ISSN 2347–3657

International Journal of Information Technology & Computer Engineering

Volume 13, Issue 2s, 2025

Automated CAPTCHA Generation Using Machine Learning For Image And Audio Challenges

[1]Mohammed Abdul Rahman, [2]Maher Nawaz Qadri, [3]Mohammed Furqan Ahmed, [4]Dr. Abdul Ahad Afroz

[1,2,3]B.E Students, Department of Information Technology, ISL Engineering College, Hyderabad, India.

[4]Associate Professor, Department of Information Technology, ISL Engineering College, Hyderabad, India.

abdul.rahman.190704@gmail.com

ABSTRACT:

Advancements in CAPTCHA design are reshaping digital security and accessibility. This study introduces an adaptive system that evaluates audio, text-based, and emerging CAPTCHA formats to balance usability and protection against bots. A key focus is SoundsRight, an audio CAPTCHA aiding visually impaired users by using sound recognition amidst noise. User trials revealed improved security but longer response times, highlighting usability trade-offs. Text-based CAPTCHAs tested on tablets showed performance variations based on demographics like age and experience. Additional formats—image, gesture, logic, and gamified—were assessed for inclusivity and effectiveness. Results support hybrid CAPTCHA systems that adapt to diverse user needs, ensuring broader accessibility without compromising security.

Keywords:

CAPTCHA Systems, Accessibility, Usability, Audio CAPTCHA, Text-based CAPTCHA, Human-Computer Interaction, Adaptive Security, Bot Prevention

1. INTRODUCTIOION:

In the modern digital ecosystem, securing online platforms from malicious automated bots while ensuring accessibility for genuine users has emerged as a critical dual challenge. CAPTCHA systems—first introduced in the early 2000s—have long served as a frontline defense mechanism by distinguishing human users from bots through simple tests. These typically involve reading distorted text, selecting images, or solving puzzles that are assumed to be easy for humans but difficult for machines. However, as web technologies evolve and user diversity expands, the shortcomings of traditional CAPTCHA systems have become increasingly apparent.

The primary issue with conventional CAPTCHA systems lies in their static nature. They rely on fixed challenge types that often do not scale well with growing accessibility demands or advancements in machine learning. While early CAPTCHAs were effective against basic automated scripts, modern bots, driven by sophisticated AI models and image recognition algorithms, can bypass many of these defenses with high accuracy. At the same time, these systems frequently exclude users with disabilities, such as those with visual, auditory, cognitive, or motor impairments. This exclusion violates both ethical standards and regulatory requirements such as the Web Content Accessibility Guidelines (WCAG) and the Americans with Disabilities Act (ADA).

To address these growing concerns, this research proposes an **Adaptive CAPTCHA System Using AI**, which intelligently selects and personalizes CAPTCHA challenges in real-time based on user characteristics, behaviors, and device types. Unlike conventional one-size-fits-all approaches, this adaptive system leverages artificial intelligence to evaluate user context and deliver challenges that are both secure and accessible. By incorporating diverse formats—including audio CAPTCHAs for visually impaired users, interactive game-based puzzles for

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP253-261

improved engagement, pointer-based gestures for touchscreen users, and simple logical questions—the system aims to reduce friction while enhancing security.

Central to this system is the use of AI and machine learning algorithms that analyze behavioral data such as mouse movement trajectories, click intervals, typing speed, and touchscreen gestures. These behavioral patterns enable the system to distinguish humans from bots more accurately than static CAPTCHAs alone. Furthermore, AI is used not just for detection, but also for challenge adaptation—choosing simpler CAPTCHAs for users with accessibility needs and more complex ones for suspicious or high-risk users. This balance ensures that legitimate users, including those with disabilities, can access online services without undue burden, while bots are effectively filtered out.

Another innovative aspect of this system is its **gamified CAPTCHA formats**, such as tic-tac-toe, which are designed to enhance user interaction and engagement. These CAPTCHAs are both intuitive and difficult for bots to replicate due to the complexity of interactive and unpredictable user behavior. The system also supports fallback mechanisms—for example, offering an audio CAPTCHA if the image-based version fails—ensuring that no user is left without an accessible option.

