International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 15(2023) pp. 438-459

Submitted: 25 Feb, 2023; Accepted: 22 May, 2023; Publish: 23 June, 2023

A Novel and Efficient Multilayered Approach to CAPTCHA: Design, Performance and Usability Evaluation

Navansh Goel[1]*, Tejaswi Kumar[2]* and C. Oswald[3]

[1]Vellore Institute of Technology, School of Computer Science and Engineerings,
Chennai, India, 600127
navansh.goel2019@vitstudent.ac.in

[2]Vellore Institute of Technology, School of Computer Science and Engineerings,
Chennai, India, 600127
tejaswi.kumar2019@vitstudent.ac.in

[3]National Institute of Technology Tiruchirappalli, Department of Computer Science and Engineering,
Tiruchirappalli, India, 620015
oswald@nitt.edu

* These authors contributed equally to this work.

Abstract: **With the wide usage of the World Wide Web getting more and more popular nowadays, the impact of the Internet on all user aspects of computer applications is huge. Almost all the web applications focus on the authentication and security preserving factors in order to overcome breach of data, spamming and to prevent bots from accessing them. To prevent the misuse of online servicing platforms and to provide a good level of reliable, secure and authenticated service, CAPTCHAs (Completely Automated Public Turing test to tell Computers and Human Apart) play a vital role. Usability aspect of CAPTCHA is defined as the ease of humans dealing with its challenges and overcoming them. CAPTCHAs should not be too easy for a human to identify while maintaining high difficulty for the bots to recognize the solution. Many of the existing text and image CAPTCHAs do not provide an acceptable level of usability for various groups of people. In this paper, we have designed a novel, usable, robust and reliable text CAPTCHA schemes framework named MACS-TCHA (Manipulation of Alphanumeric characters, Colors and Shapes - Turing test to tell Computers and Human Apart). A total of 4 primary variations of CAPTCHAs were designed and implemented by incorporating multilayered randomness approach, involving alphanumeric characters, shapes and colors for obtaining better usability. A large scale user study involving more than 200 distinct users was conducted in order to compare and evaluate the MACS-TCHA's accuracy and usability. A thorough investigation has been made on the reliability, robustness and performance of the proposed MACS-TCHA schemes with the existing methods in order to thwart recognition attacks from the bots. We also identified various usability factors for our evaluation and the results show that the users achieved high accuracy and consider our framework to be highly usable, reliable, fun and efficient.**

Keywords: CAPTCHA, evaluation, MACS-TCHA, robustness, security, usability, performance

I. Introduction

With the humongous amounts of data flowing over the Internet medium, there has also been an increase in the risks that web applications are facing. Nowadays, websites are being attacked by bots for different reasons, like stealing data, intrusion of malicious users, crashing the server, etc. The past few decades saw a variety of sophisticated mechanisms to deal with these problems. To deal with overcoming these issues and to fight these attacks and prevent the bots from accessing the web data, various security methods have been developed by researchers.

One of the prominent methods used is a Turing Test, designed by Alan Turing. A Turing Test is a test for determining whether the user is a human or a computer. A Turing test is a game like test in which a human judge with 2 other individuals, one being a human and one is a computer. The test proceeds with the judge asking both of them some questions based on which the judge decides which of them a computer is. For modern-day websites, a system called CAPTCHA was introduced [1]. CAPTCHAs are Completely Automated Public Turing tests to tell Computers and Humans Apart, meaning that instead of a human judge, there is a program that decides between humans and bots. The ba-

sic idea remains the same as the traditional Turing test. There is some sort of text or image displayed along with some distortion/noise. The user has to enter the text or follow a set of instructions and this user input becomes the deciding factor in CAPTCHAs.
CAPTCHAs have become a necessity today, as they continue to provide authentication for humans to enter a website. The CAPTCHA challenges the user by posing some questions that can easily be solved by humans. Such questions might become a hassle for automation bots, thus providing an upper hand for humans [2]. Further, by providing some sort of distortion, this authentication process can be strengthened. Because of certain characteristics that can only be recognized by humans, this type of authentication method has been successful. CAPTCHAs can be divided into two broad categories:

OCR-based authentication- The user is asked to type characters from distorted images. This method is mainly based on the weakness of the OCR software because it faces difficulty in reading text from distorted images, and produces inaccurate results.
Non-OCR-based authentication- This type of CAPTCHA authentication focuses on multimedia support to authenticate the user. Multimedia such as audio, video, pictures, etc. are used to make such CAPTCHAs, for example being the Text-to-Speech CAPTCHA [3], Video CAPTCHA [4], etc. Such types of CAPTCHAs are sometimes easier for many users but can require additional resources and time to crack.
The most common form of CAPTCHAs is text-based CAPTCHAs. The basic idea of a Text CAPTCHA is to obtain a text input from the user and analyze the input to conclude the user's identity as a human or bot. Text CAPTCHAs use a variety of techniques. There could be involvement of distortion of characters, noisy background, crossed-out text, and even some sort of instructions that the user has to follow to reach the answer. All these factors are based on the assumption that a human can recognize characters despite some sort of variations.
There have been more attempts to exploit such CAPTCHAs because of the fact that the text CAPTCHAs are the most prominent ones used in various web applications. Using appropriate machine learning techniques and image recognition, various types of exploitations have been done on the different types of text CAPTCHAs.
The bottleneck in-text CAPTCHAs come due to their simplicity. Because there is not enough variation in the text CAPTCHAs, the characters can be recognized easily and thus the answer can be retrieved. The distortion in the text can also create irritation and delay for the user to enter the valid answer. Another bottleneck is the confusion and delay caused because of the overlapping of the characters, presented in some text CAPTCHAs. This overlapping is introduced in order to defend the CAPTCHA from bots while enabling human users to complete the process. Although this seems to be an efficient method to avoid bots, this also comes with a disadvantage for human users. Such variations in text CAPTCHAs tend to give a hard time to many users because of the time taken by them to understand simple alphabets or characters.

To bridge the gaps in the literature, Tejaswi et al. have proposed novel text CAPTCHA schemes named MACS-TCHA based on colors and shapes along with their usability evaluation [5]. This work deals with proposing four more novel variations of MACS-TCHA which fulfill both security and usability factors required for a robust and an efficient text CAPTCHA, as an extension to the previous work [5]. A total of 4 primary variations of CAPTCHAs were designed and implemented as a usability system. The key contributions in our work, compared to [5], are summarized below:

- Design of novel, efficient and robust text CAPTCHA's by increasing the number of layers using multiple levels of randomness that will make it difficult for the bots to predict the correct answer while maintaining an acceptable level of ease of use for humans.
- Performing extensive usability evaluation investigations to analyze the accuracy, ease of use and other usability factors for MACS-TCHA, through data collected from a large user survey.
- Design of variations of MACS-TCHA intended for any age group including children and mentally challenged people using elementary level knowledge of basic colors and shapes, instead of irritating distortion in CAPTCHA.
- A detailed and thorough theoretical analysis to show the robustness of our proposed MACS-TCHA schemes in order to analyze the effect of recognition attacks from bots using inferences from the existing works on primitive deep learning models.

The different variations of MACS-TCHA have several applications for practical security domains. Web Registration, Online Polling and Mitigating Comment Spam are the most common examples of the practical usage of MACS-TCHA. Since flooding attacks and bot attacks can create a nuisance for web applications, MACS-TCHA can effectively prohibit bots from entering the homepage. Since MACS-TCHA asks the user to perform a specific task, Dictionary attacks and Phishing attacks can also be prevented. Even in the field of online gaming, usage of bots is a massive threat, which can be prevented by using MACS-TCHA to distinguish between a human user and a bot. Since these applications fall under the domain of security and privacy of applications connected over the internet, MACS-TCHA serves a solid web security and privacy protection application.
This paper is organized as follows: Section 2 discusses the literature review of text and image CAPTCHAs. Our proposed MACS-TCHA schemes are explained in Section 3. In Section 4, we present the usability evaluation studies and discussions of our proposed schemes. Reliability Analysis and Robustness of proposed MACS-TCHA Schemes are presented in Section 5. Section 6 presents the Human Perception Analysis from the Usability evaluation of MACS-TCHA Schemes. Section 7 concludes with future directions.

II. Literature Review

In this section, various types of CAPTCHAs existing in the literature have been elaborated with their design along with

a short analysis of the same.

A. Gimpy CAPTCHA

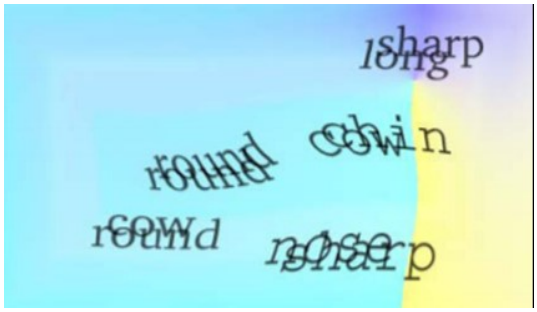


Figure. 1: GIMPY CAPTCHA

The methodology behind this CAPTCHA is to select a few words and present them as distorted and corrupted images by adding colors in the background and making non-linear modifications and asking the user to type the words correctly [5]. The words generally consist of simple words from the English dictionary as seen in Figure 1. Generally, the user is asked to type in any three words, so that it becomes difficult for a machine to solve the CAPTCHA. By adding noise, rotation, or scattering, Walid Hassan [7] showed that these types of CAPTCHAs are not easy to break as they create variability in detecting. In 2003, Mori and Malik [8] showed a way to break Gimpy using shape context. This method from then on is used for various exploitation techniques for CAPTCHA recognition and shape matching. There are other variations of this Gimpy CAPTCHA like EZ-Gimpy and Gimpy-r[34].

B. Baffle Text

Figure. 2: Baffle Text CAPTCHA

In this type of CAPTCHA, an image is filled with random alphabets in a sequence, to form a random word, which may or may not have any meaning, and then the image is processed in different ways, to introduce noise into the image. As seen in Figure 2, this will distort the image visibility and increase the difficulty for the user in reading the letters [9]. A great merit of this type of CAPTCHA is that it is safe from dictionary attacks since the words are not standard English words. A major demerit of this design is the distortion of images displayed to the user, making it difficult to solve.

C. Animated CAPTCHA

The Animated CAPTCHAs came into existence due to the limitations in static CAPTCHAs. Such CAPTCHAs involve some sort of manipulation with frames and movement of characters. This is an approach widely presented by many researchers like Cui et al. [10][11]. The basic idea is to present some sort of movement to the primary subject of the CAPTCHA over and above some sort of noise in the background.

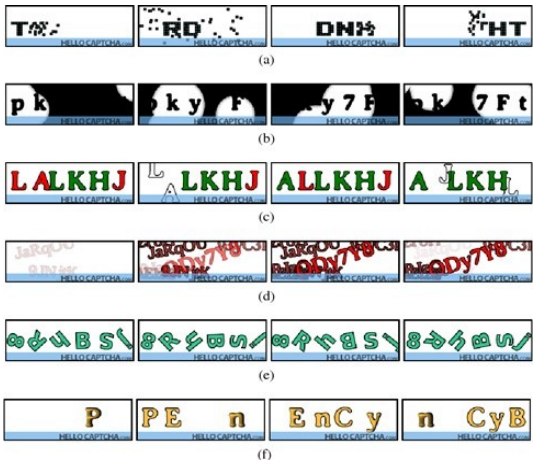


Figure. 3: Animated CAPTCHA

As stated by Yang-Wai Chow and Willy Susilo [12], this type of CAPTCHA takes the advantage of the ability in humans to recognize depth through motion, which is still unknown to a computer. Further, as mentioned in Figure 3, the change in frames makes it difficult for a machine learning bot to recognize the correct answer. Willy Susilo [13] further mentions the methods of breaking an animated CAPTCHA - HelloCAPTCHA wherein parameters like time delay in showing frames in animated CAPTCHAs can become a major reason for the loss of security and result in the breakdown of such CAPTCHAs using effective exploitation methods.

