail.com

Received: 10 August 2024 Revised: 08 September 2024 Accepted: 26 September 2024 Published: 30 September 2024

Abstract *- As IoT-enabled devices and smart technologies become increasingly integrated into daily life, ensuring accessibility for all users, including those with disabilities, is critical. Traditional accessibility testing methods, such as manual testing and automated tools, have scope, accuracy, and scalability limitations, especially for dynamic and context-aware IoT environments. This research paper proposes a novel AI-driven approach to automated accessibility testing that leverages machine learning, Natural Language Processing (NLP), and computer vision techniques to identify and predict complex accessibility issues in real time for IoT devices. By learning from real user interactions and continuously updating its knowledge base, this AI-powered system can provide more context-aware and comprehensive accessibility assessments for connected devices, enhancing user usability. The proposed approach aims to enhance accessibility testing by improving the detection of context-specific issues, reducing dependency on manual testing, and promoting more inclusive IoT environments.*

Keywords *- AI-Driven Accessibility Testing, Smart Home Accessibility, Machine Learning for Accessibility, NLP for Accessibility Evaluation, Computer Vision in Usability Testing.*

1. Introduction

As IoT devices become more widespread, they are integrated into various parts of people's daily lives, such as smart homes, healthcare, urban development, and industrial automation. Dependence on IoT devices like smart speakers, wearables, and connected healthcare devices is growing. This makes it crucial to ensure these technologies are accessible to all users, including those with disabilities. Accessibility is vital to meet legal and ethical standards and promote inclusive design. It ensures that everyone can use technology fairly and equitably. Guidelines like the Web Content Accessibility Guidelines (WCAG) and the Americans with Disabilities Act (ADA) set standards that organizations must follow to make devices inclusive for all users. Meeting these standards helps avoid legal repercussions and improves usability for all users.

Traditional accessibility testing methods, such as manual testing and rule-based automated tools, are commonly used for websites and mobile apps. However, these methods must be revised for dynamic IoT environments involving multimodal, context-sensitive interactions with multiple devices. While automated tools efficiently detect common issues like poor color contrast, they need help to handle IoT systems' complex, dynamic, and context-specific challenges. Manual testing, though thorough, is time-consuming and requires specialized expertise, making it less scalable. This research proposes leveraging Artificial Intelligence (AI) to enhance automated accessibility testing specifically for IoT devices. It addresses these limitations and provides more accurate, adaptive, and comprehensive testing solutions [1].

2. Challenges in Accessibility Testing for IoT

The accessibility of IoT devices presents unique challenges that differ significantly from traditional web and mobile applications. These challenges arise due to the diversity of devices, interaction modalities, and user contexts, such as Multimodal Interactions. IoT devices often require users to interact through various modalities such as voice, touch, gesture, and visual feedback. For example, smart speakers rely heavily on voice commands, whereas wearable health devices may use gestures or touchscreens for input. Ensuring accessibility across all these interaction types, especially for users with varying disabilities, is complex. Another challenge is Dynamic and Context-Sensitive Environments - IoT devices operate in dynamic environments where interaction context can change rapidly. For instance, a smart home setup may involve multiple devices communicating with each other, requiring consistent accessibility across devices. An automated vacuum cleaner interacting with a smart thermostat and an intelligent security system should provide consistent and accessible feedback to users. Testing such dynamic, multi-device interactions is challenging using conventional methods. Also, the IoT devices serve a diverse user base, including individuals with visual, auditory, motor, and cognitive impairments. Testing must account for different scenarios, such as a visually impaired user navigating a smart home system or an elderly user managing health data on a wearable device.

Standard rule-based tools often fail to capture these complex scenarios. There is also the challenge of performing large-scale accessibility testing with manual testing only. Automated tools must be able to scale and provide meaningful feedback across various devices and contexts without overwhelming developers with false positives or superficial issues.

3. Proposed AI-Driven Accessibility Framework for IoT Devices

The proposed AI-driven framework addresses the unique challenges of accessibility testing in IoT environments. Traditional methods focus mainly on static content. However, IoT devices are dynamic and interact in complex ways.

This framework uses AI techniques—like machine learning, natural language processing (NLP), and computer vision to provide a more comprehensive, adaptive, and automated solution for testing accessibility.

It aims to detect issues more accurately, offer real-time feedback, and continuously learn from user interactions. The framework consists of several interconnected components. Each component plays a specific role in testing accessibility across different types of IoT devices. See Fig-1 for details on how this framework could be used to identify the accessibility issues.

