GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY

Private University Estd. in Karnataka State by Act No. 41 of 2013

BANGALORE

PRESIDENCY GROUP OVER 40 YEARS OF ACADEMIC WISDOM

A Project Report

On

“ML Model based solution to refine CAPTCHA”

Batch Details

Sl. No.	Roll Number	Student Name
1	20211CSG0026	Gnanavika M
2	20211CSG0035	R Kamal Raj
3	20211CSG0005	Shreyas DM

School of Computer Science,

Presidency University, Bengaluru.

Under the guidance of,

Mr.Yamanappa

School of Computer Science,

Presidency University, Bengaluru

INTRODUCTION

In the digital age, the proliferation of online services has necessitated robust security measures to protect sensitive information and ensure the integrity of user interactions. One of the most common methods employed to safeguard web applications from automated attacks, such as Denial of Service (DoS) and Distributed Denial of Service (DDoS), is the use of Completely Automated Public Turing tests to tell Computers and Humans Apart (CAPTCHA). While effective in thwarting bots, traditional CAPTCHA systems often hinder user experience, leading to frustration and abandonment of services. This is particularly critical for organizations like the Unique Identification Authority of India (UIDAI), which manages the Aadhaar system—a vital identification framework for millions of residents in India.

The UIDAI recognizes the need for a more seamless user experience while maintaining robust security protocols. As such, the organization is exploring innovative solutions that can passively differentiate between human users and automated bots without the need for intrusive CAPTCHA challenges. This shift towards a passive solution leverages advancements in machine learning (ML) and artificial intelligence (AI) to analyze environmental parameters and user interactions in real-time, thereby enhancing user engagement while ensuring the security of its portals.

Introduction to the Domain of the Problem Statement Chosen: The domain of this problem statement lies at the intersection of cybersecurity, user experience design, and machine learning. As online threats evolve, so too must the strategies employed to combat them. Traditional CAPTCHA systems, while effective, are increasingly viewed as barriers to user engagement, particularly in high-stakes environments where user satisfaction is paramount. The challenge is to develop a solution that not only protects against automated attacks but also respects user experience and privacy.

In this context, the proposed solution aims to capture a range of environmental parameters—such as mouse movements, keystroke dynamics, browser characteristics, and interaction patterns—through a compliant frontend interface. These parameters will serve as features for a machine learning model designed to classify user interactions as either human or bot-driven. By employing a passive approach, the solution minimizes the need for direct user interaction, thereby enhancing the overall experience while maintaining security.

The integration of machine learning into this domain presents unique opportunities and challenges. It requires a careful selection of features that can effectively differentiate between human and bot behavior, as well as the development of a robust model that can adapt to evolving threats. Furthermore, adherence to privacy policies is crucial, especially given the sensitive nature of the data handled by UIDAI.

In summary, the proposed solution seeks to redefine the approach to online security by moving away from traditional CAPTCHA systems towards a more sophisticated, user-friendly, and privacy-compliant method of distinguishing between human and automated interactions. This innovative approach not only addresses the immediate security concerns of UIDAI but also sets a precedent for future developments in the realm of online user engagement and security.

LITERATURE REVIEW

In the pursuit of refining CAPTCHA systems and enhancing user experience while ensuring security, various existing methods have been explored. This literature review examines several approaches to bot detection and user verification, highlighting their advantages and limitations.

1.Traditional CAPTCHA systems are widely recognized for their simplicity and effectiveness in blocking automated scripts and bots. They are easy to implement and understood by users, making them a common choice for web applications. However, these systems can frustrate users, leading to service abandonment, and they pose accessibility challenges for individuals with disabilities. Furthermore, as bots become increasingly sophisticated, they are capable of solving traditional CAPTCHAs using machine learning techniques, diminishing their effectiveness.

2.hCaptcha and reCAPTCHA represent advancements in CAPTCHA technology, offering a more user-friendly experience through image recognition tasks. These systems continuously learn from user interactions, improving their effectiveness over time. However, they still require user interaction, which can disrupt the user experience. Additionally, privacy concerns arise from the data collection and user tracking practices associated with these services. Moreover, sophisticated bots can mimic human behavior, rendering these systems vulnerable.

