ISSN 2347 - 3983

Volume 9. No. 7, July 2021

International Journal of Emerging Trends in Engineering Research

Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter19972021.pdf

https://doi.org/10.30534/ijeter/2021/19972021

reCAPTCHA: Human Based Embedded Image Generation and Recognition for Web Security

Khaled M. Alalayah
Department of Computer Science, College of Science and Arts, Sharurah, Najran University, Saudi Arabia
Department of Computer Science, Faculty of Science. IBB University, Yemen.
kmalalayah@nu.edu.sa
kh101ed2005@yahoo.com

ABSTRACT

CAPTCHAs (Completely Automated Public Turing test to tell Computers and Humans Apart) are well-known safety process on the internet that prevents present programs from harmful online services. It's functioning like asking humans to complete a task that computers cannot perform, such as entering in the provided box with jumbled string bits, where the string is also considered for case sensitivity. reCAPTCHA is being used as the security parameter for the websites sites oppose of automatic damage through giving random auto-created conflict for users to resolve. Provided provocation are made in form that is tough for computers to resolve yet however much easier for human being. Various techniques have evolved in the past for the successful implementation of the captcha code which is not recognizable by system but are not fit as per the ML/NLP based techniques to crack the same. In the present scenario the major challenge is to provide the captcha in a way that it goes with time and the users are able to make quick response for the same. In the paper an automatic reCAPTCHA generator is being implemented which uses the embedded part of the images and text, reCAPTCHA puzzles are on the basis of the same. The embedded images are the combination of the objects and text embedded over an image. The proposed system have outperformed to the literature techniques in terms of success rate of 97%, precision 95% and recall 93%.

Key words: reCaptcha; Turing Test; Security; Integrity; Availability; Text; Image.

1. INTRODUCTION

For determining whether a user is a human or a computer, a CAPTCHA challenge response test is used on the World Wide Web. For differentiating Humans and Computers, the acronym which stands for Completely Automated Public Turing test is regarded. Various distorted characters, which are included in a typical CAPTCHA which is an image that replicated at the bottom of Web registration forms. To "prove" the users to be human, they are requested to type the wavy characters. Hence, to abuse online services, CAPTCHAs act as sentries against automated programs, as we know distorted text can't be read by the current computers programs whereas humans can. Several types of Web sites, like free e mail providers, ticket sellers, social networks, wikis, and blogs are protected with the help of CAPTCHAs, as for owing to their effectiveness as a security measure. Like, for purchase big numbers of concert tickets, CAPTCHAs were used as it prevent ticket scalpers from using computer programs, and this is just to re sell them at an inflated price. With the help of some sites like: Gmail and Yahoo Mail, CAPTCHAs is used there as to break spammers from getting millions of free e mail accounts, which they would utilise to send spam e mail.

As per our prediction, in this world, mostly human type more than 100 million CAPTCHAs in a day (see supporting online text), and they use to spend their few seconds while typing the deformed characters. Additionally, it results towards hundreds of thousands of human hours in a day. We report on an experiment that attempts to make positive use of the time spent by humans solving CAPTCHAs. Large scale abuse of online services is secured with the help of CAPTCHAs, as if it is not secured than the mental effort of each person, which they spend for solving them, is wasted. As deciphering CAPTCHAs requires people to perform a task that computers cannot due to this mental effort is considered invaluable.

Key information is flawlessly exchanged between organizations, groups or individuals with the help of web and it is also considered as the primary way. Security challenges; threats such denial of service, XPath injection attack, amongst others continue to pose a major setback in web service security were increased with the rise in the use of the web [1][2]. Subsequently, information security, integrity, and availability have become an important component of web service provisioning. The need for security in web service inevitable is emerged by the rising proliferation of web based technology and its attendant security challenges.

Vulnerabilities in web applications may be exploited by the cyber criminals. Improper authentication and authorization are also example of such vulnerabilities. As attackers gain access to private web services, leads towards degradation and information leakage [3]. To validate the identity of the user(s) to prevent malicious users from having access to information as this will pose a threat to the original owner of the information; this is regarded as one of the main purpose of authentication. A physical human can be a user or user can also be a malicious web-bot; a malicious program that performs several actions at a cybercriminal's command is regarded as a web-bot. Automated tasks over the internet is run by the help of the software applications and also consistently utilised where the emulation of human activity is needed [4].

2. TYPES OF CAPTCHA

2.1. Text Based Captcha

Text based is considered as the most frequently used CAPTCHAs, where distorted text is replicated. The distorted characters must be identified to solve the CAPTCHA, and also correctly enter them in a designated space. Despite of the fact that Text-based CAPTCHAs are easily generate, they are susceptible to optical character recognition (OCR) attacks [4][5].

Carnegie Mellon University in 2000 [6] has developed an early research-based CAPTCHA called GIMPY. 7 English words are randomly selected inside this CAPTCHA and then they have their character outlines distorted. In overlaid pairs on a colourful noisy background, the text is presented. Beside this fact that the variants exist that just use one word or string of characters, users are asked to type a certain number of these words [7][8][9]. Example of a GIMPY CAPTCHA used as an extra layer of security is shown in Figure 1.

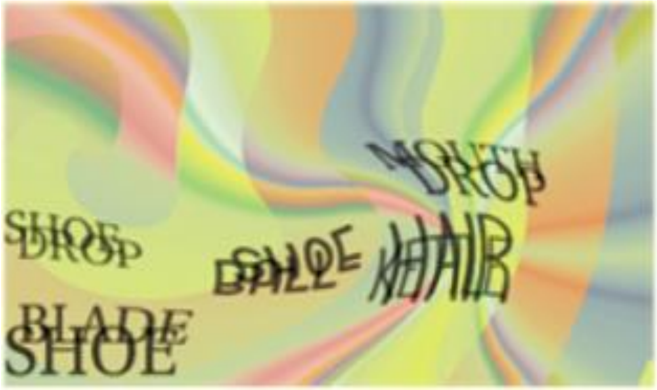


Figure. 1 Example of a GIMPY CAPTCHA (von Ahn (2005)).