D. Question-based CAPTCHA

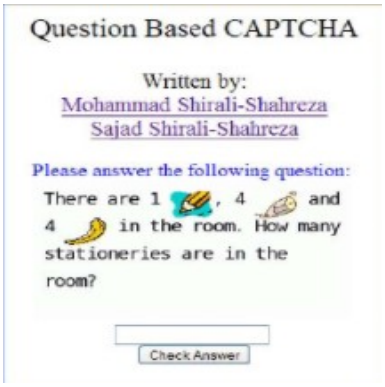


Figure. 4: Question-Based CAPTCHA

The Question-based CAPTCHA was designed to get a fast response from the user by giving them a simple situation containing images and numbers related to the images with a question at the end. The questions can be answered by any human easily while a bot would face problems in understanding the question and predicting the answer would be very difficult. For example, "There are 5 pencils, 7 oranges, and 2 sharpeners on a table.
"How many stationeries are there on the table in total?"[14], so here the answer is 7 because pencils and sharpeners come

in the category of stationeries (5+2=7) while oranges come in the category of fruits, which is very hard for a bot to detect and categorize, and then predicting the answer would be tough and the bot could be misled by the use of the images to predict a wrong answer as shown in Figure 4. The merit of this CAPTCHA is that it is very time-efficient for the user as the user just has to figure out and type a number and the demerit being that if the user does not have the knowledge of the categories then it would be difficult for the user to solve it.

E. No CAPTCHA ReCAPTCHA

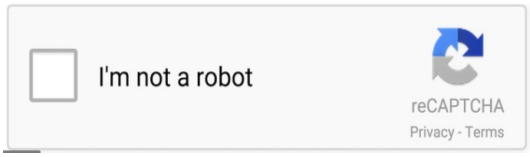


Figure. 5: No CAPTCHA ReCAPTCHA

Google ReCAPTCHA is one of the most used CAPTCHAs in the whole of the community. Basically, this CAPTCHA asks whether the user is a bot or not, as shown in Figure 5, and gathers some information from the user's behavior of his previous, current, and future actions while clicking the checkbox and analyses it to check whether the user is a bot or not [15]. The analysis can also include the user's browsing history as well as the movement of the user's mouse on the page. If still the CAPTCHA is unsure of the user then it could present the user with a visual ReCAPTCHA to ensure the security of the system and the user have to follow its instruction to pass it. The No CAPTCHA was concluded as the least frustrating test among the many available CAPTCHAs and the readiness for future use for this CAPTCHA is also great as per the survey conducted by Gafni et al.[16]
The ReCAPTCHA system considers a wrong image coupled with correct ones as a correct answer and this flexibility in terms of answer verification might be a weak point that can be used to break this CAPTCHA.[17]

F. Simple text-based CAPTCHA

Figure. 6: Simple text-based CAPTCHA

This CAPTCHA focuses on preventing breaking attacks by presenting many types of obstructions. As shown in Figure 6, obstructions like font variations, random length of CAPTCHA, disallowing replays of previously submitted CAPTCHAs, rotating the strings/code at various different angles, etc. [18] This kind of CAPTCHA prevents mainly image recognition but is different for users to crack in one go and some characters are difficult to understand for the user which sometimes demands to be refreshed.

G. 3-D CAPTCHA

Figure. 7: 3-D CAPTCHA

In this type of CAPTCHA, the assumption made by the creators is that humans can recognize 3D characters better than the OCR (Optical Character Recognition) machine learning bots [19].
The CAPTCHA requires the user to enter the characters from left to the right sequentially. These characters are in 3D form instead of 2D format. This creates a restriction for the automated bots to figure out the correct answer. Furthermore, they have given various other schemes that are variations of the CAPTCHA given in Figure 7. By considering noise and overlapping their CAPTCHA can increase the difficulty. Users with eye-related issues would find such CAPTCHAs very difficult to solve. Even for someone having healthy eyesight, reaching the correct answer for this type of CAPTCHA can be tiresome.

H. Clickable CAPTCHA

In this type of CAPTCHA, the user must click on the three English words which are mixed amongst the other sequences of letters. If the user clicks on any other words, the solution is termed as invalid. This type of CAPTCHA can have many variations, such as words in some other language, or different grid sizes, or a different number of correct solutions, and so on, as shown in Figure 8. [20] But this type of CAPTCHA has many drawbacks. One of them is that the user has to be proficient in English to solve this. Also, this CAPTCHA can be solved by bots using OCR. And also, the user cannot solve this CAPTCHA on a phone or other small devices, since the letters are disoriented and very hard to see properly.
New types of CAPTCHAs have emerged over the years, like the De-CAPTCHA [42], where the authors have used

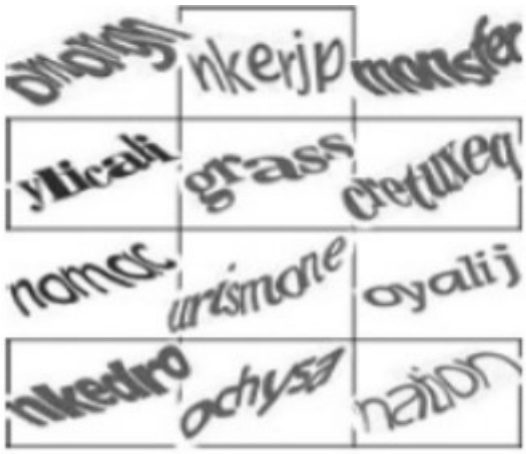


Figure. 8: Clickable CAPTCHA

character extraction from CAPTCHAs, which is done using a Depth First Search technique, while character recognition is done using a Convolutional Neural Network. Alejandro et al. [43] propose a novel method BeCAPTCHA, which works by analysing the drag and drop tasks in addition to accelerometer data from touchscreen input to recognize whether the sample is from a bot or a human. The authors use HuMIdb along with Generative Adversarial Neural Networks and handcrafted methods to generate fake samples and test their novel framework. UNI-CAPTCHA [44] is a novel robust and dynamic user-non-interaction CAPTCHA Model, which is based on hybrid biLSTM and Softmax. The author Ahmet Ali Süzen, has conducted a robust study on the behavior-based CAPTCHA, that is, UNI-CAPTCHA, which was also developed in this paper. It was developed to detect user-bot without interaction with user. The UNI-CAPTCHA engine outperforms standard bot detection and CAPTCHA solutions in terms of stability, speed, and detection. Asish et al. [45] present a two stage verification scheme using HandCAPTCHA and presentation attack detection (PAD) by taking the real hand images of legitimate users who have successfully passed a random HandCAPTCHA challenge. In comparison to HandCAPTCHA [46], which was presented by the authors earlier, the new two stage scheme overcomes vulnerability attacks and provides enhanced security. There are many other types of CAPTCHAs such as GeoCAPTCHA [30], CaRP (Captcha as gRaphical Passwords) [31], Secure Text-Based CAPTCHA [32], Microsoft's Two-Layer CAPTCHA [33], Smart CAPTCHA [35], e-banking CAPTCHAs [36], Hollow CAPTCHA [37], CAPTCHsStar [38], The Yahoo! CAPTCHA [39], Jigsaw Puzzle CAPTCHA [40] and CLAPTCHA [41].

All the above-mentioned CAPTCHAs offer some sort of distortion of characters to the users. This could include overlapping of characters, adding depth to characters, adding lines and noise in the background, tilting the characters at some angle, etc. All these are sufficient to avoid bots from breaking the CAPTCHAs, although these come with a cost at the human end. For any normal user to enter the valid answer to a CAPTCHA, a certain amount of time is required which is increased in the above cases due to the visual complexity. Because a normal user has to strain their eyes in order to complete the CAPTCHA, a certain amount of irritation and confusion is created every time the user encounters one of the above-mentioned CAPTCHA.

Along with advances in novel frameworks of CAPTCHAs over the years, significant work has been done for breaking existing CAPTCHAs. Ying Ma et al. [47] propose a novel CAPTCHA recognition algorithm called neural CAPTCHA networks (NCN). By the use of Convolution Neural Networks and bidirectional recurrent modules, NCN extracts the features from text-based CAPTCHAs and learns spatially sequential information from them. According to the authors, NCNs can efficiently solve puzzle based CAPTCHAs, character recognition and character matching CAPTCHAs along with arithmetic operation related CAPTCHAs. There are a variety of complex CAPTCHA these days, but Yao Wang et al. [48] have proposed a fast text captcha solver based on a small number of samples. They have built a generative adversarial networks to simplify the captcha images before character segmentation and recognition. By using this model, they have achieved a very high success rate of over 96% character accuracy. Similar work had been done by Chunhui Li [49], where they have proposed an end-to-end attack strategy to break text-based CAPTCHAs based on cycle-consistent generative adversarial network. The have discussed in detail about how to train and use the model and also the pros and cons of the proposed model. This is indeed the next step to break simple text based CAPTCHAs and can be used for further analysis on more such complex CAPTCHAs.

MACS-TCHA intends to solve the above mentioned issues by introducing variations in a text CAPTCHA in the form of multiple layers instead of placing distortion or noise in the image [5]. The random nature of generating the alphanumeric characters, colours and shapes helps in an effective design of the proposed CAPTCHA schemes. MACS-TCHA will reduce the irritation felt by the user and also give the user a simple puzzle-like task instead of asking the user to guess distorted characters. The best characteristic of MACS-TCHA is that it uses basic colors and shapes which can be identified by any group of people. The instructions are given in easily understandable English allowing the user to solve it quickly. Thus, this CAPTCHA does not create a high degree of confusion to a human and is very tough for a bot to predict the correct answer due to variations of layers used.The next section presents the detailed design and system of the proposed MACS-TCHA Schemes.

III. Design of our proposed MACS-TCHA Algorithms

The idea of MACS-TCHA was to create a CAPTCHA that involves concepts that can be solved by any age group of people. By using concepts like Shapes, Colors, Alphabets, Numbers, and basic mathematical operation, MACS-TCHA makes sure that the bots will find it very tough to break the system and at the same time humans can reasonably feel comfortable to solve the CAPTCHA. This enables not only the professionals but also elementary level school-going children to solve the CAPTCHA. With the increasing use of the World Wide Web and its associated technologies, many schools have already incorporated the latest technologies in their syllabi. With MACS-TCHA, we strongly believe that

it will be easy for many student communities and even people who have not received complete education to completely solve a CAPTCHA. The only knowledge the user needs to know to solve our CAPTCHA is at an elementary school level.

The main aim was to create a distinction between human users and computer bots by the means of multiple layers of design. In every layer of the design, a certain degree of randomness is used to generate the parameters (includes characters, numerals, colours and shapes) involved. This would prevent the CAPTCHA from being susceptible to various exploitation attacks. This was done to keep in mind the ease of solving the CAPTCHA by human users while maintaining enough entropy at every layer. With this thought in mind, two major classifications of MACS-CAPTCHA have been designed and furthermore superior versions of the two MACS-TCHAs are also proposed and evaluated. For each CAPTCHA, we display 5 character/digit images, 5 shapes, 1 instruction statement for the user along with an input block and a submit option. On clicking the submit button, a message is displayed on the screen validating the user's input.

The workflow of the design of our proposed novel MACS-TCHA is shown in Figure 9. It starts by mapping the digits in the number_array to their corresponding images using the image_hashmap. With this, the corresponding shape(which could be a colored circle or a black circle/triangle/square/rectangle) is placed. The shapes are then checked by referring to the rgb_array or shapes_array and placed in it accordingly. Once this step is implemented for a single character, the same is done for all the remaining characters using a loop. Further, the chosen instruction statement is displayed, followed by the input block, and the submit button below the displayed CAPTCHA.

A. Numeric_Color based MACS-TCHA

In this scheme, a sequence of random numbers inside colored circles is displayed to the user. The user has to enter the sequence of numbers from left to right as asked in the instruction statement shown. The algorithm starts by creating a static array of randomly generated numbers. Every time the user fails to enter the correct answer or reloads the CAPTCHA, this array is reinitialized with a new set of numbers generated randomly. This serves as the first layer of variation in our algorithm. Further, another static array is created to store randomly generated colors which serves as the second layer of variation. This is achieved by creating another array that contains the number of choices for the colors provided. For every choice, the first letter of the color is stored inside this static array. From this newly created color array, a random index is selected, based on which the instruction statement is devised.