3.Behavioral biometrics is another approach that analyzes user behavior patterns, such as typing speed and mouse movements, to identify anomalies. This method is passive and unobtrusive, enhancing user experience by minimizing direct interaction. However, it requires extensive data collection and analysis, which can raise privacy concerns. Additionally, advanced bots may replicate human behavior, making it challenging to distinguish between genuine users and automated scripts. The implementation of behavioral biometrics can also be complex, with the potential for false positives.

4.Device fingerprinting is a technique that identifies unique devices based on browser and hardware characteristics. This method can be used in conjunction with other security measures to enhance overall protection. However, it raises privacy concerns regarding user tracking across sessions. Users employing privacy tools, such as VPNs or incognito mode, can circumvent device fingerprinting, leading to potential security gaps. Furthermore, frequent device switching by legitimate users can result in false positives.

5.Machine learning-based anomaly detection is a promising approach that leverages historical data to identify patterns indicative of bot behavior. This method is adaptable, allowing for retraining of models with new data to address evolving threats. However, it requires a substantial dataset for effective training, which may not always be available. There is also a risk of

overfitting, leading to poor generalization on unseen data. The complexity of model deployment and maintenance can further complicate the implementation of this approach.

6.JavaScript-based interaction analysis captures user interactions in real-time, providing insights into user behavior. This method can be implemented in a non-intrusive manner, enhancing user experience. However, bots can be programmed to mimic JavaScript interactions, reducing the effectiveness of this approach. Additionally, the dependency on client-side execution can be manipulated by malicious actors, further complicating the detection of automated scripts. The processing and analysis of interaction data may also require significant resources.

7.Challenge-response tests engage users in simple tasks that are easy for humans but difficult for bots. These tests can be designed to be fun and engaging, improving user experience. However, they still require user interaction, which can be a barrier to seamless engagement. Moreover, as bots become more advanced, they are increasingly capable of solving these challenges.

8.IP reputation and rate limiting are straightforward methods that block known malicious IPs and limit requests from suspicious sources. These techniques are simple to implement and can effectively reduce server load. However, legitimate users may be blocked if they share IPs with malicious actors, particularly in corporate networks. Additionally, these methods do not address sophisticated bot attacks that utilize distributed networks, leaving potential vulnerabilities.

9.User account behavior analysis monitors user behavior over time to establish a baseline for normal activity. This method can trigger alerts or additional verification for anomalous behavior, enhancing security. However, it requires a significant amount of historical data to be effective, and privacy concerns arise from the monitoring of user behavior. There is also the potential for false positives, which can create unnecessary friction for users.

10.Multi-factor authentication (MFA) adds an additional layer of security beyond user credentials. This method can significantly reduce the risk of unauthorized access. However, it requires user interaction, which can be cumbersome and may lead to resistance from users who perceive MFA as inconvenient. If not implemented correctly, MFA can also be vulnerable to social engineering attacks.

CAPTCHA Method Drawbacks

User Frustration

Challenge-response tests are often frustrating and time-consuming.

AI Bypass

Traditional CAPTCHAs can be easily solved by AI, reducing security.

Privacy Concerns

ReCAPTCHA raises issues due to reliance on Google services.

Poor Accessibility

Text-based CAPTCHAs are difficult for visually impaired users.

Accessibility Challenges

Image/Audio CAPTCHAs are challenging for disabled users.

OBJECTIVES

Based on the observations and research gaps identified in the literature review, the following objectives have been established for the development of a machine learning model-based solution to refine CAPTCHA for UIDAI's portals:

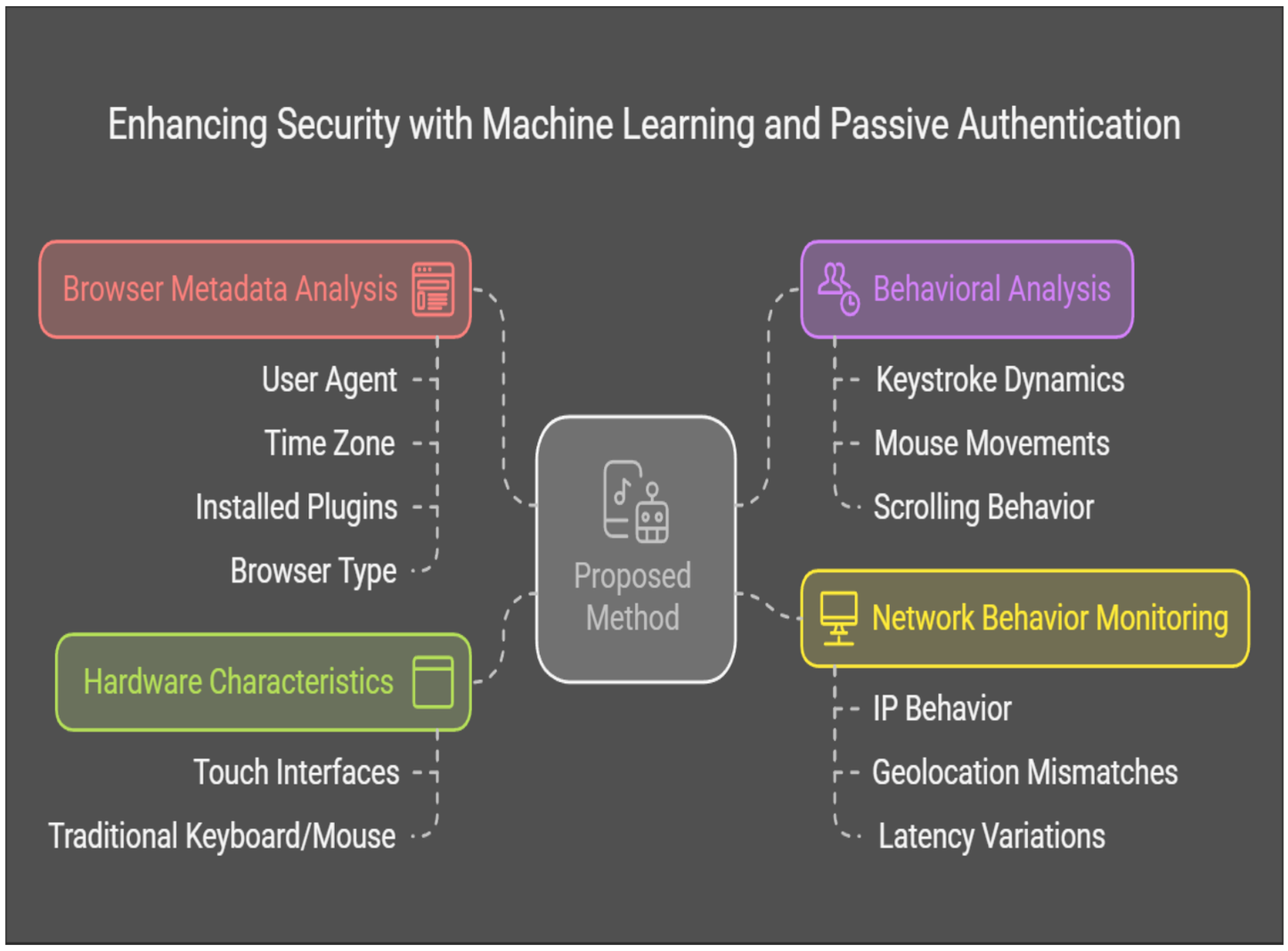
1.Identification and Selection of Environmental Parameters: The primary objective is to define and capture a comprehensive set of environmental parameters that can effectively differentiate between human users and automated bots. This includes parameters such as mouse movement patterns, keystroke dynamics, browser characteristics, and interaction timing. The goal is to ensure that the selected features are robust enough to provide meaningful insights while minimizing the need for user interaction.

2.Development of a Passive Detection Mechanism: The solution aims to create a passive detection mechanism that analyzes the captured environmental parameters in real-time using machine learning algorithms. This mechanism should be capable of accurately classifying user interactions as either human or bot-driven without requiring active user engagement. If the passive analysis is inconclusive, the system should intelligently prompt users for minimal interactions that are non-intrusive and user-friendly.

3.Implementation of a Pluggable Machine Learning Model: A key objective is to develop a machine learning model that is modular and pluggable, allowing for seamless integration with UIDAI's existing application stack. This model should be capable of continuous learning and adaptation to evolving bot behaviors and attack patterns. Additionally, it should be designed to operate efficiently within the constraints of UIDAI's backend infrastructure while ensuring scalability and performance.