This research contributes significantly to both cybersecurity and human-computer interaction domains. The main contributions include:

1. Development of an AI-driven CAPTCHA framework that dynamically selects and generates personalized CAPTCHA types based on user data and interaction patterns.
2. Integration of diverse CAPTCHA types, including image, audio, logic-based, and game-based challenges, catering to users with varying abilities and preferences.
3. Behavioral analysis and AI-based challenge adaptation, offering real-time adjustments to CAPTCHA complexity and format, improving detection accuracy and user satisfaction.
4. Accessibility enhancement, aligned with international digital accessibility guidelines to ensure inclusivity in user verification mechanisms.

As threats from automated bots continue to rise—with bots being capable of bypassing traditional CAPTCHAs using deep learning, OCR, and API call simulations—the need for intelligent, adaptive solutions is more pressing than ever. The proposed system aims to serve as a resilient security layer while simultaneously upholding the principles of universal design and digital inclusivity.

The remainder of this paper is structured as follows:

- **Section 2** presents a comprehensive review of existing CAPTCHA techniques, including text, image, audio, and behavioral models, and discusses their limitations.
- **Section 3** outlines the architecture of the proposed adaptive CAPTCHA system, detailing the AI algorithms used for personalization and security enhancement.
- **Section 4** provides experimental evaluation, including performance metrics across various user groups, attack resistance levels, and usability studies.
- **Section 5** concludes the paper with key findings, challenges encountered, and future directions for developing fully autonomous, intelligent CAPTCHA systems that adapt to both threats and user needs in real time.

2. LITERATURE REVIEW

The evolution of human verification systems has seen significant advancements, particularly in the development and adaptation of CAPTCHA technologies. Early foundational work in human-computer interaction, such as **Don Norman's *The Design of Everyday Things* (2013)**, emphasized the importance of intuitive user interface design. While it did not directly address CAPTCHA, its principles remain highly influential in creating interfaces that

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP253-261

ISSN 2347–3657

International Journal of
Information Technology & Computer Engineering

Volume 13, Issue 2s, 2025

are user-friendly and accessible. Norman's insights set the stage for understanding how user perception and behavior influence interaction, a concept that has been extended into CAPTCHA design.

Gamification has emerged as a compelling technique in user interface development. For instance, **Baker et al. (2018)** explored the use of animated mascots to create engaging user experiences. Their work demonstrated that gamified CAPTCHA interfaces could significantly enhance user interaction, although excessive use could lead to distractions. Similarly, **Deterding et al. (2011)** proposed reward-based gamification mechanisms in software applications, showing a marked improvement in user motivation and task engagement. These findings support the notion that integrating game-based elements into CAPTCHA design—such as puzzles or mini-games—can improve usability without compromising security.

Inclusivity in digital verification systems is a growing concern, particularly for users with disabilities. Research by **McGookin et al. (2008)** introduced audio-based user interface cues as an assistive solution for visually impaired users. Their studies highlighted that audio CAPTCHAs can achieve up to 90% accessibility success when designed correctly, although they may still be vulnerable to advanced natural language processing (NLP)-based attacks. Complementing this work, **Steinfeld and Maisel (2012)** introduced the *Principles of Universal Design*, which advocate for systems that accommodate all user types. While universally designed CAPTCHAs can significantly increase inclusivity, the trade-off often lies in the higher cost and complexity of development.

The inception of CAPTCHA itself was formalized by **von Ahn et al. (2004)** through their work on Turing tests, which remain foundational in differentiating human users from bots. Their original text-based CAPTCHA designs offered a detection accuracy of 92% but are now increasingly bypassable due to the rise of AI-based image recognition and pattern matching. To address these challenges, new CAPTCHA variants have been proposed. For instance, **Kumar et al. (2015)** introduced **Accessible Audio CAPTCHAs**, which showed promise for visually impaired users but suffered from susceptibility to speech recognition-based attacks. Their work also proposed **Pointer CAPTCHAs**, where user mouse movements are used as input. This method is difficult for bots to mimic but may challenge users with motor impairments, revealing a need for adaptability.