Since the colors are randomly selected, the instruction statement would also be different each time. With the help of the randomly chosen index and randomized colors in the color array, the instruction statement from a previously stored string array can be chosen. This variation in the instruction statement serves as the third layer of variation in our algorithm. Because the integer array and the color array are of equal size, individual digits to the colors can be easily mapped using the index. Further, every index with the color

Algorithm 1 Numeric_Color based MACS-TCHA Algorithm

Ensure: $n = 5$
function GENERATOR
  for $i = 0$ to $n - 1$ **do**
    number_array[i] = random_number(0 to 10)
  end for
  for $i = 0$ to 9 **do**
    image_hashmap.insert (i, "Digit.jpg")
  end for
  rbg_hashmap.insert("first_letter_of_color",color)
  for $i = 0$ to $n - 1$ **do**
    if $pos == 1$ **then**
      color_array[i] = "r"
    else if $pos == 2$ **then**
      color_array[i] = "g"
    else
      color_array[i] = "b"
    end if
  end for
  {Declare variable condition by choosing a random index from 0 to n-1}
  {Declare variable find value = shape array[condition]}
  if find_value=="first_letter_of_color" **then**
    instruction_statement = options[corresponding_index_of_first_letter]
  end if
  for $i = 0$ to $n - 1$ **do**
    if color_array[i] == find_value **then**
      result_vector.append(number_array[i])
    end if
  end for
  for $i = 0$ to result_vector.size() **do**
    answer += integer_to_string(result_vector[i])
  end for
end function

function USER_INPUT(answer)
  read user_input
  if user_input == answer **then**
    message = "VERIFIED"
  else
    message = "TRY AGAIN"
    answer = ""
    result_vector.clear()
    GENERATOR()
  end if
end function

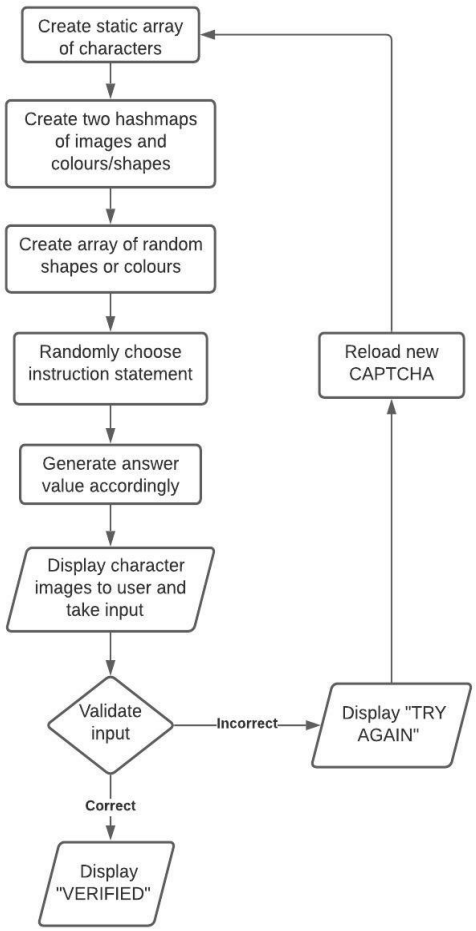


Figure. 9: Flowchart depicting workflow of MACS-TCHA

in the chosen instruction statement is compared and used to reach a valid answer. Since this answer is not of a fixed length, a dynamic array is used, like a vector in our case to store this result. The flexibility in the size of the answer serves as the fourth layer of variation.

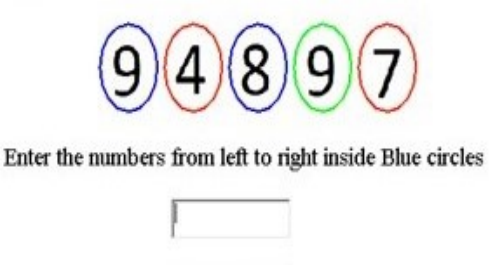


Figure. 10: Numeric_Color based MACS-TCHA

Furthermore, the CAPTCHA is displayed to the user by merging pictures of numbers that were retrieved with the help of the Hashmap, which stores the number as the key and the image address of each of the numbers as the value. This is then retrieved using the randomly generated number array bypassing each of its numbers as a key for the hashmap and getting the address of the image which could be parsed and displayed by the selected language in a sequential way so that it looks like a continuous CAPTCHA. Then another hashmap is used to store the first letter of the color as its key and the complete color name as its value. The color could then be retrieved by the language for making the colored circle around each displayed number according to the color array. Effectively, the user sees a set of numbers encircled within different colors as given in Figure 10. Along with this, instruction is displayed to the user which they can use to reach a suitable answer. The corresponding algorithm for this scheme is given in Algorithm 1.

Two different variations for Numeric_Color based MACS-TCHA are listed below in sub sections 3.1.1 and 3.1.2.

1) Sum_Color based MACS-TCHA

In the first variation, instead of asking the user to enter the numbers sequentially from left to right, the user is required to enter the sum of numbers according to the instruction statement as displayed in Figure 11. Instead of traversing through the vector that stores the answer characters and appending them to the string variable answer in Algorithm 1, the characters can be directly added to an integer type variable answer. Further, this variable can be type-casted into a string in order to match the user's input.

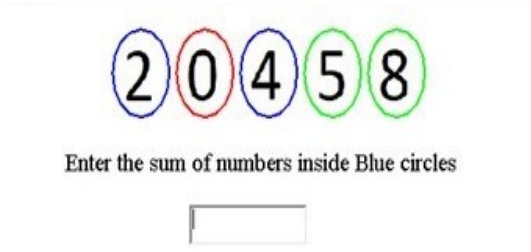


Figure. 11: Sum_Color based MACS-TCHA

2) Alpha_Color based MACS-TCHA

In the second variation, as shown in Figure 12, the total number of characters inside the hashmap is increased by involving small case alphabets. This decreases the probability of retrieving a character from 0.1(1/10) to 0.02778 (1/36) and makes it harder for bots to predict the result for the CAPTCHA.

After creating the static array of the random numbers,

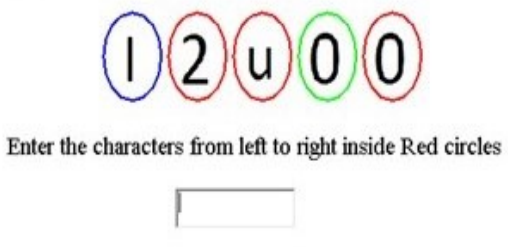


Figure. 12: Alpha_Color based MACS-TCHA

Algorithm 2 is appended to Algorithm 1. The primary

change is that the dataset size is increased by including alphabets so that the probability of predicting the answer is reduced, as discussed above.

Algorithm 2 Alpha_Color based MACS-TCHA

```
alphabets = "abcdefghijklmnopqrstuvwxyz"
for i = 0 to 36 do
    if i < 10 then
        digits_hashmap.insert(i, Integer_to_String(i))
    else
        digits_hashmap.insert(i, alphabets.substring(i - 10, i - 9))
    end if
end for
for i = 0 to 36 do
    image_hashmap.insert( digits_hashmap[i] , digits_hashmap[i] + ".jpg" )
end for
```

B. Numeric_Shape based MACS-TCHA

In this scheme, a sequence of random numbers inside shapes is displayed. The user has to enter the sequence of numbers from left to right as asked in the instruction statement shown. Just as explained in Section 3.1, we start by generating a static array of randomly generated numbers. Every time a user fails to provide the correct solution to the CAPTCHA, this array is reinitialized with a new set of numbers generated randomly which acts as the first layer of variation in our algorithm. Similarly, a static array is created, consisting of random numbers which will serve as the first variation layer. This is achieved by creating another array that contains the number of choices for the shapes provided. For every choice, the first letter of the shape is taken and stored inside this static array. From this newly created shape array, a randomly selected index decides the instruction statement and our result. The instruction statement is then randomly chosen from a previously stored string array which is the third layer of variation in our algorithm. Since the number array and the shape array are of equal size, individual digits to the colors can be mapped using the index. Similar to what is mentioned in section 3.1, every index with the shape in the chosen instruction statement is compared and the valid answer for the user to enter, is calculated. Since this answer is not of a fixed length, a dynamic array like a vector in our case is used to store this result. This flexibility in the size of the required answer serves as the next layer of variation. Effectively, a sequence of numbers is displayed, where each digit is inside a shape which could be a square, triangle, or circle. As seen in Figure 13, an instruction statement for the user is displayed along with this CAPTCHA. The detailed algorithm is shown below in Algorithm 3.

Algorithm 3 Numeric_Shape based MACS-TCHA Algorithm

```
Ensure: n = 5
  function GENERATOR
    for i = 0 to n − 1 do
        number_array[i] = random_number(0 to 10)
    end for
    for i = 0 to 9 do
        image_hashmap.insert (i, "Digit.jpg")
    end for
    for i = 0 to n − 1 do
        if pos == 1 then
            shapes_array[i] = "c"
        else if pos == 2 then
            shapes_array[i] = "t"
        else
            shapes_array[i] = "s"
        end if
    end for
    {Declare variable condition by choosing a random index from 0 to n-1}
    {Declare variable find value = shape array[condition]}
    if find_value=="first_letter_of_color" then
        instruction_statement = options[corresponding_index_of_first_letter]
    end if
    for i = 0 to n − 1 do
        if shapes_array[i] == find_value then
            result_vector.append(number_array[i])
        end if
    end for
    for i = 0 to result_vector.size() do
        answer += integer_to_string(result_vector[i])
    end for
  end function

  function USER_INPUT(answer)
    read user_input
    if user_input == answer then
        message = "VERIFIED"
    else
        message = "TRY AGAIN"
        answer = ""
        result_vector.clear()
        GENERATOR()
    end if
  end function
```

1) Sum_Shape based MACS-TCHA

In this variation, instead of asking the user to enter the numbers inside a certain colored shape sequentially from

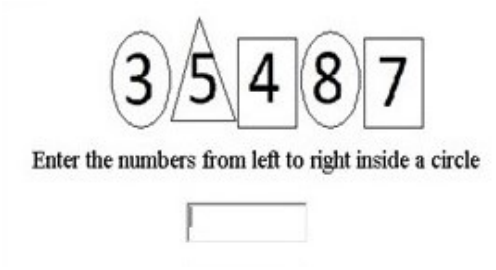


Figure. 13: Numeric_Shape based MACS-TCHA

left to right, the user is required to enter the sum of numbers according to the instruction statement as given in Figure 14. Instead of traversing through the vector that stores the answer characters and appending them to the string variable answer in Algorithm 2, the characters can be directly added to an integer type variable answer. Further, this variable can be type-casted into a string in order to match the user's input.

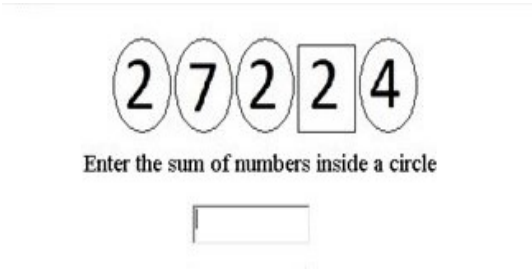


Figure. 14: Sum_Shape based MACS-TCHA

2) Alpha_Shape based MACS-TCHA

In the second variation, as shown in figure 15, the total number of characters inside the hashmap is increased by involving small case alphabets. This decreases the probability of retrieving a character from 0.1(1/10) to 0.02778 (1/36) and makes it harder for bots to predict the result for the CAPTCHA.

After creating the static array of the random numbers, Al-

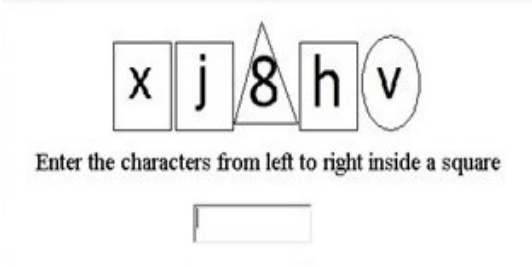


Figure. 15: Alpha_Shape based MACS-TCHA

gorithm 2 is appended to Algorithm 3. The change is that the dataset is magnified by including alphabets so that the probability of predicting the answer is reduced as discussed above.

C. Time and Space Complexity Analysis of MACS-TCHA

Time Complexity: The proposed algorithms for all the versions of Alpha_Color and Alpha_Shape based MACS-TCHA work in $O(n)$ and $\Omega(n)$ where n denotes the size of the Hashmap used for mapping the alphanumeric characters in generating the MACS-TCHA. Linear time complexity is achieved since the algorithm iterates over the Hashmap only once. Numeric_Shape, Numeric_Color, Sum_Shape and Sum_Color based MACS_TCHA schemes have a smaller Hashmap size since they contain only numeric characters.