2.2. Image based Captcha

On image classification, the existing image-based CAPTCHAs commonly rely, where users are accessible with a series of images and questioned to recognize the bond among them. While on the other hand the ESP-PIX are one such CAPTCHAs and it replicate 4 images & ask users to choose a general description from a drop-down list (Carnegie Mellon University, 2004). Hence, the image groups are chosen from a permanent record, and beside this there is an elevated possibility of random guessing acquiescent an accurate respond.

From Petfinder.com [10], the Asirra image-based CAPTCHA uses a closed database of animals. All images of cats are selected by the users from a diverse set of 12 cats and dogs, which are strained from a big source database of more than 3 million images. Asirra is vulnerable to attack by a classifier trained to differentiate between cats and dogs with 82.7% accuracy [11], beside this fact that the random selection only has a 0.02% chance of correctly selecting the cats (Microsoft, 2010).

2.3. Video Based Captcha

In spite of static images or text, these CAPTCHAs use videos. Users are shown YouTube videos in one such CAPTCHA and also been asked to tag them with descriptive keywords. Despite of the fact that the computer attack rates were about 13%, humans achieved 90% accuracy in tests [12].

2.4. Audio Based Captcha

Audio-based CAPTCHAs were developed by some websites for visually-impaired users. Generally, they work by playing a taping of a set of words or characters with users and being urged to type-in what they hear. Regrettably, by using speech recognition software, these CAPTCHAs are subject to attacks [13][14][15]. Approx 71% was the estimation of the attack rate on audio CAPTCHAs, which is used by Google and Digg (Tam et al., 2008).

3. RESEARCH OBJECTIVES

- To study and analyze the various techniques for automatic captcha generator.
- To study the various issues and challenges related to the CAPTCHA generator and recognition.
- To present a novel technique for the automatic CAPTCHA generation to differentiate between robots/system and human.
- To implement the proposed methodology for validating the outcomes and comparison study with some latest techniques of CAPTCHA generation.
- To consider the proposed technique for reCAPTCHA generation and presenting the same as for present and future investigations.

4. MOTIVATION

There is rise in security problems like, the attack on online polling, attack on E-ticketing, dictionary attacks and email spam amongst others have are on the increase because of the prevalence of web-bot; (Ahnet al., 2003; Singh and Pal, 2014). To apprehend or mitigate the problem of web-bots, CAPTCHAs are systems that have been developed. Moreover, due to advances in Artificial Intelligence (AI) methods and applications, criminals continue to evolve innovative methods of outsmarting existing CAPTCHA systems. It becomes more difficult to distinguish a bot from a human user as machines become more intelligent. Therefore, there are various CAPTCHA systems, which have futile or deemed to have high rate of false positives. By using real-time face detection and trivial gestures as the Turing tests, we will implement an improved CAPTCHA system in this study.

5. LITERATURE REVIEW

Rafaqat [16] is there work have described a novel technique for recognition and segmentation where the easy images and related processing techniques like thinning, pixel count, threshold techniques along with the use of the ANN for the text related CAPTCHs. In general, the system got an wholesome accuracy of 51.3%, 27.1% and 53.2% for Taobao, MSN and eBay datasets along 1000, 500 and 1000 CAPTCHAs correspondingly. This research advantages are dual: through realizing text-based CAPTCHAs, Author did not solely survey the disadvantages within the present design however also consider a method to recognize and segment the attached string bit considered of images.

So as to block the submission of the URLs automatically using CAPTCHA the first website was Alta. At CMU, on HIPs advanced efforts have been made [17]. Authors have been introduced the concept of CAPTCHA and declared that it is the compulsory properties. Numerous CAPTCHA processes (e.g. Bongo, Pix, Gimpy) are present to users on respective website. More than three years, PARC and UC Berkeley considered a new provocations [18], in this aspect another type of authentication is Mandatory Human Participation (MHP) that utilizes a character-morphing algorithm for producing the character identification riddles. In current scenario entire CAPTCHAs are within commercial use to obtain the benefit of advanced human capability in understanding machine written wording. Another technique used facial features, graphical tests, speech [19][20]. With our depth and keen knowledge, this paper demonstrates primary attempt made in particular ”Handwritten CAPTCHAs”.

CAPTCHA was distinct since single cryptographic protocol of whom basic stability statement is relied upon an AI issue (Von Ahn et al., 2003). Accordingly CAPTCHA in reality compose single AI-based encryption system to encode the data which results either CAPTCHA is neither reduced down and there is a method to discriminate humans against a computer systems, either the CAPTCHA is broken through computer systems and a purposeful AI issue is solved(Von Ahn et al., 2003).

As far as the market concerns are being considered the CAPTCHA are considered to make difference between the computer and human tasks and understanding the things which are quite impossible for the system to perform. As per Chellapilla and Simard machine learning can be easily broke the pure recognition tasks and the complex form of HIPs can be generated from a combined processing of segmentation and recognition process. As per them the most effective way to confuse machine learning is building segmentation tasks.

Many of the character segmentation techniques were developed for make harmful to text CAPTCHAs [21]. Gabor filters propose to solve CAPTCHA using image signal processing with localization of spatial and frequency information [22]. Log-Gabor filters were used CAPTCHA images along four directions to extract character components while k-Nearest Neighbours were used for recognition (H. Gao, et al., 2016). (Bursztein et al., 2014) tried for detecting entire possible slice ends to partition a CAPTCHA into particular characters (Bursztein et al., 2014). The effective slices were being retrieved through checking second derivative considering graphs. Based upon cut points, the understood potential portions may be taken out. The ensemble learning concept was adhered to vote realization scores, as is stronger with of noise (Bursztein et al., 2014). Chen et al. categorized the CAPTCHA segmentation procedures into various classifications likewise structure, width, filter, projection, contour, connection and centered the string bit recognition procedures based upon neural network ,deep learning techniques [23].