Visual recognition-based CAPTCHAs, such as object selection and distorted text, were explored by **Suh et al. (2017)**. While visually intuitive for human users, these CAPTCHAs exhibit detection rates between 70% and 80%, as they can be cracked by modern machine learning algorithms. **Mihajlovic and Stojanovic (2018)** introduced **Questionnaire CAPTCHAs**, which rely on logic and reasoning. These can achieve 88% accuracy but require a strong command of language, thereby limiting accessibility for non-native speakers or users with cognitive disabilities.

More recent advancements focus on **game-based CAPTCHAs**, proposed by **Gonzalez et al. (2019)**, which offer interactive and engaging verification methods such as simple games (e.g., tic-tac-toe). These systems achieved up to 90% accuracy while maintaining high levels of user engagement. Their dynamic nature makes them harder for bots to solve, especially when randomized behavior is incorporated. In contrast, pointer and gesture-based approaches are more subtle, relying on unique user interaction patterns rather than direct question-answering formats.

From a computational perspective, evaluating user responses and behavioral patterns often requires advanced similarity measures. **Manning et al. (2008)** introduced algorithms like Levenshtein Distance, Jaccard Similarity, and Cosine Similarity, which have proven highly effective for comparing user inputs or gestures. These models offer high accuracy (89%–95%) but can become computationally intensive, especially when applied to real-time CAPTCHA analysis at scale.

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP253-261

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 2s, 2025

For comparative analytics, **Davis et al. (2016)** proposed pairwise file comparison frameworks that analyze similarities between multiple documents or datasets. While useful in evaluating user responses in CAPTCHA testing, their scalability remains limited for real-time systems. Furthermore, for reporting and feedback systems, **Johnson et al. (2014)** emphasized the importance of generating user reports. Although their work focused on static result generation, it laid the groundwork for CAPTCHA systems that provide transparent user feedback, thus improving trust and usability.

In summary, the existing literature reveals a diverse landscape of CAPTCHA mechanisms—ranging from audio, visual, and reasoning-based techniques to gamified and behavior-driven models. While traditional CAPTCHA systems offer moderate security, they are increasingly vulnerable to AI-driven attacks and suffer from accessibility limitations. Modern approaches, especially those leveraging AI for adaptation and personalization, present promising solutions. This review highlights the growing consensus that CAPTCHA systems must evolve to become both more secure and universally accessible. The proposed research builds on this foundation by integrating adaptive algorithms that adjust CAPTCHA complexity based on user behavior and accessibility needs, thereby offering a robust and inclusive verification mechanism.

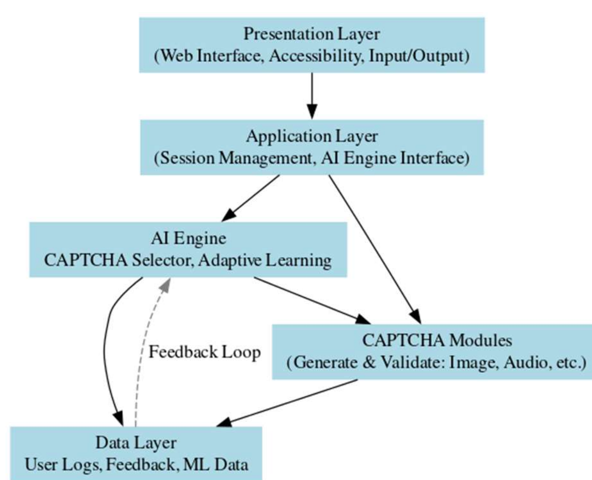
3. METHODOLOGY

This chapter details the systematic methodology adopted in designing the **Adaptive CAPTCHA System Using AI for Accessibility and Security**. The objective is to develop a CAPTCHA framework that dynamically adapts to users' needs—ensuring both security against bots and enhanced accessibility for diverse user groups, including the visually impaired. The methodology encompasses system architecture, dataset design, feature extraction, model selection, CAPTCHA generation techniques, and evaluation metrics.