Space Complexity: The proposed algorithms for all the versions of Alpha_Color and Alpha_Shape based MACS-TCHA have a space complexity of $O(n)$ and $\Omega(n)$ since we store a Hashmap of size n and a few linear arrays. Numeric_Shape, Numeric_Color, Sum_Shape and Sum_Color based MACS_TCHA schemes have a smaller Hashmap size since they contain only numeric characters.

IV. Usability Evaluation Analysis and Discussions

Overall simulation is performed on an Intel i7 processor 9th generation with 16GB Main Memory and 1TB Solid State Drive on Windows 10 Platform. Apache NetBeans, an Integrated Development Environment (IDE) is used for developing Java applications. It is much easier to visualize and run GUI based programs in Java comparison to C or C++. The libraries used for making the working prototypes are AWT(Abstract Window Toolkit), Applet, Utility. The platform used for developing these CAPTCHAs is JAVA Applet. Java Applet provides a good option for developers to play with dynamic content. Also, it can run inside the browser and works at the client-side. It executes very quickly as the process of caching is employed by computers thus shortening the program's runtime as well as it can work on various platforms, thus making it platform-independent.

An exhaustive usability evaluation survey was conducted for a total of four CAPTCHAs, for which we received responses from 204 users from various age and professional groups. Out of these, 135 being male and 69 being female and a good variety of users were surveyed belonging to different professional groups. Precisely 126 university students, 38 school students, and 40 working professionals participated in the survey. These consist of people from different age groups, of which 121 belonged to the age group 15-20, 37 people were above 30, 30 people were between 20 and 25, 6 people between 15-20 and 10 below 15. The responses were categorized based on their gender, age, and profession which allowed us to analyze the data accurately.

All the users were requested to rate our MACS-TCHAs based on the following parameters - Ease of Use, Clarity of Instruction Set, Clarity in understanding Colors and Shapes, and finally Clarity of Characters inside the Shapes. The respondents had to choose between easy, moderate, and difficult for all the above parameters for the CAPTCHAs shown to them. The following factors form the basis for the usability rank we created.

Factors used:

- Ease of Use - This factor is used to determine the basic complexity for solving the CAPTCHA. The respondents were asked to provide their feedback on the difficulty they faced in reaching the final answer for every CAPTCHA.
- Clarity of Instruction Set - This is used to determine the understanding level of the instruction statement given to the user. The respondents were asked to give their feedback on the difficulty to understand and comprehend the instruction statement.
- Clarity in understanding Colors and Shapes - This is to map the confusion based on shapes and colors. By gathering the responses based on this parameter, we can understand the difficulty faced by the user while recognizing the shapes and colors.
- Clarity of characters inside the shapes - This factor is used to find out whether the given characters inside the shapes are clear to the respondents while understanding and filling the response.

Further, the respondents were asked to solve the CAPTCHA. This helped us in understanding the overall difficulty as well as the success rate of each CAPTCHA, which is represented in the form of a Hit Ratio as given below.

$$HitRatio = \frac{Totalnumberofcorrectresponses}{Totalnumberofresponses}$$

Apart from this, we put forth a question to the respondent, asking their opinion on the need to refresh the CAPTCHA. This gave us a very promising response where 96.1% of the people didn't feel the need to refresh the Sum_Color based MACS-TCHA. 94.6% of people for Alpha_Color based MACS-TCHA and Sum_Shape based MACS-TCHA and 92.2% of the people for Alpha_Shape based MACS-TCHA vouched for not refreshing. This proved to us the usability of our CAPTCHAs in real-world applications and because of the simplicity yet a small challenge, resembling a puzzle, our CAPTCHA showed meaningful and interesting results. For clarity sake, all the data represented in the graphs and tables for the 4 proposed CAPTCHA schemes are represented as:
For clarity sake, all the data represented in the graphs and tables for the 4 CAPTCHA schemes is represented as:

- Sum_Color MACS-TCHA (Scheme 1)
- Alpha_Color MACS-TCHA (Scheme 2)
- Sum_Shape MACS-TCHA (Scheme 3)
- Alpha_Shape MACS-TCHA (Scheme 4)

A. *Analysis of MACS-TCHAs based on Hit Ratio*

It is observed from Figure 16 that the Sum_Shape based MACS-TCHA is the most accurately attempted while the Alpha_Color based MACS-TCHA is the least accurate among them. In fact, both Alpha_Color based MACS-TCHA and Alpha_Shape based MACS-TCHA are almost equal in terms of Hit ratio.

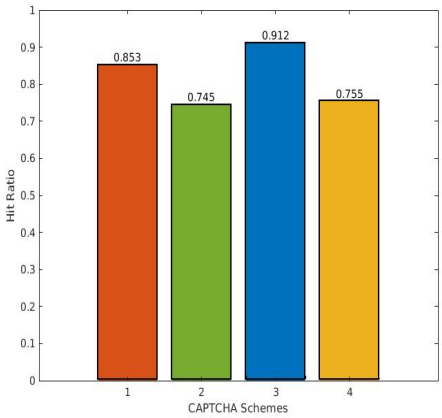


Figure. 16: Hit Ratio vs CAPTCHA Schemes

Since the Sum_Color based MACS-TCHA and Sum_Shape based MACS-TCHA have only numeric characters, they have a more accurate response from the users as seen in Table 1. Due to the similarity in few alphabets, users can attempt the CAPTCHA incorrectly, although the chance for this is very low considering the fact that the English language has distinct characters. The example provided in the survey had confusing characters in order to get the worst case hit ratio. In the case of the original program, these numbers will increase since every user will get a different sequence of characters.

B. *Analysis of Genders based on each factors*

Here, we have performed a detailed study of the usability evaluation of MACS-TCHA by various genders for every factor considered which are discussed in subsections.

1) *Effect of Gender on Hit Ratio*

Figure 17 depicts the Hit Ratio based on the gender of the user, for the four CAPTCHA schemes. For the Sum_Color based MACS-TCHA, a higher percentage of female users solved the CAPTCHA correctly than the male users, whereas a higher percentage of male users solved the fourth CAPTCHA correctly. The accuracy for both the genders was almost the same in the case of the Alpha_Color based MACS-TCHA and Sum_Shape based MACS-TCHA. It is scientifically proven that females have a better understanding and perception when it comes to colors [21]. Since Sum_Color based MACS-TCHA includes colors, more number of females have correctly attempted the CAPTCHA which can be seen from Table 2.

2) *Effect of Gender on Ease of Use*

Figure 18, as shown below, depicts the percentage of people who found the CAPTCHAs either easy, moderate, or difficult, based on their genders. Although almost all the four graphs show that the users found the CAPTCHAs easy, females have a higher percentage for the moderate level in terms of ease of use. In Sum_Color based MACS-TCHA and Sum_Shape based MACS-TCHA, users were meant to calculate the sum of numbers according to the instruction

Table 1: Hit Ratio for all CAPTCHA Schemes

	CAPTCHA Schemes			
	Scheme 1	**Scheme 2**	**Scheme 3**	**Scheme 4**
Hit Ratio	0.853	0.745	0.912	0.755

Table 2: Factors vs Gender for all CAPTCHA Schemes

		CAPTCHA Schemes							
		Male				**Female**			
		CAPTCHA Scheme				**CAPTCHA Scheme**			
		1	**2**	**3**	**4**	**1**	**2**	**3**	**4**
Ease of Use(%)	**easy**	94.82	92.59	96.30	91.10	89.86	86.96	91.30	85.51
	medium	3.70	6.67	2.96	8.15	10.14	11.59	7.25	11.59
	hard	1.48	0.74	0.74	0.74	0.00	1.45	1.45	2.90
Clarity of Instruction Set(%)	**easy**	90.37	88.89	94.07	90.37	86.96	86.96	92.75	82.61
	medium	8.15	10.37	4.45	7.41	13.04	11.59	7.25	15.94
	hard	1.48	0.74	1.48	2.22	0.00	1.45	0.00	1.45
Clarity in understanding Colors(%) and Shapes	**easy**	88.89	89.63	95.56	89.63	86.96	85.51	91.3	88.4
	medium	8.89	6.67	2.96	8.89	13.04	13.04	7.25	7.25
	hard	2.22	3.70	1.48	1.48	0.00	1.45	1.45	4.35
Clarity of characters(%)	**easy**	95.56	93.34	97.04	91.85	85.51	85.51	89.86	82.61
	medium	3.70	3.70	1.48	5.93	11.59	13.04	8.69	14.49
	hard	0.74	2.96	1.48	2.22	2.90	1.45	1.45	2.90
Hit Ratio		**0.82**	**0.79**	**0.91**	**0.80**	**0.91**	**0.78**	**0.91**	**0.70**

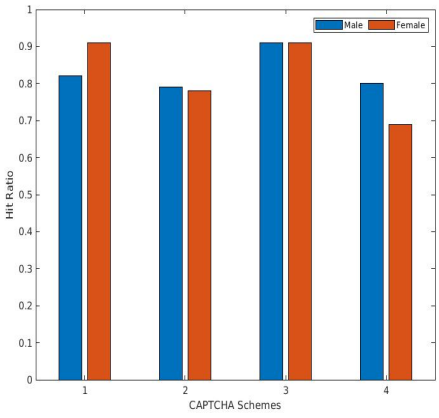


Figure. 17: Hit ratio vs CAPTCHA Schemes based on Gender

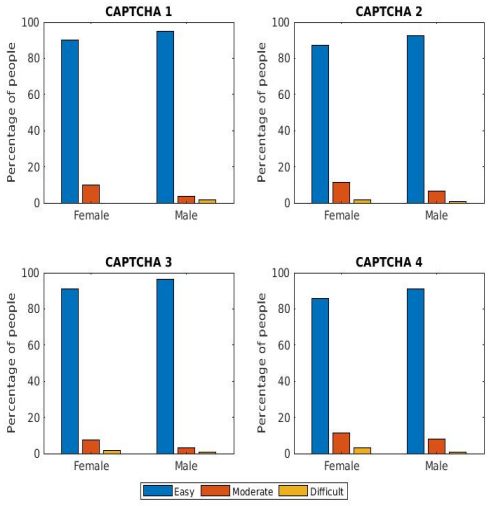


Figure. 18: Ease of Use vs Gender for all CAPTCHA Schemes

statement, which was easy for them to understand. But in Alpha_Color based MACS-TCHA and Alpha_Shape based MACS-TCHA, the moderate and difficult percentages increased as compared to the other two CAPTCHAs. This is because of the example that was given to the user which contained all alphabets as the answer and users were confused and tried making a meaningful word out of the letters. This led them to answer incorrectly and find the CAPTCHAs a bit more difficult than the others.

3) Effect of Gender on Clarity in understanding Instruction Statement

The percentage of people who found the instruction statements of the CAPTCHAs either as easy, moderate, or difficult, based on their gender is depicted in Figure 19. Although all the four graphs show that users found the CAPTCHAs easy, a higher percentage of females found the instruction statements moderate than the males. In this case, we found that the percentage of users who found the instructions difficult is very less. The instruction statement in all 4 CAPTCHAs was relatively easier in comparison to the existing CAPTCHAs. Because of this, the number of people opting for the easy option is very high irrespective of gender. Although Alpha_Shape based MACS-TCHA had both alphanumeric characters and Shapes, the instruction set played an important role in this case. This is again a reason for a slightly higher percentage of people opting for moderate level, which can be clearly seen from the studies in Table

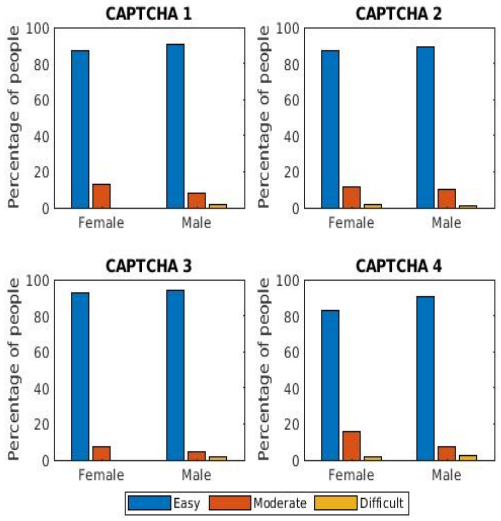


Figure. 19: Clarity in understanding Instruction Statement vs Gender for all CAPTCHA Schemes

2.