For the purpose of segmentation provocations, the present CAPTCHA solution procedures are majorly relied upon known-plaintext attack where count of educating samples is confined. Considering the work by, Chen et al. collected 1000 CAPTCHA dataset which contains 4000 characters in total, confined through the prevailing of CAPTCHAs [24], in which selective learning confusion class (SLCC) launched a complicated confusion class for enhancing character identification. This kind of known-plaintext attack has confined capacity within training understanding CNN models.

Because of complexity of possessing a big amount of tagged CAPTCHA data sets, Stark et al. proposed an working deep learning model which adheres only few training data set extracted from Cool PHP CAPTCHA architecture for their CNN framework without any human being interference [25]. They elaborated procedures of localization and segmentation steps which permits for CAPTCHA to be used and cracked for CNN training (F. Stark et al., 2015). These segmentation techniques tried in [26] were able to not solely crack elementary CAPTCHAs however also display the vulnerability inside various another corporate level CAPTCHAs. Further, CAPTCHA provocations created through Google are often reutilized and may be much vulnerable to being broken (S. Sivakorn, 2016).

Bostik et al. [27] attempted various machine learning procedures for testing a CAPTCHA of 4950 synthetic string bits created through a PHP generator (O. Bostik, 2018). Given experimental inferences displayed which a feed-forward network possess better production than k-NN, SVM, and another ML procedures. The same views are considered through various research that are been validated workability of neural network into breaking CAPTCHAs [28]. For instance, two primary stages, recognition and localization, were merged into their procedure within literature (M. Kopp, 2017). The former phase uses a k-means and heat map technique to demonstrate if a string bit is situated at middle with help of ANN. The recognition phase generates a CNN within results have evident it as workable technique for identifying the characters. The procedure also accepted BotDetect CAPTCHA, a up-to-date and paid service utilized through various government companies and institutions whole around the globe, for testing and data training.

Similarly, Jaderberg et al. launched a procedure for text spotting from whole image (M. Jaderberg, 2014) [29]. It calculated a text saliency map through four state-of-the-art CNN categorisers within a sliding window in which 16 various scales were iterated for targeting text heights. Character case-sensitive , Insensitive classification ,Text detection, bigram classification categorization were made through changed CNN networks(M. Jaderberg, 2014). CNN was launched for solving the document identification issue (Y. Leven, 1998) [30] and the current variants have been extensively utilized for suitable image categorization (Y. Jia, 2014). Other models also could have been utilized for labelling CAPTCHA pictures, likewise Caffee, a deep learning architecture that operates returns and images sets of labels being one of major expected (Y. Jia, 2014) [31].

6. RESEARCH METHODOLOGY

6.1. System Overview

During the design stage the main principles which play an important role for providing a more robust CAPTCHA. In our proposed scheme, multiple secure features which are extremely effective to obfuscate challenges for the breaking attack but easy to solve by users have been applied. In the proposed methodology the major focus is for the generation of the captcha which actually detects the threats easily like machine login and other similar tasks. The proposed system is designed in two different phases:

Phase I: Captcha Generation- For this step two different input variables are considered as image (Ii) and text defining the image and also the reshuffled text (Tij) of the correct one by shifting the position of the words in the considered string. In the complete generation process two different databases are maintained as of integrated images and second is of the real text embedded over image. For the single image considered number of captcha's can be generated as under:

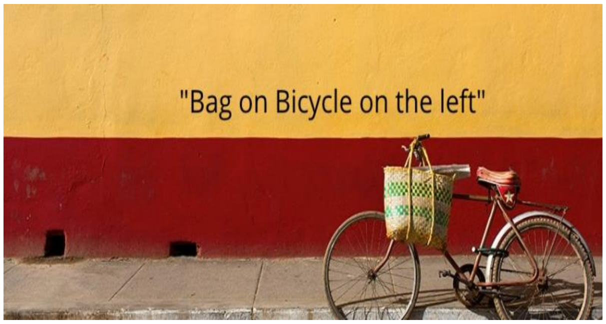


Figure. 2 Image taken as sample.