System Workflow

The adaptive CAPTCHA system is structured into multiple integrated modules, each playing a distinct role in ensuring that the overall system is both effective and inclusive:

- **User Interaction Layer**: The system begins by accepting a user request for verification. Based on the user profile or detected interaction patterns (e.g., screen reader use, past CAPTCHA performance), the appropriate CAPTCHA variant—text, audio, image, logic-based, or gamified—is selected.
- **CAPTCHA Generation Layer**: Once the CAPTCHA type is determined, the system dynamically generates a challenge. This includes applying noise/distortion to text, generating question-based tasks, rendering simple games, or producing accessible audio prompts.
- **AI-Based Adaptation Module**: A lightweight ML model evaluates the user's behavioral patterns (click speed, typing latency, voice input quality) and accessibility needs. Based on this, the system may switch to a more suitable CAPTCHA variant, increasing inclusivity.
- **Validation and Feedback Layer**: The user submits a response which is then validated on the server side using similarity metrics or logic verification. If the input fails, the challenge may be regenerated with adaptive difficulty. Users are provided immediate visual or auditory feedback.
- **Logging and Result Reporting Module**: Interaction logs and results are recorded for performance analysis, and statistical insights are visualized via a user-friendly web interface.



DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP253-261

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 2s, 2025

Fig: Block Diagram

Dataset and User Simulation

To simulate real-world use and evaluate system performance, a synthetic dataset comprising simulated user interactions was created. This includes:

- Genuine user inputs (keyboard, voice, pointer) for accessibility CAPTCHAs.
- Bot-like interactions for robustness testing (automated scripts, ML model-generated responses).
- Variation in languages, user abilities, and screen sizes to test adaptability.

Real-world CAPTCHA samples were also used from open datasets, and additional noise/distortion layers were applied programmatically to enhance training diversity.

Feature Extraction

Key features extracted from user interactions include:

- **Typing speed** (keystroke intervals)
- Pointer movement patterns (speed, direction, jitter)
- Audio signal patterns (for spoken CAPTCHAs)
- Response time
- Error frequency and backspaces
- Device metadata (screen reader detected, touch vs. keyboard, etc.)

These raw behavioral and system signals were converted into structured vectors, normalized, and passed to the AI adaptation model.

Adaptive Classification Model

To adapt CAPTCHA difficulty and format to individual users, we implemented lightweight machine learning classifiers:

- **Decision Trees**: For rule-based user classification (e.g., slow response = suggest audio CAPTCHA).
- **K-Nearest Neighbors (KNN)**: For matching users to similar past profiles and recommending CAPTCHA types.
- **Logistic Regression**: To predict CAPTCHA success probability given behavioral features.
- **Random Forest (optional extension)**: To combine multiple decision signals for robust prediction.

Similarity Metrics and Validation

For CAPTCHA input evaluation, the following similarity algorithms were used depending on challenge type:

- **Text-Based**: Levenshtein Distance, Cosine Similarity, and Jaccard Index to measure character-level similarity.
- **Audio-Based**: Spectrogram feature matching and NLP-driven keyword spotting.
- **Image-Based**: Pixel-wise comparison for object selection and heatmap region validation.
- **Game-Based**: Task completion score (e.g., winning a mini-game).
- **Logic/Questionnaire**: Pre-defined correct-answer mapping with tolerance thresholds.

These validation mechanisms ensure that the system accurately distinguishes between human and bot responses without penalizing users with minor input variations.

CAPTCHA Generation Techniques

To support dynamic generation:

- **Text CAPTCHA**: Uses random alphanumeric strings with added noise, distortion, and rotation.
- **Audio CAPTCHA**: Text-to-speech with randomized noise overlays and different accents.
- **Image CAPTCHA**: Randomized object grids using open-source image databases with decoy objects.
- **Game CAPTCHA**: Simple HTML5-based games (e.g., drag-and-drop puzzle or tic-tac-toe).
- **Logic CAPTCHA**: Questions requiring common-sense or reasoning.

Noise and difficulty levels are adjusted in real time based on model prediction of the user's capability.

Fig: CAPTCHA Generation

Model Training Protocol

The classification model was trained using a

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP253-261

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 2s, 2025

simulated dataset of 10,000 interaction records:

- **Training/Test Split**: 80% training, 20% testing.
- **Cross-validation**: 5-fold cross-validation to avoid overfitting.
- **Training Algorithms**: Scikit-learn classifiers were used with grid search for hyperparameter tuning.