4.Adherence to Privacy and Compliance Standards: The solution must prioritize user privacy and comply with UIDAI's core privacy policies. This objective involves implementing data anonymization techniques, ensuring that any captured data is handled securely, and providing transparency to users regarding data collection practices. The solution should also include mechanisms for obtaining user consent where necessary, thereby fostering trust and compliance with regulatory requirements.

METHODOLOGY



(i)Hardware and Software Used

Hardware:

A standard server or cloud-based infrastructure for hosting the backend machine learning model.

User devices (desktops, laptops, tablets, and smartphones) for testing the frontend implementation.

Software:

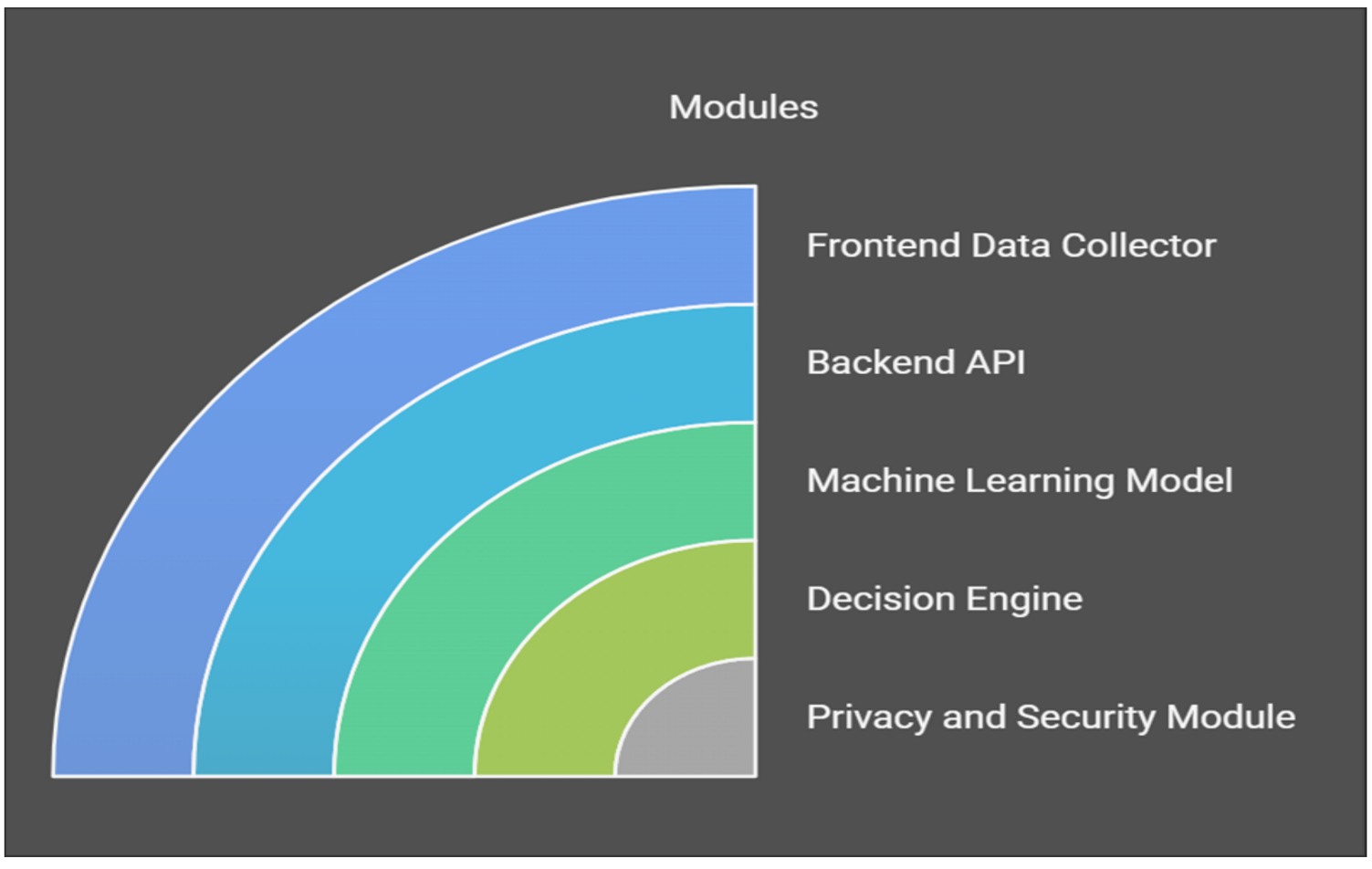
Frontend Framework: React.js or TypeScript for building the user interface to capture environmental parameters.