3.1. Data Collection & Input Layer

This layer gathers data from IoT devices such as smart speakers, wearable monitors, and smart home systems. It collects voice commands, text inputs, gestures, and user visual feedback. This ensures that all types of interactions are considered. The components consist of:

- Sensors and Input Devices: These include microphones, cameras, touchscreens, and motion sensors embedded in IoT devices. Smart speakers use microphones to capture voice commands. Wearables use touchscreens and accelerometers to detect gestures.
- Data Integration Module: This module collects input data from different devices and formats it into a consistent structure. It ensures that all types of user interactions, such as voice commands or swipe gestures, are recorded and processed correctly.

This layer monitors and captures user interactions with IoT devices in various settings (e.g., smart homes, healthcare). It sends the collected data to the AI modules for analysis.

3.2. NLP Module for Voice and Text Analysis

The NLP module examines voice commands and text inputs to spot issues with language clarity and usability [2]. It has two essential parts:

- Speech Recognition: Converts voice into text. It checks if spoken commands are clear and understandable for users with disabilities. This feature is crucial for users with difficulty using traditional input methods like visual disabilities. The NLP module ensures that commands are recognized accurately, considering variations in speech, accents, and speech impairments.
- Text Simplification: Reviews text for readability. It simplifies complex texts to make it easier for users who have visual disabilities. For example, it can generate alt text descriptions for images.

NLP's integration with machine learning further enables continuous improvement, making IoT devices more intuitive and user-friendly.

3.3. Computer Vision Module for Visual and Gesture-Based Analysis

The Computer Vision module analyses visual content and gestures on smart displays and wearables [3]. Its essential functions include:

- Object Detection: Identifies items on screens, such as buttons or text, to check if they are accessible (e.g., proper size and color contrast). It uses computer vision algorithms to detect screen elements such as buttons or text. It checks for appropriate size, color contrast, and labeling. For example, a smart thermostat checks if temperature control buttons are large and have good color contrast for visually impaired users.
- Gesture Recognition: This module evaluates user gestures to ensure they are correctly detected and easy to perform. For example, a wearable health monitor detects a "double-tap" gesture to start a workout. The module checks if it is correctly recognized and offers feedback if not.
- Screen Layout Evaluation: This check arranges visual elements and ensures the layout is intuitive for users with visual or motor impairments. Look for overlapping elements, poor alignment, or non-accessible items.

This module ensures visual and gesture-based interactions are intuitive and accessible, identifying issues like low contrast or confusing layouts.

3.4. Machine Learning Module for Predictive Issue Detection

The Machine Learning module uses algorithms to predict accessibility problems. It learns from past data and real user interactions to detect patterns and new issues. Supervised Learning Models like Random Forests and SVMs are trained on labeled datasets to detect similarly problems in interactions with new IoT devices.