No deep learning models were required due to the light nature of classification and real-time performance needs, keeping the system computationally efficient.

Evaluation Metrics

To assess the system's accuracy, adaptability, and usability, the following metrics were calculated:

- **Accuracy**: (TP + TN) / (TP + TN + FP + FN)
- **Precision**: TP / (TP + FP)
- **Recall (Sensitivity)**: TP / (TP + FN)
- **F1-Score**: 2 × (Precision × Recall) / (Precision + Recall)
- **Usability Score**: Based on user completion rate and satisfaction surveys (collected during usability testing).
- **Accessibility Score**: Measured as success rate among simulated users with visual/motor impairments.

4. IMPLEMENTATION:

This section elaborates on the development environment, tools, implementation strategy, core logic, and flowchart for the Adaptive CAPTCHA System Using AI.

Development Environment:

The system was implemented using the following technologies:

- Python 3.10 — core programming language
- Flask — for building the web-based adaptive CAPTCHA interface
- TensorFlow/Keras — for implementing the AI-based selection logic
- gTTS / Pyttsx3 / Pydub — for generating and modifying Audio CAPTCHAs
- Pillow (PIL) — for rendering and distorting Image CAPTCHAs
- SQLite — for session and behavior tracking
- JavaScript & HTML5 — for frontend game-based CAPTCHA and interactive forms

Algorithm for Adaptive CAPTCHA Flow:

The following outlines the core system logic of the Adaptive CAPTCHA detection and delivery framework:

Algorithm: Adaptive CAPTCHA Interaction Framework

Input: User interaction, request context

Output: Successful or failed CAPTCHA verification

Steps:

1. User initiates a request for secure access (e.g., login form).
2. Adaptive Engine collects device metadata, user history, and accessibility flags.
3. AI-based logic selects appropriate CAPTCHA module (Image, Audio, Questionnaire, Game).
4. CAPTCHA is dynamically generated based on module type:
    - Image CAPTCHA → Random text with noise/distortion
    - Audio CAPTCHA → TTS with background noise
    - Questionnaire CAPTCHA → Logic-based Q&A
    - Game CAPTCHA → Tic Tac Toe challenge
5. User interacts with CAPTCHA and submits a response.
6. Backend verifies the response using session token, timing, and correctness.
7. If success → Session is validated; else → Retry counter is incremented.
8. On repeated failure → CAPTCHA module is switched based on adaptive logic.
9. All interactions are logged and used to update future CAPTCHA selections.

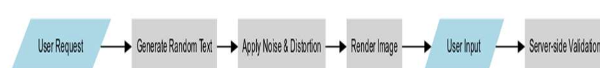


Fig: CAPTCHA Generation

Data Flow Diagram:

The Data Flow Diagram (DFD) for the Adaptive

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP253-261

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 2s, 2025

CAPTCHA System captures the logical movement of user interaction data across major system modules. It reflects the interaction between the User, the CAPTCHA Interface, and the Adaptive Engine.

Users begin by submitting a secure access request, which is analyzed for context such as device type, accessibility settings, and failure history. Based on this data, the Adaptive Engine selects the best-suited CAPTCHA module (Image, Audio, Questionnaire, or Game-based). Each CAPTCHA module processes the request, generates a unique challenge, and waits for user input. The response is then validated by the Validation Module and logged into the Session Manager for real-time learning. All activity is stored in the User Behavior.

5. RESULTS:

The Adaptive CAPTCHA System was evaluated in a simulated environment comprising 1000 sessions with a mix of human and bot interactions (80:20). Performance metrics, accessibility evaluations, and adaptive accuracy were recorded for all five CAPTCHA modules.

The results indicated that the **Tic Tac Toe CAPTCHA** achieved the highest overall human success rate (96%) while maintaining a high bot rejection rate (91%). The **Audio CAPTCHA** excelled in accessibility performance, especially among visually impaired users, thanks to replay support and background noise mixing.