Backend Framework: Node.js or Python Flask for developing the server-side application.

Machine Learning Libraries:Scikit-learn, TensorFlow, or PyTorch for building and training the machine learning model.

Database: MongoDB or PostgreSQL for storing user interaction data and model outputs.

Version Control: Git for source code management and collaboration.



(ii)DESIGN PROCEDURE

The design procedure for the proposed solution involves several key steps:

1.Requirement Analysis: The first step is to identify the specific requirements for the solution. This includes determining the environmental parameters that need to be captured, such as mouse movements, keystroke dynamics, and browser characteristics. Additionally, privacy policies that must be adhered to will be outlined.

2.Feature Selection:In this step, the environmental parameters that will be captured from user interactions are defined and selected. This may involve analyzing various user behaviors and identifying which parameters are most indicative of human versus bot interactions.

3.Frontend Development: The frontend application will be implemented using the chosen JavaScript framework (e.g., React.js). This application will be responsible for capturing the defined environmental parameters and sending them to the backend for analysis. The design will prioritize user experience, ensuring that the data capture process is seamless and unobtrusive.

4.Backend Development: The backend application will be developed using Node.js or Python Flask. This application will receive the captured data from the frontend, process it, and pass it to the machine learning model for classification. The backend will also handle any necessary data storage and retrieval.

5.Machine Learning Model Development:A machine learning model will be built and trained using the selected libraries. The model will analyze the captured data to classify user interactions as either human or bot-driven. This step will involve selecting appropriate algorithms, training the model on labeled data, and validating its performance.

6. Integration: The frontend and backend components will be integrated to ensure that the machine learning model can be easily plugged into the UIDAI application stack. This will

involve setting up APIs for communication between the frontend and backend, as well as ensuring that the model can be accessed and utilized effectively.

7. Testing and Validation: Thorough testing of the entire system will be conducted to validate its performance and accuracy in differentiating between human and bot interactions. This will include unit testing, integration testing, and user acceptance testing to ensure that the solution meets the specified requirements.

8. Deployment: Once testing is complete and the solution is validated, it will be deployed in a production environment. This deployment will ensure compliance with UIDAI's privacy policies and security standards, allowing for safe and effective use of the system.

9. Monitoring and Maintenance: After deployment, the system's performance will be continuously monitored. This includes tracking the accuracy of the machine learning model and updating it as needed to adapt to new bot behaviors and attack patterns. Regular maintenance will ensure that the system remains effective and secure over time.

By following this methodology, the proposed solution aims to effectively refine CAPTCHA systems for UIDAI's portals, enhancing user experience while maintaining robust security against automated threats.