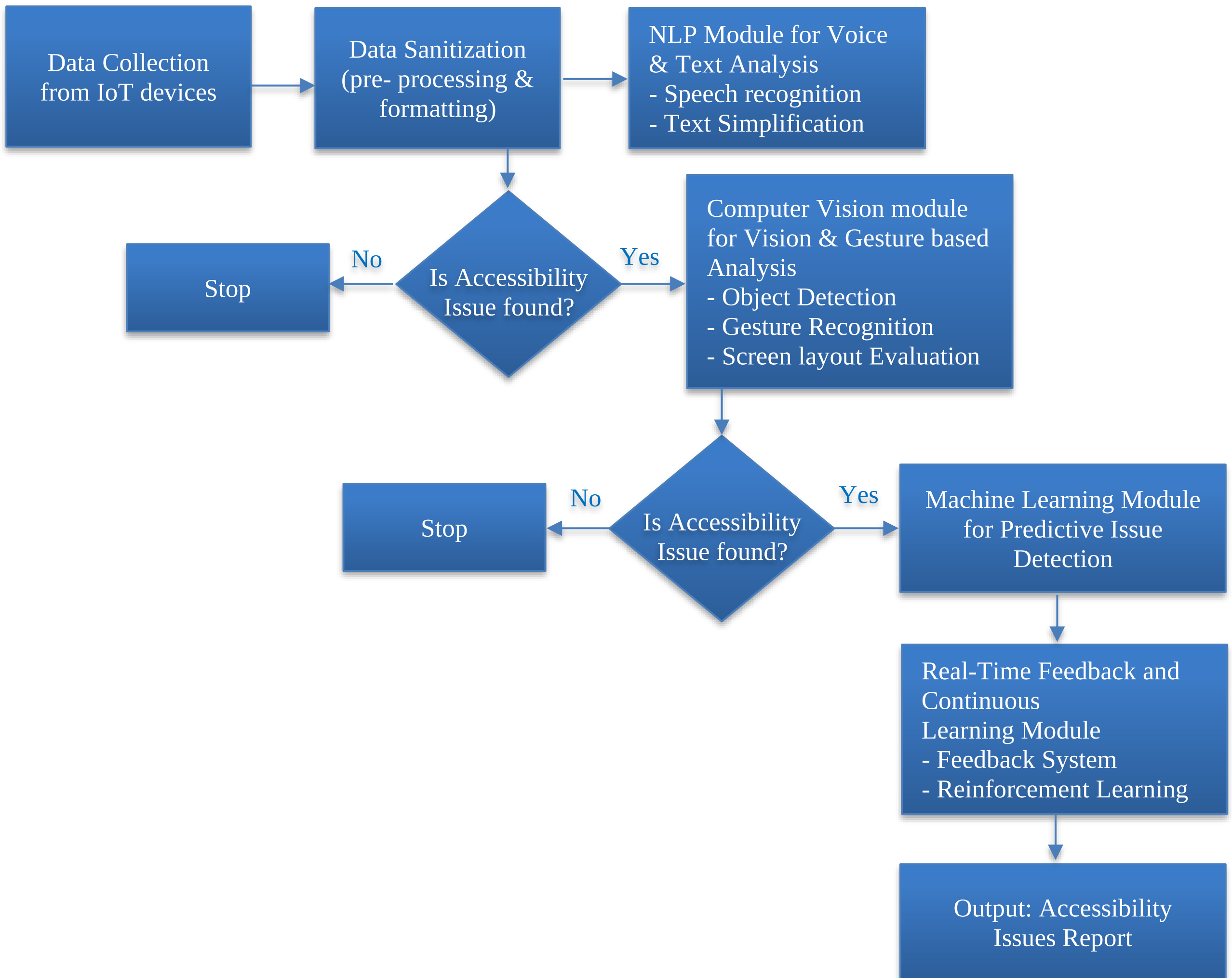

Fig. 1 Framework for identifying accessibility issues

Learning techniques like Clustering and Anomaly Detection could find new accessibility issues that do not fit known patterns [4]. For example, clustering may group user interactions that often cause errors, helping identify new patterns of accessibility problems. The module learns from past data and user feedback to improve its predictions, assisting developers in prioritizing which issues to address.
Deep learning models like CNN (Convolutional Neural Networks) and RNN (Recurrent Neural Networks) can be leveraged to analyze large datasets [5]. For example – CNN can be applied to image data from smart home cameras to detect potential accessibility issues.

3.5. Real-Time Feedback and Continuous Learning Module:

The Real-Time Feedback module gives developers immediate suggestions to fix detected issues, supporting quick updates and continuous improvements. The Continuous Learning framework adapts Reinforcement Learning over time [6] [7]. It learns from feedback and improves its detection accuracy and recommendations as new data comes in. It uses methods like Q-Learning to optimize detection and recommendations. This module keeps the framework updated and adaptive. It helps developers quickly identify and resolve issues, supporting agile development.

Figure 2 illustrates the framework integrating Natural Language Processing (NLP) and Machine Learning (ML) components to enhance accessibility.

4. Framework Evaluation

Evaluating the testing framework involves assessing its performance, robustness, and ability to scale across different IoT environments.

4.1. Test Setup & Scenarios

This includes the core components responsible for data analysis, machine learning, and natural language processing. The setup comprises IoT devices and a test suite of accessibility test cases. These tests consisted of voice command scenarios, visual interaction scenarios, gesture-based interactions, and multi-device interactions.