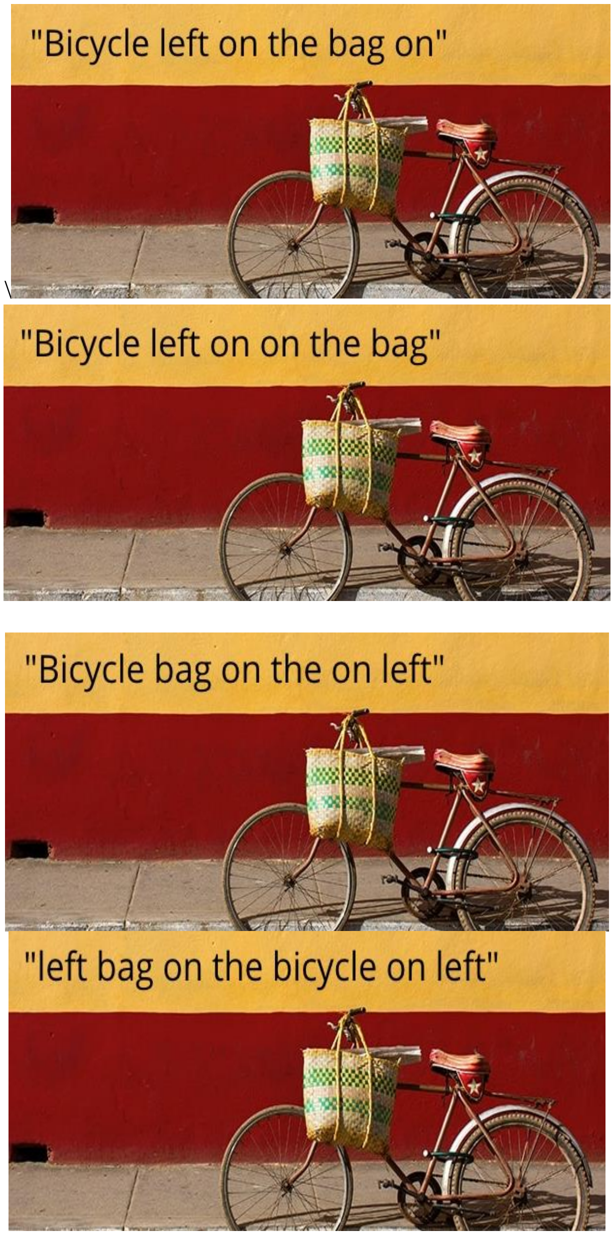

"Bicycle left on the bag on"
"Bicycle left on on the bag"
"Bicycle bag on the on left"
"left bag on the bicycle on left"

Figure 3. Further image generated after shuffling the text.

From the above images image 2 is the real and rest are the fake images for which two different databases are considered which will help in reducing the number of iterations when checking the correct captcha.

Phase II: Captcha Validation- Once the captcha database is generated the captcha is randomly selected from the dataset created and is checked for efficiency after selection is made from user's end. The text embedded over the image is retrieved from the image is checked for accuracy for which n-gram (unigram, bigram and trigram) model is used. The extracted text is checked for matching from the text database and if the text retrieved from the selected image from user's end matches then access is granted and if not matching then next captcha is presented for user for selection and same process is repeated further.

6.2. Mathematical Model

System S,considered the collection of integrated images as input which comprises of text over image and are generated using the steps discussed in phase I. In the complete system two different datasets are managed as common dataset (Dc) which comprises of all generated embedded images (EI) and another dataset (Dt) is the collection of the original text (Ti).

Generation:$EI_i= \{I_i, T_i,T_j,\}$ Where

I_i: Input Images,
T_i: Original Text,
T_j: Blend of text,
EI: Embedded images.

Validation: For all EI_i text $T_i$is extracted and checked for accuracy after selection by user and is validated using the n-gram model (unigram, bigram and trigram).

The first step in phase II is all about the extraction of the text from the image which then pre-processed for stop word removal, stemming, etc. In the proposed work ORC technique is being for the extraction of the text from the image as discussed in Chidiac, Damien, and Yaacoub (2016) [32].

The other element is the "Entity Extractor", which simply aims at removing the structure out of text. Therefore, following are some steps through which the meaning of Entity Extractor can be easily understand. In the stating, the text was removed from the image in order to get the number of entities involved. Further, each and every organization follows the path of disinfect text and results in following:

Segmentation of characters others than alphabets,

Avoiding the use of similar terms,

Review either the correct English words are used or not,

Monitoring the spelling errors.

In the case when the n-gram model is being counted it checks the probability of the words coming next to considered word on the basis of the training dataset available and classifies the text as fake or real. In the case when the probability of the very word is higher as per the training dataset then the text embedded is real else fake. N-gram model goes with unigram, bigram, trigram, in the case of the unigram considering "Dhoni is a robot" as the extracted text then for unigram probability of 'is' is checked for 'dhoni' and similarly probability of 'a' is checked for 'is' and so on, now if the probability of occurrence of words is higher as per the training dataset then the text retrieved is real else fake.

Considering the working example of the n-gram model working,

S (News): I I am not,
Training dataset:
S1: I am a human,
S2: I am not a stone,
S3: I I live in delhi,

- Model applied is unigram

P(S) = P(I I am not) = P(I / <S>) * P(I / <S>) * P(am / <S>) * P(not / <S>)

= 3/3*3/3*2/3*1/3

=1*1*.6667*.3333

=0.22219, is the probability of the occurrence of these defined words as combination.

Model applied is bigram

P(S) = P(I I am not) = P(I / <S>) * P(I / I) * P(am / I) * P(not / am)

= 3/3 * ¼ * 2/4 * ½

=0.0625, is the probability of occurrence of these words as combination using the bigram model.

Model applied is bigram

P(S) = P(I I am not) = P(I I / <S>) * P(am / I I) * P(not / I am)

= 1/3 * 1/1 * ½

= 0.1666, is the probability of occurrence of these words as combination using trigram.

On the basis of the calculation made above it is quite clear that "I I am not" is a false selection which actually is being calculated on the basis of the training dataset available. In the all of the applied n-gram models the highest probability showing the together occurrence of the words is 0.222 and is even less then half which shows that the text represented is false.

7. RESULT AND DISCUSSION

In the research methodology defined the dataset is being created for the images embedded with text over them and also for single image several other images are generated by shuffling the text associated over image. In the normal cases to ensure that the machine is not auto answering the website request some queries are generated online for security of the website and data generated online. In normal cases images, mathematical queries and some integrated segments are used to ensure that the user is human to avoid multiple visits and also to avoid the unwanted traffic over the network. In the work presented the efficiency of the technique is measured and compared using certain parameters as loading time, response time, success rate, cloud interface, security, precision, recall and accuracy. In the very initial stage the dataset is being created using the images and text embedding as shown below. In the very initial phase of testing 100 different images are considered and from that 100 images and text associated with them other 1000 embedded images are generated by shuffling the text associated with images.