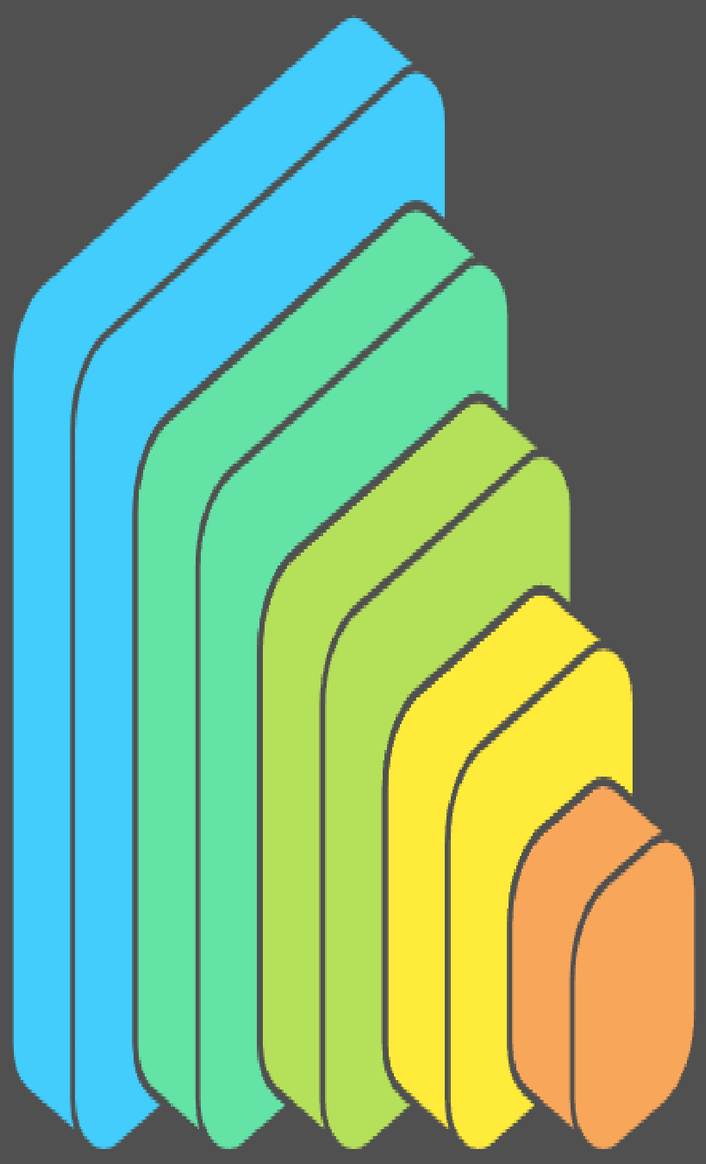
Data Processing and Decision Funnel

Data Capture

Collecting user interactions and environment data

Data Processing

FastAPI processes and prepares data



Similarity Scoring

ML model applies Siamese Networks and KNN

Integration

Ensures compatibility with UIDAI systems

Decision Making

Decision engine implements fallback

Timeline for Execution of Project

Task	Start Date	End Date	Duration
Research & Requirement Analysis	Feb 20, 2025	Feb 26, 2025	1 Week
Data Collection & Feature Selection	Feb 27, 2025	Mar 5, 2025	1 Week
Model Development & Training	Mar 6, 2025	Mar 19, 2025	2 Weeks
Backend API Development	Mar 20, 2025	Mar 26, 2025	1 Week
Frontend Data Collection Implementation	Mar 27, 2025	Apr 9, 2025	2 Weeks
Integration & Testing	Apr 10, 2025	Apr 23, 2025	2 Weeks
Deployment & Final Testing	Apr 24, 2025	May 7, 2025	2 Weeks

Project Timeline Explanation

The project begins with **Research & Requirement Analysis (Feb 20 – Feb 26, 2025)** to understand UIDAI's needs and define key passive detection parameters. This is followed by **Data Collection & Feature Selection (Feb 27 – Mar 5, 2025)**, where environmental and behavioral data will be gathered from users' browser interactions.

Next, the **Model Development & Training (Mar 6 – Mar 19, 2025)** phase involves building and refining the machine learning model to differentiate between bots and humans. Once the model is ready, **Backend API Development (Mar 20 – Mar 26, 2025)** ensures seamless integration with UIDAI's systems, allowing real-time bot detection.

The **Frontend Data Collection Implementation (Mar 27 – Apr 9, 2025)** phase focuses on integrating JavaScript-based trackers in the user interface for passive data collection. Then, **Integration & Testing (Apr 10 – Apr 23, 2025)** ensures that all components work together effectively.

Finally, **Deployment & Final Testing (Apr 24 – May 7, 2025)** ensures system stability and reliability before full-scale implementation. This structured approach ensures an efficient and privacy-compliant passive CAPTCHA solution for UIDAI.

Expected Outcomes

The development of a machine learning model-based solution to refine CAPTCHA for UIDAI's portals is anticipated to yield several significant outcomes that align with the objectives outlined in the problem statement. These outcomes include:

1.Enhanced User Experience: By replacing traditional CAPTCHA with a passive solution, users will experience a smoother and more seamless interaction with the Aadhaar portals. The reduction or elimination of active CAPTCHA challenges will minimize user frustration and abandonment rates, leading to higher engagement and satisfaction.