4) Effect of Gender on Clarity in understanding Shapes/Colors

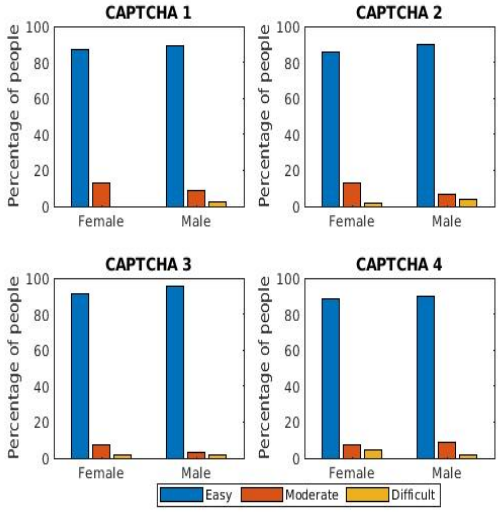


Figure. 20: Clarity in understanding Shapes/Colors vs Gender for all CAPTCHA Schemes

Figure 20 shows how male users have a higher percentage of finding the shapes or colors easy in all the four CAPTCHAs. The colors used in Sum_Color based MACS-TCHA and Alpha_Color based MACS-TCHA and the shapes used in Sum_Shape based MACS-TCHA and Alpha_Shape based MACS-TCHA were very basic colors and shapes which were easier to understand for all groups of users.

5) Effect of Gender on Clarity of characters inside Shapes

Figure 21 shows the percentage of people responding to the clarity of characters which are inside shapes, who are categorized on the basis of their gender. In all the four CAPTCHAs mentioned in Figure 21, it is observed that the male users seem to find the clarity of characters to be easy to detect and answer in comparison to female users.

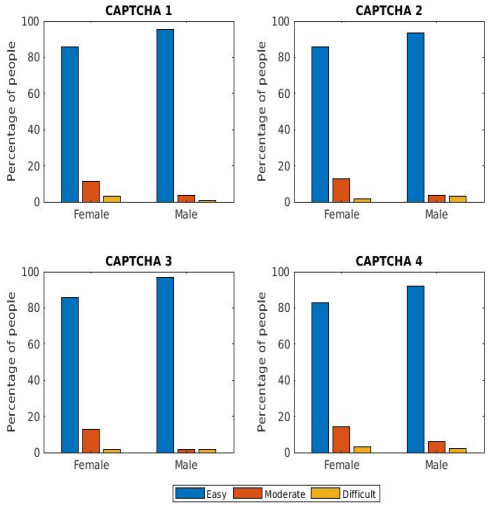


Figure. 21: Clarity of Characters inside Shapes vs Gender for all CAPTCHA Schemes

Since there could have been some confusion between letters like ‘q’,’g’, and ‘9’, a few female users found Alpha_Color based MACS-TCHA and Alpha_Shape based MACS-TCHA moderately difficult and this value was much lower in case of male users.

C. Analysis of Age groups based on each Usability factors

1) Effect of Age on Hit Ratio

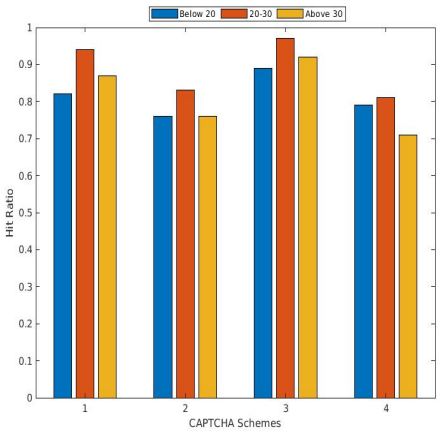


Figure. 22: Hit ratio vs CAPTCHA Schemes based on Age

The Hit Ratio for all the four CAPTCHAs based on the age of the respondent is depicted in Figure 22. According to our analysis, Table 3 shows that the people falling under the age group of 20-30 had the maximum success when it comes to solving the CAPTCHA. This was followed by the respondents with age above 30 for all but, the last CAPTCHA. In Alpha_Color based MACS-TCHA and Alpha_Shape based MACS-TCHA, the Hit Ratio is less as compared to the other two
CAPTCHAs. This shows that in the CAPTCHAs containing alphanumeric values, people of all age categories were

Table 3: Age vs Usability Factors for all CAPTCHA Schemes

		Age											
		Below 20				20-30				Above 30			
		CAPTCHA Scheme				CAPTCHA Scheme				CAPTCHA Scheme			
		1	2	3	4	1	2	3	4	1	2	3	4
Ease of use	**easy**	96.18	93.13	95.42	91.60	86.11	86.11	97.22	88.89	89.47	86.84	89.47	81.58
	medium	3.05	6.11	3.82	6.87	13.89	11.11	2.78	11.11	7.90	13.16	7.90	15.79
	hard	0.76	0.76	0.76	1.53	0.00	2.78	0.00	0.00	2.63	0.00	2.63	2.63
Clarity of Instruction Set	**easy**	90.84	91.60	94.66	90.08	86.11	83.33	97.22	88.89	86.84	81.58	86.84	78.95
	medium	8.40	7.63	5.34	8.40	13.59	13.89	2.78	11.11	10.53	18.42	7.89	15.79
	hard	0.76	0.76	0.00	1.53	0.00	2.78	0.00	0.00	2.63	0.00	5.26	5.26
Clarity in understanding Colors and Shapes	**easy**	91.60	92.37	95.42	92.37	86.11	86.11	97.22	86.11	78.95	76.32	86.84	84.21
	medium	6.87	5.34	3.82	4.58	13.59	11.11	2.78	13.89	18.42	18.42	7.89	13.16
	hard	1.53	2.29	0.76	3.05	0.00	2.78	0.00	0.00	2.63	5.26	5.26	2.63
Clarity of characters	**easy**	93.89	92.37	95.42	91.60	91.67	88.89	97.22	88.89	86.84	86.84	89.47	78.95
	medium	5.34	4.58	3.82	6.11	5.55	11.11	2.78	11.11	10.53	10.53	5.26	15.79
	hard	0.76	3.05	0.76	2.29	2.78	0.00	0.00	0.00	2.63	2.63	5.26	5.26
Hit Ratio		0.82	0.76	0.89	0.79	0.94	0.83	0.97	0.81	0.87	0.76	0.92	0.71

giving more numbers of wrong answers as compared to the other CAPTCHAs.

2) Effect of Age on Ease of Use

In Figure 23 given below, in all the four CAPTCHAs, the respondents below the age of 20 found the CAPTCHAs easier in comparison to other age groups. For the Age group 20-30, there were still less but a decent amount of users found the CAPTCHAs moderately difficult to use. This number went slightly higher for the respondents above the age of 30 in the case of Sum_Color based MACS-TCHA, Alpha_Color based MACS-TCHA and Alpha_Shape based MACS-TCHA.

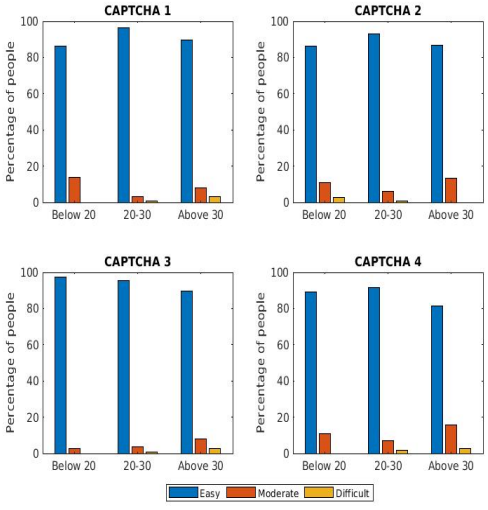


Figure. 23: Ease of use vs Age for all CAPTCHA Schemes

3) Effect of Age on Clarity in understanding Instruction Statement

The percentage of people who faced difficulty in understanding the instruction statement and whether they found it easy, moderate, or difficult, based on their age is depicted in Figure 24. It is observed that almost all the four graphs show that users didn't face any significant difficulty. Figure 24 shows a trend that as the age group increased the percentage of people who found the shapes or colors easy to understand decreased and the percentage of people who found it moderate, increased which is also evident from Table 3. This trend was strictly seen in the case of Alpha_Color based MACS-TCHA and Alpha_Shape based MACS-TCHA.

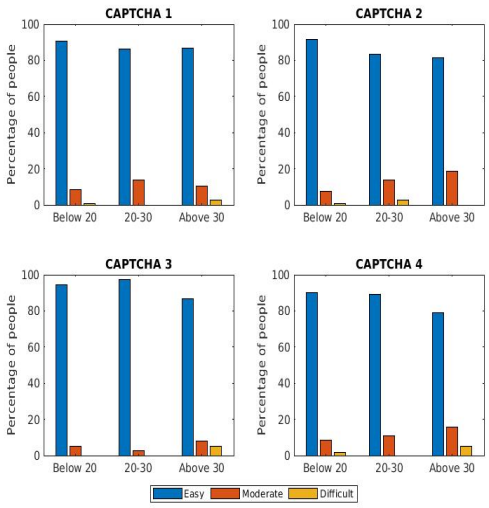


Figure. 24: Clarity in understanding Instruction Statement vs Age for all CAPTCHA Schemes

4) Effect of Age on Clarity in understanding Shapes/Colors

The percentage of people who faced difficulty in understanding the shapes or colors and whether they found it easy, moderate, or difficult, based on their age is presented in Figure 25. In almost all the four graphs, we find that the users didn't face any significant difficulty. The graphs show a trend that as the age group increased the percentage of people who found the shapes or colors easy to understand, decreased and the percentage of people who found it moderate, increased. This trend was strictly seen in the case of Sum_Color based MACS-TCHA, Alpha_Color based MACS-TCHA and Alpha_Shape based MACS-TCHA.

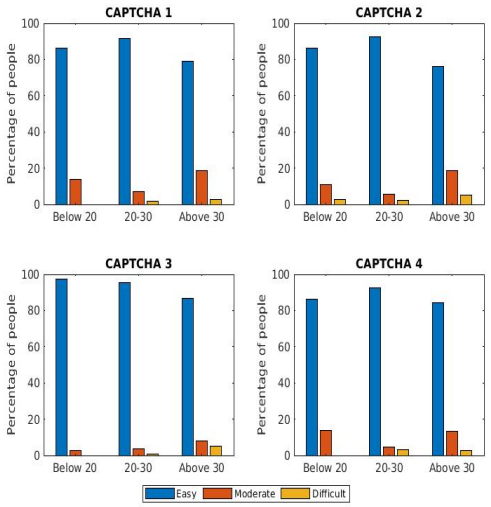


Figure. 25: Clarity in understanding Shapes/Colors vs Age for all CAPTCHA Schemes

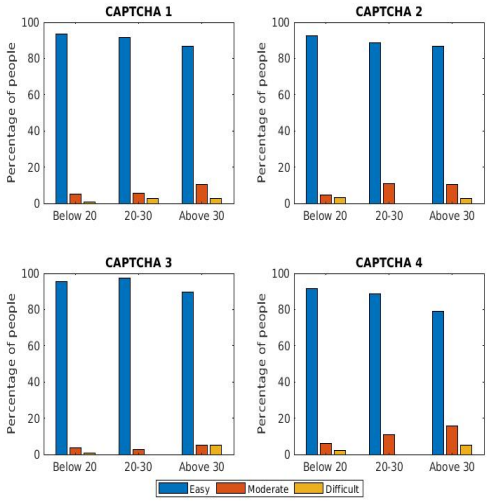


Figure. 26: Clarity of Characters inside Shape vs Age for all CAPTCHA Schemes

5) Effect of Age on Clarity of Characters inside Shapes

Almost all the four graphs in Figure 26 show that users didn't face any difficulty in the CAPTCHAs. We observe a trend in all the graphs that as the age group increased, the percentage of people who found the shapes or colors easy to understand decreased and the percentage of people who found it moderate or difficult, increased. This trend was strictly seen in the case of Alpha_Shape based MACS-TCHA.