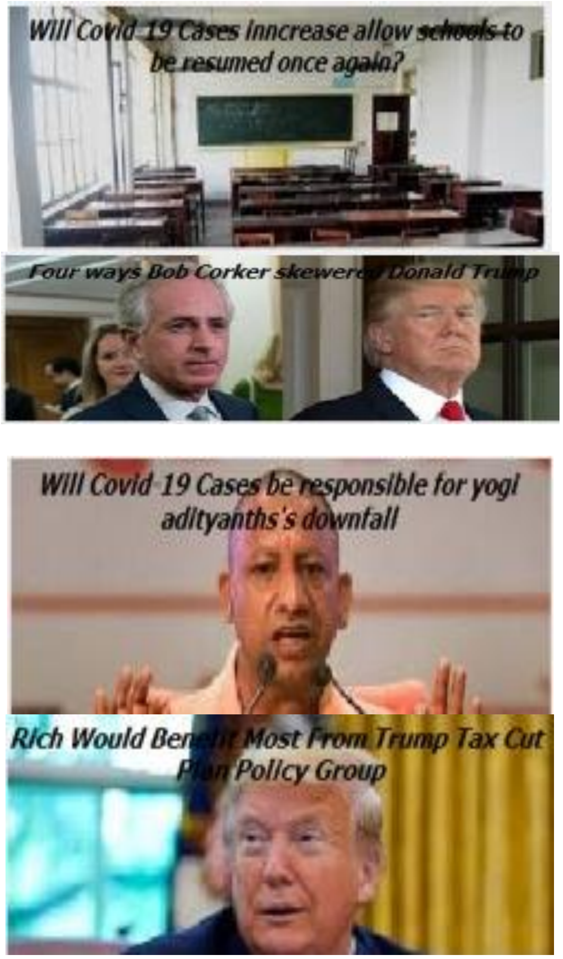
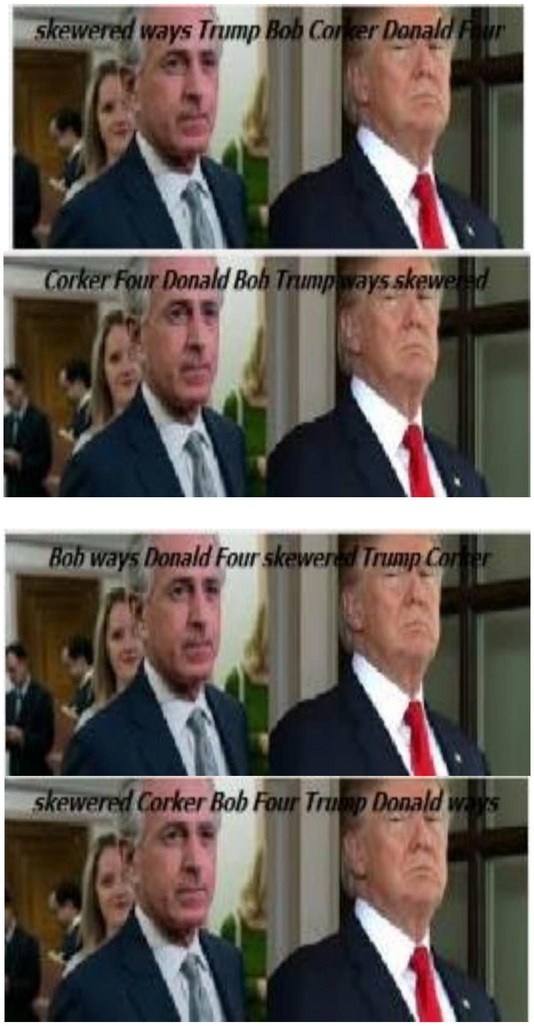

Will Covid 19 Cases Inncrease allow schools to be resumed once again?
Four ways Bob Corker skewered Donald Trump
Will Covid 19 Cases be responsible for yogi adityanths's downfall
Rich Would Benefit Most From Trump Tax Cut Plan Policy Group

Figure 4. Example of the images embedded with text for which a dataset is created.

As shown in the figure above 4 different images are considered and also all of the images are having the text embedded over them, “will covid19 cases increase allow schools to be resumed over again”, “four ways bob corker skewered Donald Trump”, “will covid19 cases be responsible for yogi adityanath’s downfall” and “Rich would benefit most from trump tax cut plan policy group”. From all of the available text different embedded images can be further generated by shuffling the text words as under:

1. “will covid19 cases increase allow schools to be resumed over again” (original text),
2. “will increase resumed over again schools to be covid19 cases allow”,
3. “covid19 school resumed again over cases allow increase to be will”,
4. “schools over again covid19 resumed increases cases will to be allow”, and so on.

For the other texts available similar other texts can be further generated and are embedded over the images to create multiple copies of images, the dataset considers about 6 different shuffling of the text over the image, a single example of the image generation is shown below:


skewered ways Trump Bob Corker Donald Four
Corker Four Donald Bob Trump ways skewered
Bob ways Donald Four skewered Trump Corker
skewered Corker Bob Four Trump Donald ways

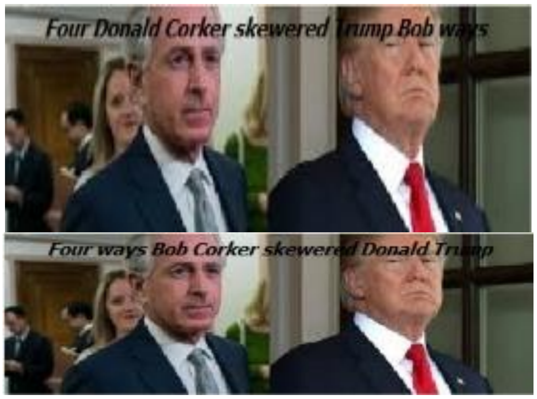

Four Donald Corker skewered Trump Bob ways
Four ways Bob Corker skewered Donald Trump

Figure 5. Multiple images generation for verification.