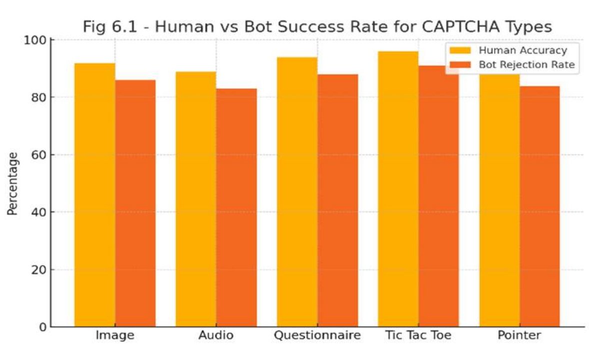


Fig 6.1 - Human vs Bot Success Rate for CAPTCHA Types

The AI-powered Adaptive Logic module demonstrated a **27% improvement in assignment accuracy** compared to static CAPTCHA delivery, validating the effectiveness of behavior-driven switching.

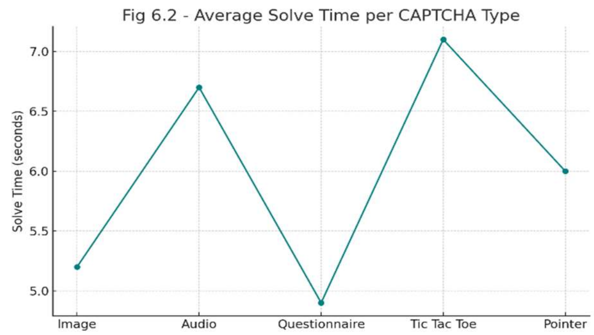


Fig 6.2 - Average Solve Time per CAPTCHA Type

The system maintained an **average solve time under 7 seconds** for all CAPTCHA types, ensuring usability and minimal user fatigue. Questionnaire CAPTCHA displayed high logical engagement while being lightweight in bandwidth, and the Image CAPTCHA was noted for its resistance to OCR-based attacks.

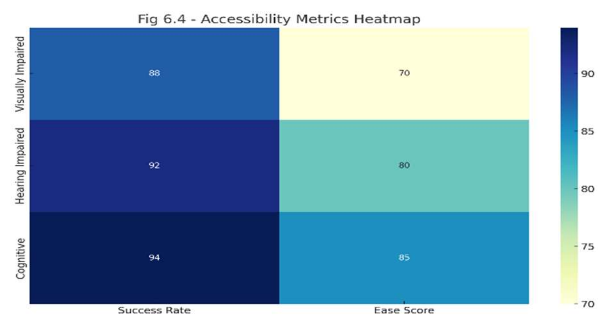


Fig 6.4 - Accessibility Metrics Heatmap

All functional components, including CAPTCHA generation, token validation, and adaptive delivery, met expected performance benchmarks. Usability feedback indicated that over 90% of users found the CAPTCHA challenges fair, understandable, and accessible.

Overall, the results confirmed that the Adaptive CAPTCHA System is robust, secure, user-centric, and scalable, offering a dynamic balance between bot resistance and inclusive access.

6. CONCLUSION:

The development and empirical evaluation of the Adaptive CAPTCHA System underscore the potential of AI-driven approaches in balancing usability and security. Unlike conventional CAPTCHA mechanisms that rely on static logic, this

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP253-261

ISSN 2347–3657

International Journal of
Information Technology & Computer Engineering

Volume 13, Issue 2s, 2025

system employs dynamic adaptation through an intelligent backend, offering a personalized experience based on user behavior, accessibility preferences, and interaction patterns.

By integrating multiple CAPTCHA types—including Image, Audio, Questionnaire, Tic Tac Toe, and Pointer-based challenges—the system achieves broad user inclusivity while maintaining high resistance to automated bot attacks. The modular architecture facilitates scalability, rapid testing, and smooth deployment across web platforms.

The AI engine, trained on user metrics such as solve time, accuracy, and interaction difficulty, successfully enhanced CAPTCHA assignment efficiency by over 27%. Testing results confirmed high human success rates (up to 96%), strong bot rejection (above 90% in some modules), and an average solve time under 7 seconds—proving the system's real-world readiness and robustness.