D. Analysis of Professions of each usability factors

1) Effect of Profession on Hit Ratio

The Hit Ratio based on the profession of the user, for the four CAPTCHA schemes is depicted in Figure 27. According to the analysis done using Table 4, for the Sum_Color based MACS-TCHA and Alpha_Shape based MACS-TCHA, university students and working professionals had a much higher hit ratio as can be seen from Figure 27. For the

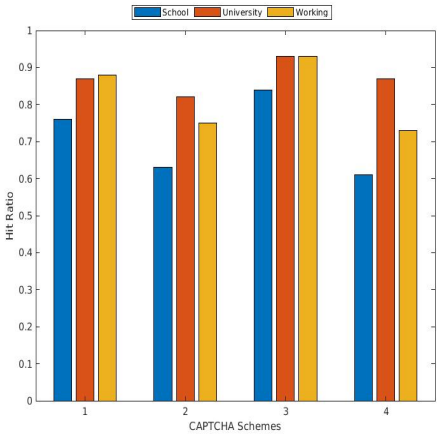


Figure. 27: Hit ratio vs CAPTCHA Schemes based on Profession

Alpha_Color based MACS-TCHA and Alpha_Shape based MACS-TCHA, university students scored the highest, followed by working professionals.

2) Effect of Profession on Ease of Use

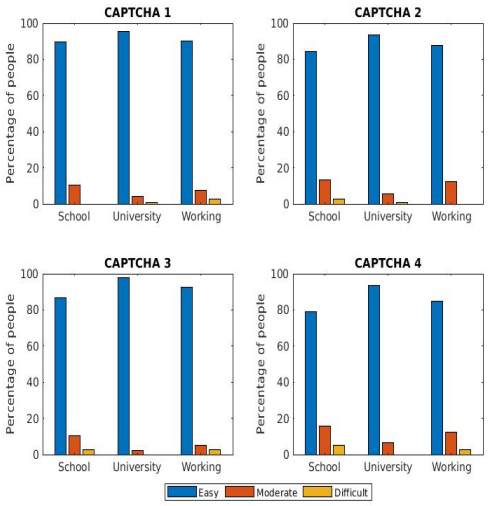


Figure. 28: Ease of Use vs Profession for all CAPTCHA schemes

The respondents were categorized into 3 categories according to their profession, the categories being, School, University, and Working profession as given in Figure 28. The graph shows that the users belonging to the category of University found the CAPTCHAs easier to solve than other categories of users. Further, it was found that school students find the CAPTCHAs comparatively more difficult than the other category of users. It is also seen that working professionals found Alpha_Color based MACS-TCHA and Alpha_Shape based MACS-TCHA lightly more difficult than the other two CAPTCHAs.

Table 4: **Profession vs Usability factors for all CAPTCHA Schemes**

		Profession											
		School Student				University Student				Working Professional			
		CAPTCHA Scheme				CAPTCHA Scheme				CAPTCHA Scheme			
		1	2	3	4	1	2	3	4	1	2	3	4
Ease of use	**easy**	89.47	84.21	86.84	78.95	95.24	93.65	97.60	93.65	90.00	87.50	92.50	85.00
	medium	10.53	13.16	10.53	15.89	3.97	5.56	2.38	6.35	7.50	12.50	5.00	12.50
	hard	0.00	2.63	2.63	5.26	0.79	0.79	0.00	0.00	2.50	0.00	2.50	2.50
Clarity of Instruction Set	**easy**	86.84	84.21	89.47	76.32	95.24	92.06	96.03	92.06	87.50	80.00	90.00	82.50
	medium	13.16	13.16	10.53	21.05	3.97	7.15	3.97	7.14	10.00	20.00	5.00	12.50
	hard	0.00	2.63	0.00	2.63	0.79	0.79	0.00	0.79	2.50	0.00	5.00	5.00
Clarity in understanding Colors and Shapes	**easy**	86.84	81.58	86.84	84.21	96.03	92.06	97.60	89.68	82.50	80.00	90.00	87.50
	medium	10.53	13.16	10.53	74.89	3.18	6.35	2.38	9.52	15.00	15.00	5.00	10.00
	hard	2.63	5.26	2.63	74.89	0.79	1.59	0.00	0.79	2.50	5.00	5.00	2.50
Clarity of characters	**easy**	81.58	84.21	86.84	78.95	96.03	92.06	97.60	93.65	90.00	90.00	92.50	82.50
	medium	15.79	13.16	10.53	15.89	3.18	5.56	2.38	5.56	7.50	7.50	2.50	12.50
	hard	2.63	2.63	2.63	5.26	0.79	2.38	0.00	0.79	2.50	2.50	5.00	5.00
Hit Ratio		0.76	0.63	0.84	0.61	0.87	0.82	0.93	0.87	0.88	.0.75	0.93	0.73

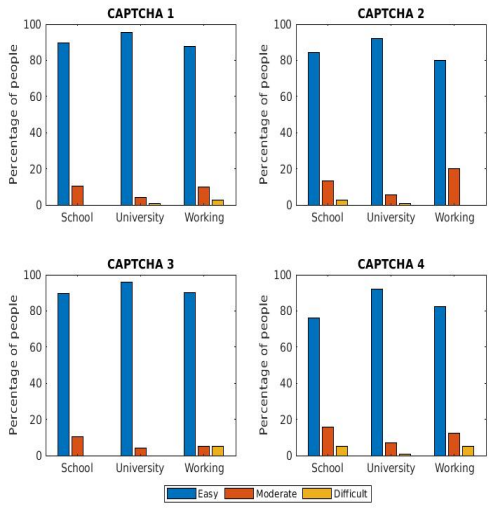


Figure. 29: Clarity in understanding Instruction Statement vs Profession for all CAPTCHA schemes

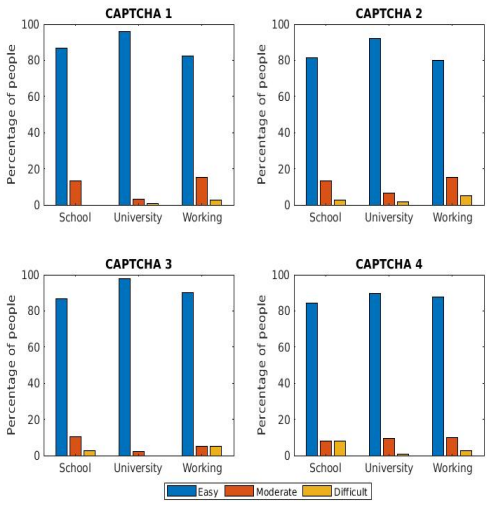


Figure. 30: Clarity in understanding Shapes/Colors vs Profession for all CAPTCHA schemes

3) Effect of Profession on Clarity in understanding Instruction Statement

Figure 29 depicts the graph showing the percentage of people finding the instruction statement easy, moderate or difficult. It is clearly shown that the university students found all the CAPTCHAs easier in comparison to other professions. The working professionals group found the CAPTCHAs moderately difficult despite having a huge percentage favouring easy on the difficulty range. There were a few students and working professionals that found Alpha_Color based MACS-TCHA and Alpha_Shape based MACS-TCHA difficult in comparison to the rest.

4) Effect of Profession on Clarity in understanding Shapes/ Colors

The percentage of people who faced difficulty in understanding the shapes or colors and whether they found it easy, moderate, or difficult, based on their profession is given in Figure 30. Although almost all the four graphs show that users did not face any difficulty, the graphs show a trend that the university students generally faced less difficulty in understanding the shapes and colors than the other users. Further, it was observed that people found Sum_Color based MACS-TCHA and Alpha_Color based MACS-TCHA a bit more difficult to understand than other CAPTCHAs, as they involved colors.

5) Effect of Profession on Clarity of Characters inside Shape

In the given graphs in Figure 31, in all the four CAPTCHAs, we can observe that the University users seem to find the clarity of characters to be easy to detect and answer in comparison to the other two category users. Further the school category students faced a bit more difficulty than the other two category users. As seen from Figure 31, difficulty was slightly more in the case of Sum_Color based MACS-TCHA and Alpha_Shape based MACS-TCHA.

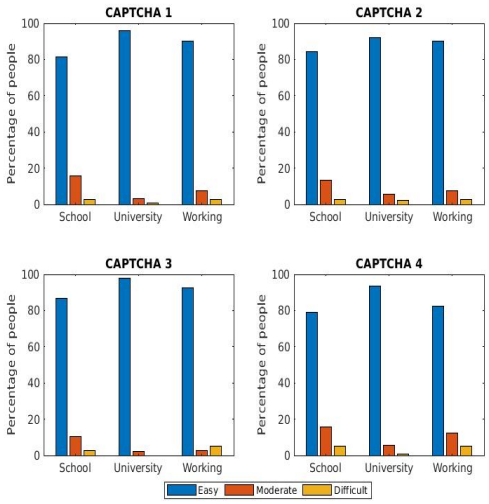


Figure. 31: Clarity of character inside shapes vs Professions for all CAPTCHA schemes

V. Reliability Analysis and Robustness of MACS-TCHA Schemes

Reliability is the degree to which the result of a measurement, calculation, or specification is deemed to be accurate and the analysis of this accuracy is termed as reliability analysis. This helps in defining the novelty and reliability of a system. On the other hand robustness is the degree to which a system continues to function in the presence of invalid inputs or stressful environmental conditions. In general, the reliability analysis for a text CAPTCHA system consists of machine learning and deep learning models that are trained to successfully bypass the system. This is done by removing the noise to recognize the alpha-numeric characters and feeding these to the system to get the correct output. In order to show that our MACS-TCHA's are robust/reliable to a bot attack, we have theoretically compared the efficiency of the design of our CAPTCHA schemes with inferences from some of the existing deep learning models. Since MACS-TCHA is a multilayered model, the first step to reliability analysis would be to define the problems to break individual layers.

The different tasks that are necessary to generate a valid output for the MACS-TCHA system are given below:

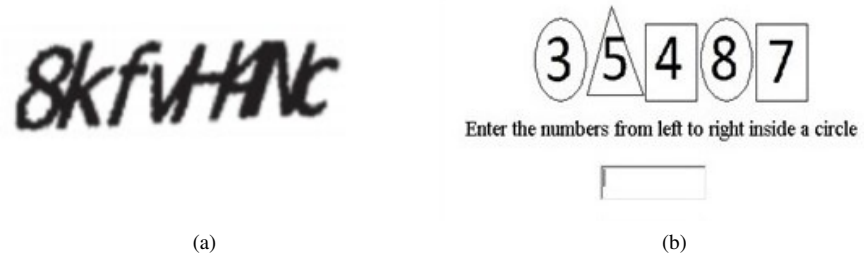
- Recognizing the alphanumeric characters
- Recognizing the colored circles in the variations of Numeric_color based MACS-TCHA
- Recognizing the shapes in the variations of Numeric_shape based MACS-TCHA
- Recognizing the color/shape from the instruction statement

A. Recognizing the alphanumeric characters

There are multiple machine learning and deep learning models that have successfully bypassed the existing text CAPTCHA systems. Most of these models follow a pattern to recognize the correct output. The models start by removing the unnecessary noise, followed by character recognition. [27] Jun Chen et al. have compared the different techniques that can be used to break text based CAPTCHAs in their article published in 2017. The text CAPTCHA, program generation CAPTCHA as seen in Figure 32 (a) closely resembles the font of MACS-TCHA as depicted in Figure 32 (b). Jun Chen et al. have surveyed that by applying a RNN model on a text CAPTCHA, a success rate of 55% can be easily achieved.
This accuracy can be increased if we apply more recent deep learning models that have emerged over the past few years. Although, there will still be some hindrance because of the different shapes around the characters in MACS-TCHA. Because most algorithms are using segmentation to differentiate the characters from noise, the shapes in MACS-TCHA will be a hindrance in recognition. It will be safe to assume that recognizing the characters will not be very difficult for MACS-TCHA. The existing models have a high accuracy of breaking CAPTCHAs with irregular fonts. Since we wanted to maintain human readability and have kept the font very simple and legible, it will be a relatively easy task to recognize the characters. Although this is not the only thing required for solving MACS-TCHA. The following subsections are equally important and discuss the greater barriers in breaking MACS-TCHA.