Below are screenshots considered for methodology defined as the text after retrieval from the image is checked for the probability of occurrence of words next to one another for which the results are as under:

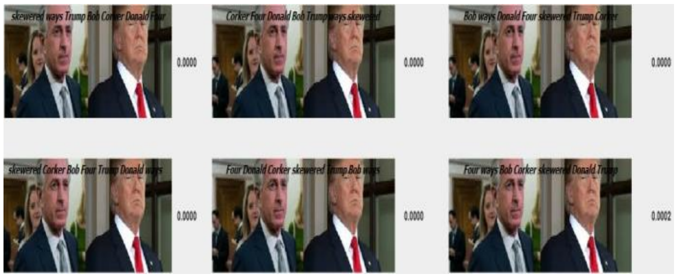

skewered ways Trump Bob Corker Donald Four
0.0000
Corker Four Donald Bob Trump ways skewered
0.0000
Bob ways Donald Four skewered Trump Corker
0.0000
skewered Corker Bob Four Trump Donald ways
0.0000
Four Donald Corker skewered Trump Bob ways
0.0000
Four ways Bob Corker skewered Donald Trump
0.0002

Figure 6. shuffled group of images 1 generated and verification on the basis of the probability of the text occurrence.

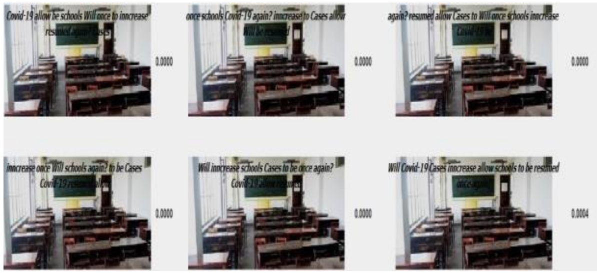

Covid-19 allow be schools Will once to increase resumed again? Cases
0.0000
once schools Covid-19 again? increase to Cases allow Will be resumed
0.0000
again? resumed allow Cases to Will once schools increase Covid-19 be
0.0000
increase once Will schools again? to be Cases Covid-19 resumed allow
0.0000
Will increase schools Cases to be once again? Covid-19 allow resumed
0.0000
Will Covid-19 Cases increase allow schools to be resumed once again?
0.0004

Figure 7. shuffled group of images 2 generated and verification on the basis of the probability of the text occurrence.

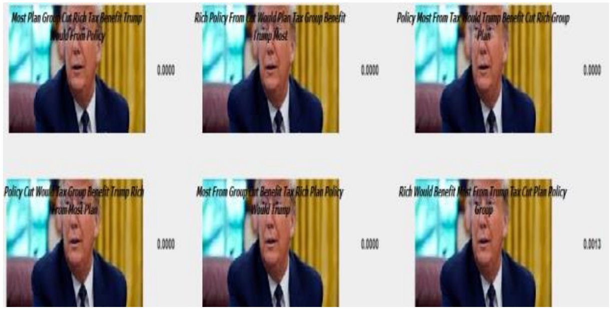

Most Plan Group Cut Rich Tax Benefit Trump Would From Policy
0.0000
Rich Policy From Cut Would Plan Tax Group Benefit Trump Most
0.0000
Policy Most From Tax Would Trump Benefit Cut Rich Group Plan
0.0000
Policy Cut Would Tax Group Benefit Trump Rich From Most Plan
0.0000
Most From Group Cut Benefit Tax Rich Plan Policy Would Trump
0.0000
Rich Would Benefit Most From Trump Tax Cut Plan Policy Group
0.0013

Figure 8. shuffled group of images 3 generated and verification on the basis of the probability of the text occurrence.

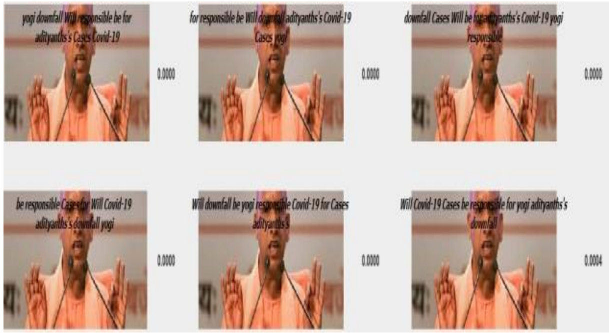
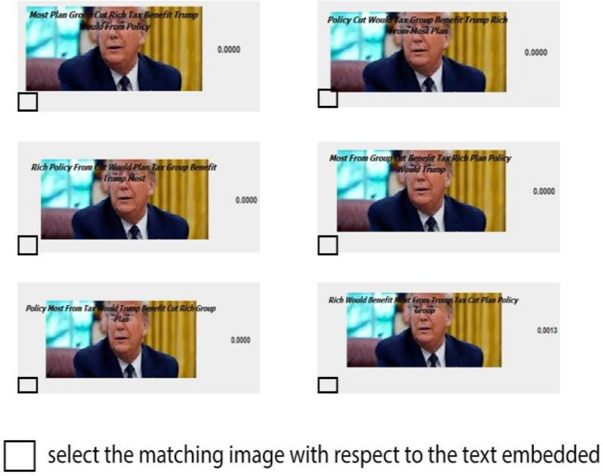

yogi downfall Will responsible be for adityanths's Cases Covid-19
0.0000
for responsible be Will downfall adityanths's Covid-19 Cases yogi
0.0000
downfall Cases Will be for adityanths's Covid-19 yogi responsible
0.0000
be responsible Cases for Will Covid-19 adityanths's downfall yogi
0.0000
Will downfall be yogi responsible Covid-19 for Cases adityanths's
0.0000
Will Covid-19 Cases be responsible for yogi adityanths's downfall
0.0004

Figure 9. shuffled group of images 4 generated and verification on the basis of the probability of the text occurrence.


Most Plan Group Cut Rich Tax Benefit Trump Would From Policy
0.0000
☐
Policy Cut Would Tax Group Benefit Trump Rich From Most Plan
0.0000
☐
Rich Policy From Cut Would Plan Tax Group Benefit Trump Most
0.0000
☐
Most From Group Cut Benefit Tax Rich Plan Policy Would Trump
0.0000
☐
Policy Most From Tax Would Trump Benefit Cut Rich Group Plan
0.0000
☐
Rich Would Benefit Most From Trump Tax Cut Plan Policy Group
0.0013
☐

☐ select the matching image with respect to the text embedded

Figure 10. User interface available for selection to access the web application.

