SMART INDIA HACKATHON - 2024

Problem Statement ID

1672

Problem Statement

Develop a ML Model based solution to refine CAPTCHA

Category

Software

Theme

Smart Automation

Team ID

46424

Team Name

invisCaptcha_2024

Organization

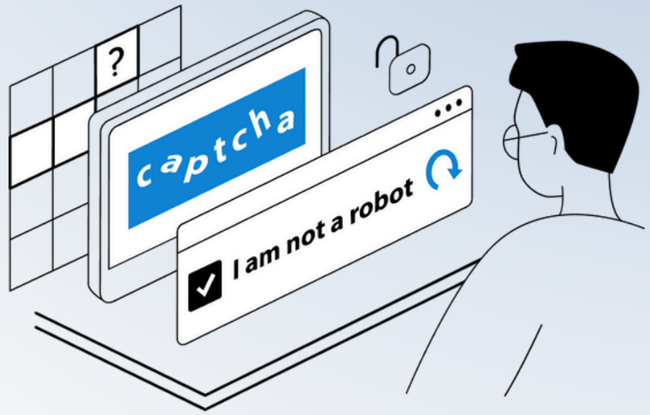
Ministry of Electronics and Information Technology

I'm not a robot invisCAPTCHA Privacy - Terms

Submit

SMART INDIA HACKATHON 2024

invisCAPTCHA



invisCaptcha

invisCAPTCHA

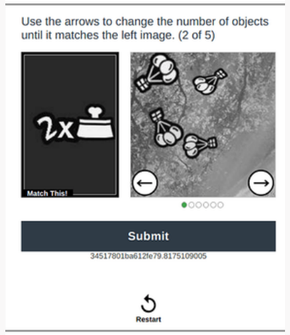
Passive Captcha Defense: Behavioral Analytics and Honeypot Traps with Advanced ML for Bot Detection

SMART INDIA HACKATHON 2024

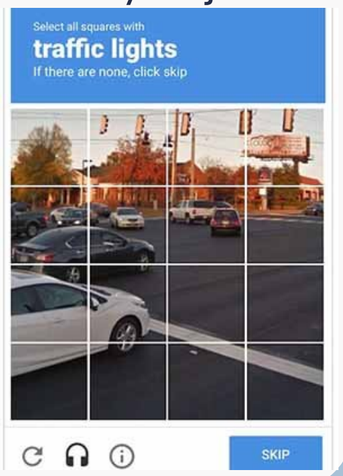
Tired of proving to every website that that you're not a bot?
We are here to help you out.

Active Captcha

Traditional CAPTCHA



Needs user to solve trivial puzzles or repeatedly identify objects.



We introduce a new technique that uses ML and statistical methods to eliminate the need of direct user interaction

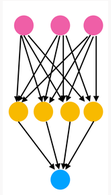
Our Proposal (Passive++ Solution)

Primary Check

Honeypot(s)

HTTP Methods

Neural Network



Probability of being a bot

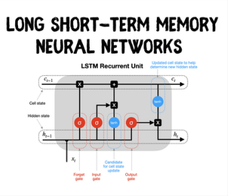
Discrete Time Markov Chains

Classify user based on a statistical score

Secondary Check

Mouse & Keyboard logs

LSTM model at backend



if (output score < threshold)

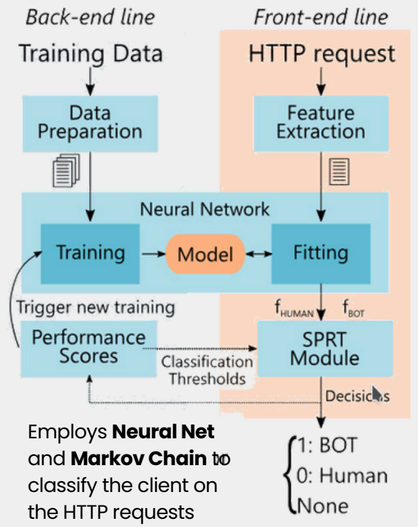
User prompted to resort to traditional CAPTCHA method.