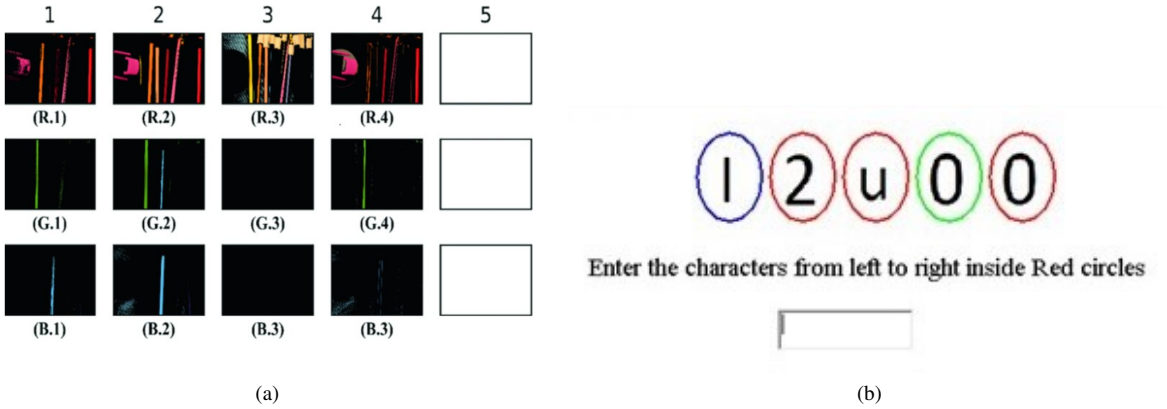
B. Recognizing the colored circles in the variations of Numeric_color based MACS-TCHA

Detection of thin color boundaries is a difficult task even for some of the new deep learning algorithms. The low density of pixels makes it difficult for models to recognize the colors. Forero et al. analyzed the extreme color detection methods to detects the color of thin pencils [27]. As seen in Figure 33 (a), the columns correspond to the methods studied by them, the rows correspond to the color channels R for red, G for green and B for blue respectively. Column (1) corresponds to the proposed method,while the other columns are results of the different models analyzed in the paper. The empty boxes correspond to cases where the corresponding method does not provide for the detection of that color. The limitation here is that the model detects a single color, where the time taken for detection is more than 1 second. As seen in Figure 33 (b) , for MACS-TCHA, the model should recognize multiple colors at the same time while maintaining the order in which they appear.
This model can be modified to find the color of circles in Numeric_color based MACS-TCHA, but the success rate would not be very high. Since MACS-TCHA has 3 different colors, it will be difficult to recognize all at the same time and map them to the correct position.



(a) (b)

Figure. 32: Program Generation CAPTCHA Vs Our proposed NumericShape based MACS-TCHA.



(a) (b)

Figure. 33: Color Detection for Model vs Colored circles in MACS-TCHA

C. Recognizing the shapes in the variations of Numeric_shape based MACS-TCHA

There are many models which can recognize shapes which are filled with colors. Most algorithms use neighbouring pixels to determine the boundaries of filled shapes. The image is first converted into gray scale and then the resulting image is fed to deep learning models like ANN and SVM for classification, giving a very high accuracy as analyzed by Chandra et al. [28]. The results for their work are shown in Figure 34 (a). This can be used to recognize the shapes in MACS-TCHA, though there are a few modifications required in the algorithm used. As seen in Figure 34 (b), in the case of MACS-TCHA, there is a very thin boundary and they are not filled with colors, rather they contain a character inside them. This makes it difficult for deep learning models to recognize the shapes with characters inside.

Contrary to recognizing the characters, here the shapes are not noise elements. In fact the characters are a hindrance in the recognition process and must be considered as noise in order to recognize the shapes efficiently. This increases the task complexity by a great amount. Methods like segmentation would prefer the numbers over the thin bordered shapes. This will decrease the accuracy of the current models drastically.

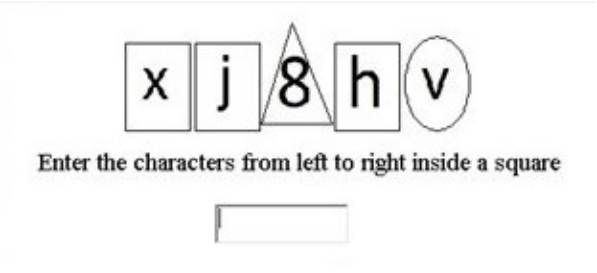
D. Recognizing the color/shape from the instruction statement

The model has to extract the color/shape from the instruction statement given in MACS-TCHA. This can be done using machine and deep learning models with a high accuracy. But after extracting the color/shape from the MACS-TCHA, the model has to map this with the information it gets from the above parameters sequentially, which increases the complexity.

As discussed in the above subsections, there are a few layers that can be detected using complex models. At the same time, there are layers which will offer high resistance in detection. Even if a complex model is designed to detect all the above layers, there is one other constraint that will increase the complexity manifold. The fact that any model would have to map the findings from each layer and then reach an optimal answer makes MACS-TCHA difficult to break. For example, if a model for recognising characters is applied to the image, it will only extract the characters. Separate models will be required to extract the information from other layers. This brings light to the final constraint, where no single existing model can break MACS-TCHA. Multiple models with equally high complexity and modifications will have to be combined in order to break MACS-TCHA. Also, the combination of models would have to ensure that all the colors, shapes or characters are mapped to proper location according to the output image. The complexity of such a model is very high because the mapping cannot be wrong in any way. If even a single position mapping is wrong in any of the layers, it will result in an incorrect output, resetting MACS-TCHA. This shows the novelty, reliability and robustness of MACS-TCHA.

Input Image	Final Processed Image for Counting Object	Desired number of Objects	Detected Objects	Accuracy (%)
		4	4	100
		4	4	100
		4	4	100
		5	4	80
		4	3	75

(a)



(b)

Figure. 34: Shapes for Deep Learning Model vs Shapes in MACS-TCHA

E. Ablation Study

The proposed MACS-TCHA schemes deal with different layers and parameters involving shapes, colors, instruction statements and alpha-numeric characters. Removing a single parameter or layer could greatly decrease the security of MACS-TCHA. The layers work parallel to each other, contributing equally to the answer retrieval mechanism used in the algorithm. Reducing the number of layers, will make MACS-TCHA vulnerable to Deep learning techniques, hence breaking the purpose of using multi-layer randomness approach. Taking an example, if we remove a layer like the randomization of instruction statement, and fix the instruction statement for all the schemes, a deep learning model would easily penetrate the combination of the remaining layers. Similarly, the random nature of the other layers, used in combination, helps enhance the security of MACS-TCHA and greatly reduces the vulnerabilities to bot attacks.

In contrast to MACS-TCHA, Chaurasia et al. [6] have used parameters such as number of curvy lines, confusing characters, distortion, etc. Alejandro et al. [43] propose different metrics to parameterize their CAPTCHA. By using metrics like Gyroscope Readings, Touch, Accelerometer, Gravity and Microphone readings, the authors create a new form of text CAPTCHA. Similarly, other CAPTCHAs have different parameters and comparing the metrics of one CAPTCHA to another is impractical. The Parameters used in the recent years are very subjective to a text CAPTCHA. We strongly infer that since CAPTCHA designs have outreached the basic distorted text and noise, comparing such a wide range of parameters with inputs of MACS-TCHA is not reasonable.

VI. Human Perception Analysis from the Usability evaluations of MACS-TCHA Schemes

Human Perception can be defined as a process of recognizing, organizing and interpreting information human's sense through their sensory organs. Perception of every human being is distinct from one another as different types of perception involve different signals. These are dealt with by the nervous system of a human being. Such signals are generated differently as the sensory organs of humans have different sensitivity. So the viewpoint on a similar topic can differ for each individual due to the various factors involved in the process.
There is a pattern in the mistakes which can be recognised after analyzing the user evaluation studies. They seem to be of a repetitive nature, indicating the commonalities in the way humans perceive numbers, alphabets, colors and shapes. After compressing all incorrect responses into specific categories, a few observations can be derived out of the usability evaluation studies of our MACS-TCHA schemes.

- The very first mistake people make while replicating alphanumeric characters in Alpha_Color based MACS-TCHA and Alpha_Shape based MACS-TCHA, is the confusion of letters like I, l and 1. Due to a very little difference in the structure of the above characters, there occurs a discrepancy in the accuracy of the correct response. Similar to the above set, characters like 9, g and q also pose a similar difficulty to the user. Though we reduced the first set of discrepancy by involving only small alphabets, this is a common mistake that accounts for a large percentage of incorrect responses.

- Another common mistake that people make is to write the whole sequence of alphanumeric sequences they see, without even reading the instruction statement given to them. It is also observed that people who read the instruction statement also made mistakes as they omit some words while reading, due to lack of concentration. For example, in Sum_Color based MACS-TCHA which asked for the sum of the numbers encircled by a certain color, people wrote the sequence of numbers which are encircled correctly.

- Some people get confused between the different shapes and colors, and give incorrect answers. This may happen due to various reasons, such as color-blindness,

lower resolution of the display, poor eyesight, negligence, etc. A common confusion is the inability to differentiate between squares and rectangles in Sum_Shape based MACS-TCHA and Alpha_Shape based MACS-TCHA.

- A few incorrect responses consist of words that are formed out of the alphabets displayed to the user. This kind of mistake occurs usually when all the alphanumeric characters consist of alphabets. Although the instruction statement clearly states the intent of the respective CAPTCHAs, some users try to create a word out of the selected alphabets in Alpha_Color based MACS-TCHA and Alpha_Shape based MACS-TCHA. The likelihood of this mistake to occur at a regular interval is not too high, as the chances for all the characters to be alphabets is very less, considering the algorithm generates both alphabets and numbers simultaneously.
- Many people tend to answer the CAPTCHA from their mobile phones. One of the functionalities of smartphones is that whenever we write using the virtual keyboard, the first letter is capitalized by default. As the CAPTCHAs are case sensitive, this feature can often lead people to answer Alpha_Color based MACS-TCHA and Alpha_Shape based MACS-TCHA incorrectly.
- Some people are unable to determine the required format of the input and tend to insert commas, spaces, or other separators between each character, which leads to the submission of an incorrect answer.

VII. Conclusions, Practical Implications and Future Directions

In this research, we have designed and developed various text CAPTCHAs by manipulating colors, shapes and alpha numerals in order to achieve better usability for the users. Our work evaluates the usability of our novel CAPTCHA named MACS-TCHA through a detailed survey analysis. The survey studies have shown that the usability of the text-based CAPTCHA schemes have significantly improved after using the MACS-TCHA. We performed a detailed theoretical analysis on the Reliability and Robustness of the proposed MACS-TCHA schemes with the existing deep learning models in order to thwart recognition attacks from the bots.

The proposed MACS-TCHA schemes can be used by different age groups, and does not restrict the usage to highly literate groups of people. Even individuals with limited education can solve the CAPTCHA since it requires only elementary knowledge. Our proposed MACS-TCHA schemes have several applications in the real time security domains including use cases like Web Registration, Online Polling and Mitigating Comment Spam. It has extensive usage in the privacy protection of various downstream applications in the networking domains of the World Wide Web. MACS-TCHA's does not have any distortion which will prevent the irritation faced by user when solving a text CAPTCHA with distortion. The limitation for this work was that we were unable to apply any existing CAPTCHA breaking models to our method because of the difference in the inherent nature of the design. To develop a novel CAPTCHA breaking method for MACS-TCHA using Supervised and Unsupervised learning strategies is one of the interesting future thread of research. It would be better if we could test the effects of slight noise and distortion in the existing MACS-TCHA schemes to understand its robustness. Another area of research would be to reproduce the algorithm for graphical based CAPTCHA rather than a text CAPTCHA since it reduces the human effort even further.

VIII. Declarations

A. Ethical Approval

This declaration is not applicable.

B. Compteting Interests

The authors do not have any competing interests.

C. Author's Contributions

Navansh Goel, Tejaswi Kumar and C. Oswald worked on the idea. Navansh Goel and Tejaswi Kumar implemented the idea and the writeup of the paper. C. Oswald reviewed the paper.

D. Funding

This declaration is not applicable.

E. Availability of data and materials

The collected data will be available on request.

References

[1] Alan Dix - Janet Finlay - Gregory Abowd - Russell Beale, Human Computer Interaction - 3rd Edition, PRENTICE HALL 2004.

[2] Luis von Ahn, Manuel Blum and John Langford, Telling Humans and Computers Apart (Automatically), Communications of the ACM, Volume 47 Issue 2, Pages 56-60, February 2004.

[3] T.Y. Chan, "Using a text-to-speech synthesizer to generate a reverse Turing test", Proc. IEEE Int'l Conf. Tools with Artificial Intelligence (ICTAI 2003), pp. 226-232, November 2003.