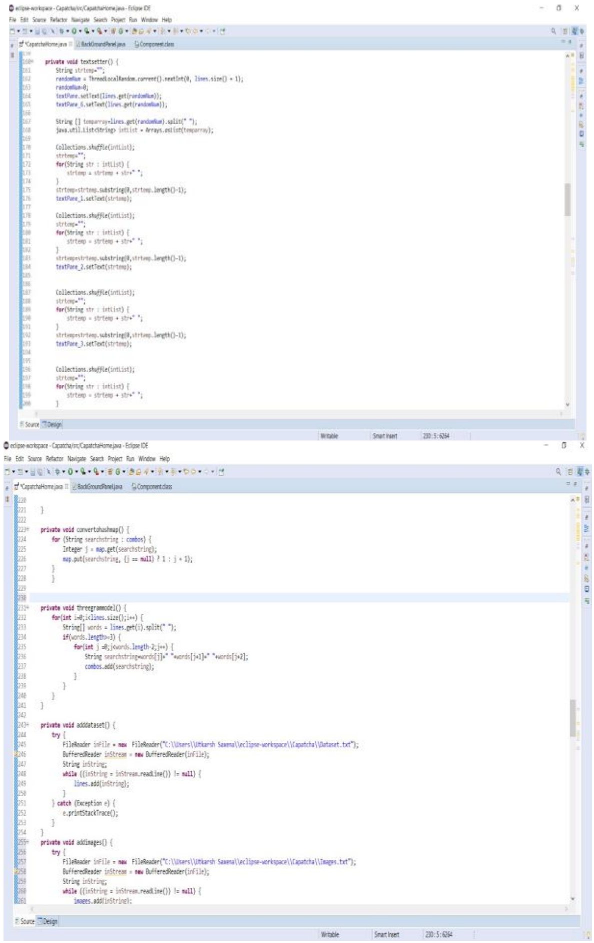
The complete image set is considered as image and text for which T_{ij} and I_{ij} are used for notation, where i defines the image number and j defines the image and text inside image/text.

Table 1. Showing the probability of occurrence of words shuffled to create further images for selection.

Images/Text from individual set	Corresponding probability of occurrence
T_{11}	0.0000
T_{12}	0.0000
T_{13}	0.0000
T_{14}	0.0000
T_{15}	0.0000
T_{16}	0.0002
T_{21}	0.0000
T_{23}	0.0000
T_{23}	0.0000
T_{24}	0.0000
T_{25}	0.0000
T_{26}	0.0004
T_{31}	0.0000
T_{32}	0.0000
T_{33}	0.0000
T_{34}	0.0000
T_{35}	0.0000

T_{36}	0.0013
T_{41}	0.0000
T_{42}	0.0000
T_{43}	0.0000
T_{44}	0.0000
T_{45}	0.0000
T_{46}	0.0004

The table above shows the probability of occurrence of words together for n-gram model is being used which defines the probability of occurrence of words in the string on the basis of the dataset considered, in the methodology uni-gram, bi-gram and tri-gram model is used to compute the probability. In the table the probability of occurrence of last image of every image set is having non-zero digit as probability which denotes that the last one is correct image with respect to the text embedded. The probability of the words is zero for all other because of the size of the dataset, by the time dataset size increases the methodology will start showing scattered or varied probabilities. The snap below shows the computational step used for generation of the probabilities using the n-gram model:


eclipse-workspace - Capatcha/src/CapatchaHome.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
*CapatchaHome.java
BackGroundPanel.java
Component.class
Source
Design
Writable
Smart Insert
230 : 5 : 6264

```
private void textsetter() {
    String strtemp="";
    randomNum = ThreadLocalRandom.current().nextInt(0, lines.size() + 1);
    randomNum=0;
    textPane.setText(lines.get(randomNum));
    textPane_6.setText(lines.get(randomNum));

    String [] temparray=lines.get(randomNum).split(" ");
    java.util.List<String> intlist = Arrays.asList(temparray);

    Collections.shuffle(intList);
    strtemp="";
    for(String str : intList) {
        strtemp = strtemp + str+" ";
    }
    strtemp=strtemp.substring(0,strtemp.length()-1);
    textPane_1.setText(strtemp);

    Collections.shuffle(intList);
    strtemp="";
    for(String str : intlist) {
        strtemp = strtemp + str+" ";
    }
    strtemp=strtemp.substring(0,strtemp.length()-1);
    textPane_2.setText(strtemp);


    Collections.shuffle(intList);
    strtemp="";
    for(String str : intlist) {
        strtemp = strtemp + str+" ";
    }
    strtemp=strtemp.substring(0,strtemp.length()-1);
    textPane_3.setText(strtemp);


    Collections.shuffle(intList);
    strtemp="";
    for(String str : intlist) {
        strtemp = strtemp + str+" ";
    }
```

```

}

private void convertohashmap() {
    for (String searchstring : combos) {
        Integer j = map.get(searchstring);
        map.put(searchstring, (j == null) ? 1 : j + 1);
    }
    }


private void threegrammodel() {
    for(int i=0;i<lines.size();i++) {
        String[] words = lines.get(i).split(" ");
        if(words.length>=3) {
            for(int j =0;j<words.length-2;j++) {
                String searchstring=words[j]+" "+words[j+1]+" "+words[j+2];
                combos.add(searchstring);
            }
        }
    }
}

private void adddataset() {
    try {
        FileReader inFile = new  FileReader("C:\\Users\\Utkarsh Saxena\\eclipse-workspace\\Capatcha\\Dataset.txt");
        BufferedReader inStream = new BufferedReader(inFile);
        String inString;
        while ((inString = inStream.readLine()) != null) {
            lines.add(inString);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
private void addimages() {
    try {
        FileReader inFile = new  FileReader("C:\\Users\\Utkarsh Saxena\\eclipse-workspace\\Capatcha\\Images.txt");
        BufferedReader inStream = new BufferedReader(inFile);
        String inString;
        while ((inString = inStream.readLine()) != null) {
            images.add(inString);
```

Figure 11. Computational steps counted for probability generation.