Technical Approach

invisCAPTCHA

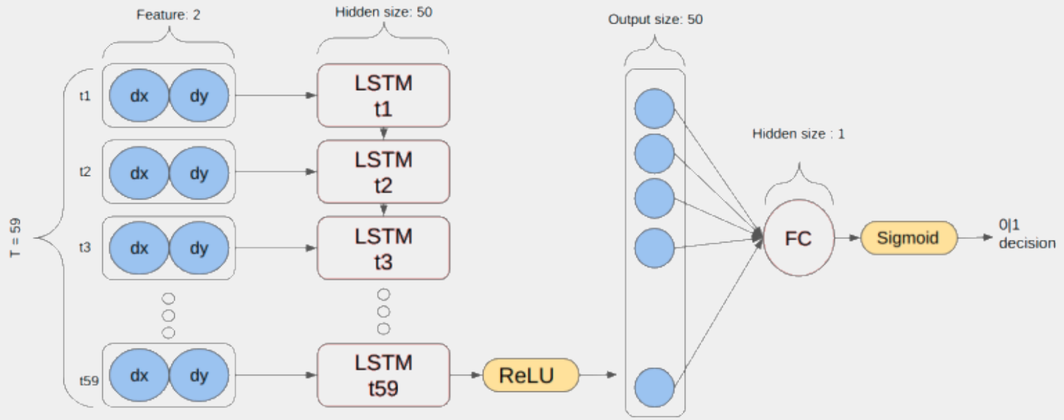
SMART INDIA HACKATHON 2024

Primary Level : HTTP Methods



Employs **Neural Net** and **Markov Chain** to classify the client on the HTTP requests made

Secondary Level (Behavioural Characteristics) : Mouse Movements



LSTM (Long Short Term Memory) Architecture:
For a given session, mouse movements are captured every **T** seconds and coordinate vector sequences of length **N** are created from this data. The coordinates are differentiated (discretely) to get the velocities in x and y directions which are then fed into a LSTM model.

Honeypot Traps (runs in parallel)

Layer 1: Surface Traps

- Hidden form fields
- Invisible links with anchor text in page metadata

Layer 2 : Behavioural Traps

- Humans are impatient
- Fake **AJAX** calls that simulate dynamic content updates

Layer 3 : Logical Traps

- **Decoy API endpoints** with realistic but non-functional responses

OR

Tech Stack

HTTP METHODS

HTML CSS JS

BACKEND

python

django

MACHINE LEARNING

TensorFlow

PyTorch

Feasibility & Viability

Every step of the idea is achievable. The model requires no human interaction to train. Apart from this, everything, ranging from the model to the Honeypots can be plugged into the JavaScript Framework, which can be rendered in the browser. The idea is also very cost effective. This all makes the approach feasible.

Several challenges

- A super intelligent bot which can evade honeytraps may be difficult to detect.
- The to-and-fro transfer of user logs may require additional bandwidth and higher transfer rate.
- Computational Overhead due to LSTM model may consume additional RAM.
- Potential False Negatives may prompt the user to reload or resort to traditional captcha.

Potential Solutions to challenges

- Dynamic HoneyTrap method, which plants HoneyTraps strategically to capture a bot.
- We can try to zip the data or encode it in a certain format to reduce data transfer rates.
- Light Gradiet Boosting Method , show promising results in preliminary testing, with comparable accuracy and a lesser computational overhead, and can be used as a alternative to LSTM.

Benefits and Impact

Benefits

- Enables seamless user verification without interrupting the natural flow of interaction with UIDAI portals
- Analyzes multiple behavioral factors to make more accurate bot/human distinctions than binary CAPTCHA systems
- Behavioral Data Driven analytics may provide valuable user interaction data to improve UIDAI's digital services
- Scalable protection which can safeguard multiple UIDAI portals and APIs with a centralized backend ML infrastructure

Impacts

- Ease of Access for Visually Challenged people.
- Over the time, user experience will get seamless and better.
- With increase in GenAI powered bots, traditional captcha methods are getting redundant. This provides us a safeguard.
- Our architecture will have a tangible impact on lives of common people when large scale integration of invisCaptcha is done across various government domains.
- This will increase accessibility of various digital servies provided by the government, especially catering to the underprivilidged.

Research & References

1. **Web Bot Detection Using Mouse Movement** Choi, J., Kim, S., & Yoon, Y. (2023), IEEE Transactions on Information Forensics and Security

2. **Efficient on-the-fly Web bot detection** Grażyna Suchacka, Alberto Cabri, Stefano Rovetta, Francesco Masulli. (2021)

3. **Bogazici mouse dynamics dataset** A. A. Kılıc¸, M. Yıldırım, and E. Anarım, Data in Brief, vol. 36, p. 107094, 2021

4. **https://github.com/vincentbavitz/bezmouse**

5. **https://github.com/hofstadter-io/self-driving-desktop**

6. **Honeypots Spitzner**, L. (2003). Honeypots: Tracking Hackers. *Addison-Wesley Professional*