[4] Prof. Anisaara Nadaph, Juwairiya Shaikh, Nikita Bodhe, Hemlata Pingale, Mrunali khunte, "Video CAPTCHA – Design Based on Moving Object Recognition", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 4, Issue 4, April 2016.

[5] Kumar T., Goel N., Roy S., Oswald C. (2022) Usability Evaluation of Novel Text CAPTCHA Schemes Based on Colors and Shapes. In: Khanna A., Gupta D., Bhattacharyya S., Hassanien A.E., Anand S., Jaiswal A.

(eds) International Conference on Innovative Computing and Communications. Advances in Intelligent Systems and Computing, vol 1388. Springer, Singapore. https://doi.org/10.1007/978-981-16-2597-8_30

[6] K. Chaurasia, S. Jain, B. Sivaselvan and C. Oswald, "Automatic Ranking of CAPTCHAs based on Usability Measures," 2017 14th IEEE India Council International Conference (INDICON), Roorkee, 2017, pp. 1-6, doi: 10.1109/INDICON.2017.8487993.

[7] Hasan, Walid. (2016). A Survey of Current Research on CAPTCHA. International Journal of Computer Science Engineering Survey. 7. 1-21. 10.5121/ijcses.2016.7301.

[8] G. Mori and J. Malik, "Recognizing objects in adversarial clutter: breaking a visual CAPTCHA," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 134–144, June 2003.

[9] Saini, Baljit Singh. (2013). A Review of Bot Protection using CAPTCHA for Web Security. IOSR Journal of Computer Engineering. 8. 36-42. 10.9790/0661-0863642.

[10] .-S. Cui, J.-T. Mei, X. Wang, D. Zhang, and W.-Z. Zhang. A CAPTCHA implementation based on 3d animation. In Proceedings of the 2009 International Conference on Multimedia Information Networking and Security - Volume 02, MINES '09, pages 179–182, Washington, DC, USA, 2009. IEEE Computer Society.

[11] J.-S. Cui, J.-T. Mei, W.-Z. Zhang, X. Wang, and D. Zhang. A CAPTCHA implementation based on moving object recognition problems. In ICEE, pages 1277–1280. IEEE, 2010

[12] Chow, Yang-Wai and Susilo, Willy: AniCAP: An animated 3D CAPTCHA scheme based on motion parallax 2011, 255-271.

[13] Nguyen, Vu Chow, Yang-Wai Susilo, Willy. (2012). Breaking an animated CAPTCHA scheme. 12-29. 10.1007/978-3-642-31284-7_2.

[14] M. Shirali-Shahreza and S. Shirali-Shahreza, "Question-Based CAPTCHA", Conference on Computational Intelligence and Multimedia Applications, 2007. International Conference on, Sivakasi, TamilNadu.2007, Volume 4, Page no. 54-58.

[15] L. von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum, ReCAPTCHA: Human-based character recognition via web security measures, Science, vol. 321, no. 5895, pp. 1465âA ¸ S1468, 2008.

[16] Gafni, R. and Nagar, I., 2016. CAPTCHA–Security affecting user experience. Issues in Informing Science and Information Technology, 13, pp.063-077.

[17] Sivakorn, S., Polakis, J. and Keromytis, A.D., 2016. I'm not a human: Breaking the Google reCAPTCHA. Black Hat, pp.1-12.

[18] Vidya, P.N., and Naika, S., 2015. Simple text-based CAPTCHA for the security in web applications. Int. J. Comput. Sci. Mob. Comput.

[19] Imsamai, M., Phimoltares, S. (2010). 3D CAPTCHA: A Next Generation of the CAPTCHA. 2010 International Conference on Information Science and Applications. doi:10.1109/icisa.2010.5480258

[20] Chow, Richard Golle, Philippe Jakobsson, Markus Wang, Lusha Wang, Xiaofeng. (2008). Making CAPTCHAs clickable. 91-94. 10.1145/1411759.1411783.

[21] Jaint, Nidhi Verma, Punam Mittal, Sunita Mittal, Sanjeev Singh, Anand Munjal, Shashi. (2010). Gender-based alteration in color perception. Indian journal of physiology and pharmacology. 54. 366-70.

[22] Rich Gossweiler, Maryam Kamvar, and Shumeet Baluja. 2009. What's up CAPTCHA? a CAPTCHA based on image orientation. In Proceedings of the 18th international conference on the World wide web (WWW '09). Association for Computing Machinery, New York, NY, USA, 841–850.

[23] Elie Bursztein, Matthieu Martin, and John Mitchell. 2011. Text-based CAPTCHA strengths and weaknesses. In Proceedings of the 18th ACM conference on Computer and communications security (CCS '11). Association for Computing Machinery, New York, NY, USA, 125–138.

[24] X. Ling-Zi and Z. Yi-Chun, "A Case Study of Text-Based CAPTCHA Attacks," 2012 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Sanya, 2012, pp. 121-124, doi: 10.1109/CyberC.2012.28.

[25] Khawandi, Shadi Abdallah, Firas Ismail, Anis. (2019). A Survey On The Different Implemented Captchas. 01-11. 10.5121/csit.2019.90101

[26] Bursztein, Elie Bethard, Steven Fabry, Celine Mitchell, John Jurafsky, Daniel. (2010). How good are humans at solving CAPTCHAs? A large scale evaluation. Proceedings - IEEE Symposium on Security and Privacy. 399-413. 10.1109/SP.2010.31.

[27] Chen, Jun, Xiangyang Luo, Yanqing Guo, Yi Zhang, and Daofu Gong. "A survey on breaking technique of text-based CAPTCHA." Security and Communication Networks 2017 (2017).

[28] Forero M.G., Ávila-Navarro J., Herrera-Rivera S. (2020) New Method for Extreme Color Detection in Images. In: Figueroa Mora K., Anzurez Marín J., Cerda J., Carrasco-Ochoa J., Martínez-Trinidad J., Olvera-López J. (eds) Pattern Recognition. MCPR 2020. Lecture Notes in Computer Science, vol 12088. Springer, Cham. https://doi.org/10.1007/978-3-030-49076-8_9

[29] Chandra, Abir Hossin, Khairat Uddin, Md. Palash Mamun, Md. Abdulla Afjal, Masud Ibn Nitu, Adiba. (2019). Detection and Classification of Geometric Shape Objects for Industrial Applications.

[30] Wei, T.E., Jeng, A.B. and Lee, H.M., 2012, December. GeoCAPTCHA—A novel personalized CAPTCHA using geographic concept to defend against 3 rd Party Human Attack. In 2012 IEEE 31st International Performance Computing and Communications Conference (IPCCC) (pp. 392-399). IEEE.

[31] Zhu, Bin Yan, Jeff Bao, Guanbo Yang, Maowei xu, Niu. (2014). Captcha as Graphical Passwords—A New Security Primitive Based on Hard AI Problems. IEEE Transactions on Information Forensics and Security. 9. 891-904. 10.1109/TIFS.2014.2312547.

[32] Kaur, Kiranjot Behal, Sunny. (2015). Designing a Secure Text-based CAPTCHA. Procedia Computer Science. 57. 122-125. 10.1016/j.procs.2015.07.381.

[33] H. Gao, M. Tang, Y. Liu, P. Zhang and X. Liu, "Research on the Security of Microsoft's Two-Layer Captcha," in IEEE Transactions on Information Forensics and Security, vol. 12, no. 7, pp. 1671-1685, July 2017, doi: 10.1109/TIFS.2017.2682704.

[34] G. Moy, N. Jones, C. Harkless and R. Potter, "Distortion estimation techniques in solving visual CAPTCHAs," Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004., 2004, pp. II-II, doi: 10.1109/CVPR.2004.1315140.

[35] Uma, P., Siddivinayak, K. and Ramachandra, P., 2019. Smart captcha to provide high security against bots. In Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering (pp. 3-5).

[36] Li, S., Shah, S.A.H., Khan, M.A.U., Khayam, S.A., Sadeghi, A.R. and Schmitz, R., 2010, December. Breaking e-banking CAPTCHAs. In Proceedings of the 26th Annual Computer Security Applications Conference (pp. 171-180).

[37] Gao, H., Wang, W., Qi, J., Wang, X., Liu, X. and Yan, J., 2013, November. The robustness of hollow CAPTCHAs. In Proceedings of the 2013 ACM SIGSAC conference on Computer communications security (pp. 1075-1086).

[38] Conti, M., Guarisco, C. and Spolaor, R., 2016, June. CAPTCHaStar! A novel CAPTCHA based on interactive shape discovery. In International Conference on Applied Cryptography and Network Security (pp. 611-628). Springer, Cham.

[39] von Ahn L., Blum M., Hopper N.J., Langford J. (2003) CAPTCHA: Using Hard AI Problems for Security. In: Biham E. (eds) Advances in Cryptology — EUROCRYPT 2003. EUROCRYPT 2003. Lecture Notes in Computer Science, vol 2656. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-39200-9_18

[40] Gao, H., Yao, D., Liu, H., Liu, X. and Wang, L., 2010, December. A novel image based CAPTCHA using jigsaw puzzle. In 2010 13th IEEE International Conference on Computational Science and Engineering (pp. 351-356). IEEE.

[41] Saha, R., Geetha, G. and Lee, G.S., 2011, December. CLAPTCHA-A novel captcha. In International Conference on Security Technology (pp. 94-100). Springer, Berlin, Heidelberg.

[42] Aditya Atri, Ankita Bansal, Manju Khari, S. Vimal, De-CAPTCHA: A novel DFS based approach to solve CAPTCHA schemes, Computers Electrical Engineering, 2021, 107593, ISSN 0045-7906, https://doi.org/10.1016/j.compeleceng.2021.107593

[43] Acien, A., Morales, A., Fierrez, J., Vera-Rodriguez, R., & Delgado-Mohatar, O. (2021). BeCAPTCHA: Behavioral bot detection using touchscreen and mobile sensors benchmarked on HuMIdb. Engineering Applications of Artificial Intelligence, 98, 104058.

[44] Ahmet Ali Süzen, UNI-CAPTCHA: A Novel Robust and Dynamic User-Non-Interaction CAPTCHA Model Based on Hybrid biLSTM+Softmax, Journal of Information Security and Applications, Volume 63, 2021, 103036, ISSN 2214-2126, https://doi.org/10.1016/j.jisa.2021.103036

[45] Bera, A., Bhattacharjee, D., & Shum, H. P. (2021). Two-stage human verification using HandCAPTCHA and anti-spoofed finger biometrics with feature selection. Expert Systems with Applications, 171, 114583.

[46] Bera, A., Bhattacharjee, D., & Nasipuri, M. (2018). Hand Biometric Verification with Hand Image-Based CAPTCHA. In Advanced Computing and Systems for Security (pp. 3-18). Springer, Singapore.

[47] Ma, Y., Zhong, G., Liu, W., Sun, J., & Huang, K. (2020). Neural CAPTCHA networks. Applied Soft Computing, 97, 106769.

[48] Yao Wang, Yuliang Wei, Mingjin Zhang, Yang Liu, Bailing Wang, Make complex CAPTCHAs simple: A fast text captcha solver based on a small number of samples, Information Sciences, Volume 578, 2021, Pages 181-194, ISSN 0020-0255, https://doi.org/10.1016/j.ins.2021.07.040

[49] Li, C., Chen, X., Wang, H., Wang, P., Zhang, Y., & Wang, W. (2021). End-to-end attack on text-based CAPTCHAs based on cycle-consistent generative adversarial network. Neurocomputing, 433, 223-236.

Author Biographies

Navansh Goel was born in Haryana on 11 June 2001. He is a Fourth year undergraduate student at the Vellore Institute of Technology, Chennai, Tamil Nadu, India. He will be graduating from his B.Tech. degree in 2023 in the field of Computer Science and Engineering with Specialization in Cyber Physical Systems.

Tejaswi Kumar was born in Bihar on 24 March 2001. He is a Fourth year undergraduate student at the Vellore Institute of Technology, Chennai, Tamil Nadu, India. He will be graduating from his B.Tech. degree in 2023 in the field of Computer Science and Engineering.

C. Oswald is an Assistant Professor in the department of Computer Science and Engineering at National Institute of Technology Tiruchirappalli. He completed his PhD from Indian Institute of Information Technology Design and Manufacturing Kancheepuram, Chennai, India.