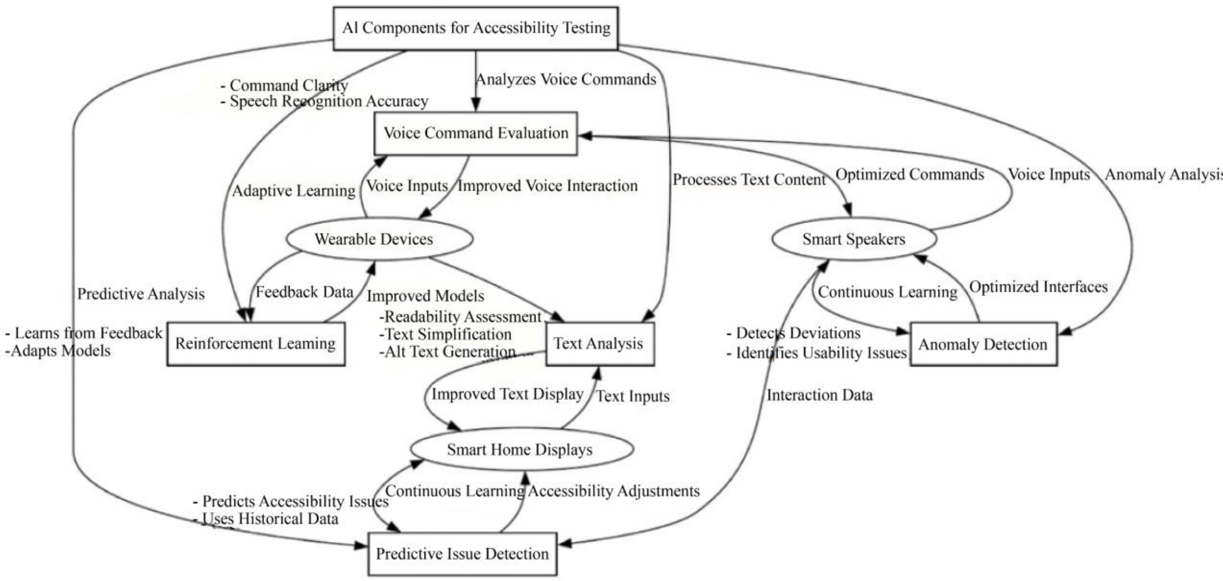

Fig. 2 (AI Components and Data Flow for Accessibility Testing in IoT Devices)

4.2. Metrics

To evaluate the effectiveness of the framework, the following metrics will be considered:

- Accuracy: It measures how correctly the framework detects accessibility issues
- Precision: It measures the framework's ability to avoid false positives.
- Latency & Response Time: Measures the time taken for the system to process data and provide actionable feedback
- Scalability: Measures the framework's ability to maintain performance when the number of devices, data volume, or complexity increases.
- Fault Tolerance: Evaluates how well the framework handles unexpected situations, such as network outages, device failures, or incorrect inputs.

5. System Design & Implementation

This section involves creating a robust architecture that integrates various hardware and software components to ensure efficient and accurate testing of accessibility features. It also discusses the algorithms, libraries, and tools used to develop this framework to test comprehensively and efficiently.

5.1. Architectural Framework

The architectural framework for the system is divided into three layers:

- Device Layer: It consists of IoT devices like smart home systems, wearable devices, and smart speakers. These serve as the input and output points for user interactions.
- Middleware Layer: This layer acts as a bridge between the IoT devices and the computing platform. It handles data aggregation, processing, and integration of NLP and AI modules.
- The application layer includes the core components responsible for data analysis, machine learning, and natural language processing. It processes data inputs, runs a predictive model, and generates actionable insights.

Refer to Figure 3, which depicts the architectural framework for accessibility testing.

5.2. Algorithms and Libraries Utilized

5.2.1. Data Collection & Input Layer

Apache Kafka can be used to collect data from IoT devices like cameras, speakers, and smart home systems. It offers real-time data collection and has low latencies. Data can be stored in Amazon S3. NumPy and Pandas can be used to pre-process data, normalize it, and format it.

5.2.2. NLP Module for Voice & Text Analysis

Speech Recognition could be implemented using Google's speech-to-text API and CMU Sphinx [8]. For Text Simplification, the TextBlob library could parse and convert complex sentences into simple forms [10]. The model can be fine-tuned by training on accessibility datasets.

5.2.3. Computer Vision Module for Visual and Gesture-Based Analysis

YOLO (You Look Only Once) with OpenCV can be used for object detection for the Computer Vision Module [9]. For gesture recognition, TensorFlow and Keras can be used to train deep learning models. The Deep learning models can be trained using COCO (Common Objects in Context) and EgoHands [11].

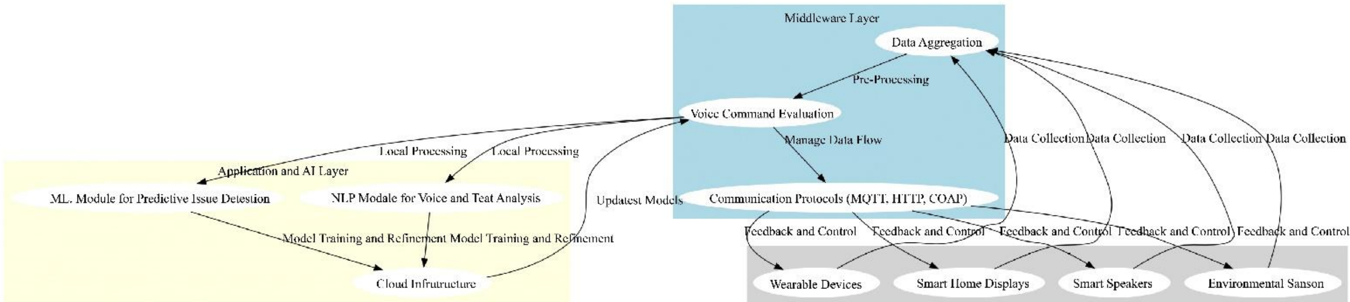

Fig. 3 (Architectural Framework for Accessibility Testing)