2.Effective Bot Detection: The solution will successfully differentiate between human users and automated bots by analyzing a comprehensive set of environmental parameters. The machine learning model will be trained to recognize patterns indicative of human behavior, thereby effectively identifying and mitigating potential threats from bots.

3.Minimal User Interaction: In cases where passive analysis is inconclusive, the system will intelligently prompt users for minimal interactions. These interactions will be designed to be non-intrusive and user-friendly, ensuring that the overall user experience remains positive while still providing an additional layer of security.

4.Scalable and Pluggable Architecture: The backend machine learning model will be designed to be modular and pluggable, allowing for easy integration with UIDAI's existing application stack. This architecture will facilitate future updates and enhancements to the model, ensuring that it can adapt to evolving threats and user behaviors.

5.Compliance with Privacy Policies: The solution will adhere to UIDAI's core privacy policies, ensuring that user data is handled securely and transparently. Data anonymization techniques will be implemented to protect user privacy, and mechanisms for obtaining user consent will be established, fostering trust in the system.

6.Robust Security Against DoS/DDoS Attacks: By effectively identifying and blocking bot traffic, the solution will enhance the security of UIDAI's backend APIs against Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks.

7.Continuous Improvement and Adaptation: The system will include mechanisms for continuous monitoring and maintenance, allowing for the ongoing evaluation of the machine learning model's performance. This will enable the model to be updated and retrained as needed to adapt to new bot behaviors and attack patterns, ensuring long-term effectiveness.

8.Comprehensive Documentation and Codebase: The project will result in a complete documentation package that outlines the design, implementation, and functionality of both the

frontend and backend components. The corresponding codebase will be well-structured and maintainable, facilitating future development and collaboration.

By achieving these expected outcomes, the proposed solution aims to provide UIDAI with a robust, user-friendly, and secure alternative to traditional CAPTCHA systems, ultimately enhancing the overall experience for residents engaging with Aadhaar portals.

Conclusion

In conclusion, the development of a machine learning model-based solution to refine CAPTCHA for UIDAI's portals represents a significant advancement in enhancing user experience while maintaining robust security against automated threats. By shifting from traditional CAPTCHA systems to a passive detection approach, this solution addresses the critical need for seamless resident engagement with Aadhaar portals, thereby reducing barriers that hinder user interaction.

The proposed solution effectively captures a comprehensive set of environmental parameters, allowing for the differentiation between human users and bots through sophisticated machine learning algorithms. This passive approach not only minimizes the need for intrusive user interactions but also ensures that any necessary interactions are designed to be minimal and user-friendly, thereby preserving the overall user experience.

Furthermore, the modular and pluggable architecture of the backend machine learning model facilitates easy integration with UIDAI's existing application stack, ensuring scalability and adaptability to evolving threats. The adherence to UIDAI's core privacy policies underscores the commitment to user data protection, fostering trust and compliance with regulatory standards.

The expected outcomes of this project include enhanced user satisfaction, improved security against DoS/DDoS attacks, and a robust framework for continuous monitoring and adaptation of the machine learning model. By achieving these outcomes, the solution not only meets the immediate needs of UIDAI but also sets a precedent for future innovations in online security and user engagement.

Overall, this project demonstrates the potential of leveraging artificial intelligence and machine learning to create intelligent, user-centric solutions that balance security and usability in an increasingly digital world. The successful implementation of this solution will contribute to UIDAI's mission of providing secure and efficient services to residents, ultimately enhancing the overall effectiveness of the Aadhaar system.

References

1. Goodfellow, I., et al. (2014). "Generative Adversarial Networks." https://arxiv.org/abs/1406.2661

2. Doshi, R., et al. (2018). "A Study on Automated CAPTCHA Solving." https://arxiv.org/abs/1805.05910

3. Sivakorn, S., et al. (2016). "CAPTCHA Challenges in Modern Web Security." https://www.usenix.org/conferences/technical-sessions/annual-technical-conference/2016/conference-program/presentation/sivakorn

4. Google Research Papers on ReCAPTCHA. https://research.google/pubs/archive/45530.pdf

5. Various IEEE and ACM published works on bot detection using ML. https://ieeexplore.ieee.org/Xplore/home.jsp https://dl.acm.org/