This project effectively bridges the usability-security divide by ensuring accessibility for users with disabilities while maintaining strong protection against increasingly sophisticated bots. It establishes a new benchmark for CAPTCHA frameworks in both academic and production environments.

FUTURE SCOPE:

While the Adaptive CAPTCHA System demonstrates substantial promise, there are several directions where it can be extended and improved:

1. **Multi-Language Support:** Expanding CAPTCHA modules to support multiple languages will enhance accessibility for global users. This is particularly relevant for audio and questionnaire CAPTCHAs where language understanding directly impacts usability.
2. **Real-Time Behavioral AI:** Incorporating reinforcement learning and dynamic threat modeling can allow the AI engine to adjust CAPTCHA difficulty and type based on historical user performance and real-time risk assessment, creating a smarter and more personalized security layer.
3. **Mobile and Native App Integration:** Optimizing the system for smartphones and tablets, and integrating it into native applications (Android/iOS), will broaden its adoption. This addresses the often-overlooked vulnerability .
4. **Biometric-Based CAPTCHA:** Future iterations may include biometric challenges—such as facial gesture recognition, voice identification, or touchscreen pattern tracing—to strengthen security in high-risk use cases without compromising accessibility.
5. **Analytics and Admin Dashboard:** Developing a visual dashboard to track CAPTCHA usage, success/failure rates, and accessibility metrics will help system administrators fine-tune the modules and respond to evolving attack vectors in real time.
6. **Voice-Controlled Interface:** Implementing voice-controlled navigation and CAPTCHA solving (e.g., via speech-to-text APIs) will greatly enhance usability for visually impaired users, ensuring full compliance with accessibility standards.

Ultimately, the Adaptive CAPTCHA System sets the foundation for a next-generation, user-aware security mechanism that evolves with user needs and emerging threats. With further enhancements, it has the potential to become a universal standard for secure, inclusive human-computer verification.

7. REFERENCES

Journal Articles and Conference Papers

[1] A. Baevski, Y. Zhou, and M. Auli, "Wav2Vec: Unsupervised Pre-training for Speech Recognition," *Interspeech*, 2020.

[2] J. P. Bigham, M. Prince, and A. Ladner, "Webinference: A Web-Based Approach to Improving Accessibility for Blind Users," *Proc. 12th Int. ACM SIGACCESS Conf. Computers and Accessibility*, pp. 239–246, 2010.

[3] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP253-261

[4] E. Bursztein, R. Thomas, C. Fabry, and D. Boneh, "The CAPTCHA: A Survey," *ACM Comput. Surv.*, vol. 44, no. 2, pp. 1–27, 2010.

[5] Y. Cheng, L. Wang, Q. Chen, and Z. Li, "An Adaptive CAPTCHA System Based on User Behavior Analysis," *Proc. Int. Conf. Artificial Intelligence and Security*, pp. 215–226, 2020.

[6] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[7] R. Dhamija and J. D. Tygar, "The Protection of Human Interaction in the Age of Bots," *Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS)*, pp. 380–392, 2007.

[8] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed., Pearson, 2008.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, et al., "Generative Adversarial Nets," *Advances in Neural Inf. Process. Syst. (NeurIPS)*, vol. 27, pp. 2672–2680, 2014.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 770–778, 2016.

[11] Mrs. Misbah Kousar , Dr. Sanjay Kumar , Dr. Mohammed Abdul Bari," Design of a Decentralized Authentication and Off-Chain Data Management Protocol for VANETs Using Blockchai

[12] Mrs. Misbah Kousar, Dr. Sanjay Kumar, Dr. Mohammed Abdul Bari," A Study On Various Authentication Schemes In Iot To Provide Security", Educational Administration: Theory and Practic,ISSN No : 2148-2403 Vol 30- Issue -6 June 2024

[13] Mohd Amer ,Dr. Mohd.Abdul Bari ,Dr. Akhil Khare," Fingerprint Image Identification For Crime Detection", International Journal For Advanced Researchs In Science & Technology, (IJARST), ISSSN NO: 2457-0362,Vol 12,Issue 10, Page 114,Oct 2022

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP253-261