5.2.4. Machine Learning Module for Predictive Issue Detection

Random Forests and Gradient-Boosting machine-learning algorithms can be used for supervised learning. Clustering techniques like K-Means or Hierarchical clustering can be used to identify anomalies and patterns [5]. To ensure the model's scalability, it can be deployed on Docker using Kubernetes for orchestration.

5.2.5. Real-Time Feedback and Continuous Learning Module

To incorporate receiving real-time feedback, OpenAI Gymnasium or Stable Baseline can used for reinforcement learning. The reinforcement model can quickly adapt to new data and optimize its recommendations.

6. Conclusion

The proposed AI-Driven Accessibility Testing approach for IoT devices offers a powerful solution. Using AI techniques like NLP, Computer Vision, and Machine Learning, the framework can help perform detailed accessibility testing across IoT devices. The framework's multi-layered architecture (data collection, NLP, visual and gesture-based analysis) provides a robust and scalable approach to accessibility testing. The evaluation of the framework demonstrated its significant advantages over traditional accessibility testing methods with higher detection rates, lower false positives, better time efficiency, and greater adaptability to dynamic environments. It demonstrates the framework's potential to improve accessibility testing, making IoT devices more inclusive for a broader range of users, including those with disabilities.

Conflicts of Interest

The author of this paper has not received any financial compensation. The views and opinions expressed in this paper are solely those of the author and do not reflect those of any person, organization, or employer. The author declares no conflicts of interest regarding the content of this article.

References

[1] Victor Takashi Hayashi et al., "Improving IoT Module Testability with Test-Driven Development and Machine Learning," *2021 8th International Conference on Future Internet of Things and Cloud (FiCloud)*, Rome, Italy, pp. 406-412, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[2] Faustinah Nemieboka Tubo, Ikechukwu E. Onyenweet, and Doris C. Asogwa, "Development of an NLP-Driven Computer-Based Test Guide for Visually Impaired Students," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 12, no. 9, pp. 1-12, 2023 [Cross Ref] [Google Scholar] [Publisher Link]

[3] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Online Object Tracking: A Benchmark," *2013 IEEE Conference on Computer Vision and Pattern Recognition*, Portland, OR, USA, pp. 2411-2418, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[4] Varun Chandola, Arindam Banerjee, and Vipin Kumar, "Anomaly Detection: A Survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1-58, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[5] Alex Graves, Abdel-Rahman Mohamed, and Geoffrey Hinton, "Speech Recognition with Deep Recurrent Neural Networks," *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, BC, Canada, pp. 6645-6649, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[6] Lorenzo Pellegrini et al., "Latent Replay for Real-Time Continual Learning," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, NV, USA, pp. 10203-10209, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[7] Le Lyu, Yang Shen, and Sicheng Zhang, "The Advance of Reinforcement Learning and Deep Reinforcement Learning," *2022 IEEE International Conference on Electrical Engineering, Big Data and Algorithms*, Changchun, China, pp. 644-648, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[8] Abhishek Dhankar, "Study of Deep Learning and CMU Sphinx in Automatic Speech Recognition," *2017 International Conference on Advances in Computing, Communications and Informatics*, Udupi, India, pp. 2296-2301, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[9] Meghana Pulipalupula et al., "Object Detection Using You Only Look Once (YOLO) Algorithm in Convolution Neural Network (CNN)," *2023 IEEE 8th International Conference for Convergence in Technology (I2CT)*, Lonavla, India, pp. 1-4, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[10] Kingsley Kuan et al., "Region Average Pooling for Context-Aware Object Detection," *2017 IEEE International Conference on Image Processing*, Beijing, China, pp. 1347-1351, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[11] Reza Hermansyah, and Riyanarto Sarno, "Sentiment Analysis about Product and Service Evaluation of PT Telekomunikasi Indonesia Tbk from Tweets Using TextBlob, Naive Bayes & K-NN Method," *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)*, Semarang, Indonesia, pp. 511-516, 2020. [CrossRef] [Google Scholar] [Publisher Link]