For the comparison of the methodology described several parameters are considered as precision, recall, accuracy, loading time, response time, success rate, etc.

Table 2. Comparison results based on discussed parameters.

Comparison Parameters	Text Captcha	Image Captcha	Face Captcha	Proposed Methodology
Loading time (sec)	2	3	3.5	3
Response time (sec)	5	6	4	4.5
Success Rate	90%	90%	93%	97%
Security	No	No	No	Yes
Precision	87	85	87	95
Recall	80	80	87	93

The results have described that the proposed methodology outperforms the other methodology used for the security of the web on the basis of the several parameters considered. As the results are on the basis of the intermediate computation hence it is difficult for the machine to detect the technique used for the generation of the captcha and also to identify the correct image.

8. CONCLUSION

CAPTCHAs are an effective way to counter bots and reduce spam. The web being the primary means of transmission of information needs to be protected from malicious web-bots which pose a hindrance to web services hence the need for a public Turing test to tell computers and bots apart was designed. The embedded image and text captcha system is able to generate a platform for authentication for the user over the web based applications which enhances the security level. The generated system best works to reduce the breaking of the generated captcha and at the same time the user interface generated is quite user friendly and easy to use at user's end. The technique presented in the work is the combination of the text and images for which several components are generated by shuffling the text embedded over the image by the means of the word replacement. In the GUI or at the user end the system asks for the selection of the image that matches the text overwritten and after selection the text is extracted from the image is then checked for probability of occurrence of the words together using the n-gram model and if for all images provided the probability is higher for the selection made then the access is granted and if not then re-selection is provided, the same is repeated for three times and if the authentication fails then the system stops responding for the query made by the user. The system is tested for several images and associated text over the image for which the results are shown in the tables and images attached in the paper. The comparison results shows that the system is prone to attacks for un-authorized access to the web content. In the future the work can be further processed by automating the process of image generation with respect to the text shuffling. As the major limitation of the work is about the computational complexity and response just because of the integration of the text and images and also the work can be further compared on the basis of the image quality provided for several associated parameters.

REFERENCES

1. Datta, R., J. Li, and J.Z. Wang. **IMAGINATION: a robust image-based CAPTCHA generation system. in Proceedings of the 13th annual ACM international conference on Multimedia**. 2005. ACM.
2. Marsico, M. D., Marchionni, L., Novelli, A., Oertel, M. (2015). **FATCHA: Biometrics lends tools for CAPTCHAs. Multimedia Tools Appl Multimedia Tools and Applications.**
3. Deepa, G., &Thilagam, P. S. (2016). **Securing web applications from injection and logic vulnerabilities: Approaches and challenges**. Information and Software Technology, vol. 74, pp. 160-180.
4. Chellapilla, K., Larson, K., Simard, P.Y. and Czerwinski, M. (2005) '**Building segmentation based human-friendly human interaction proofs (HIPs)**', *Human Interactive Proofs*, pp.1–26, Springer, Berlin.
5. Kluever, K.A. (2008) **'Evaluating the usability and security of a video CAPTCHA'**, Rochester Institute of Technology.
6. Baird, H.S. and Popat, K. (2002) '**Human interactive proofs and document image analysis**', *Document Analysis Systems V*, pp.531–537, Springer, Berlin.
7. von Ahn, L., Blum, M. and Langford, J. (2004) **'Telling humans and computers apart automatically',** *Communications of the ACM*, Vol. 47, No. 2, pp.56–60.
8. Mori, G. and Malik, J. (2003) '**Recognizing objects in adversarial clutter: breaking a visual CAPTCHA',** *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Madison, Wisconsin, June.
9. Moy, G., Jones, N., Harkless, C. and Potter, R. (2004) '**Distortion estimation techniques in solving visual CAPTCHAs**', *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Washington, DC, June.
10. Elson, J., Douceur, J., Howell, J. and Saul, J. (2007) '**Asirra: a CAPTCHA that exploits interest-aligned manual image categorization**', *14th ACM Conference on Computer and Communications Security*, Alexandria, Virginia, October.
11. Golle, P. (2008) '**Machine learning attacks against the Asirra CAPTCHA',** New York, NY.
12. Kluever, K.A. and Zanibbi, R. (2009) '**Balancing usability and security in a video CAPTCHA**', *5th Symposium on Usable Privacy and Security*, Mountain View, California, July.
13. Bursztein, E. and Bethard, S. (2009) '**Decaptcha: breaking 75% of eBay audio CAPTCHAs**', *3rd USENIX Workshop on Offensive Technologies*, Montreal, August.
14. Santamarta, R. (2008) '**Breaking Gmail's audio Captcha'**, available at http://blog.wintercore.com/?m=200803 (accessed on 31 August 2009).
15. Tam, J., Hyde, S., Simsa, J. and von Ahn, L. (2008) '**Breaking audio CAPTCHAs'**, *22nd Annual Conference on Neural Information Processing Systems*, Vancouver, British Columbia, December.
16. Rafaqat Hussain & Hui Gao & Riaz Ahmed Shaikh, "**Segmentation of connected characters in text-based CAPTCHAs for intelligent character recognition**", Springer Science+Business Media New York 2016.
17. M. Blum, L. von Ahn, J. Langford, and N. Hopper. **The captcha project: Completely automatic public turing test to tell computers and humans apart**. *http://www.captcha.net*, November 2000.
18. J. Xu, R. Lipton, I. Essa, M. Sung, and Y. Zhu. Mandatory **human participation: A new authentication scheme for building secure systems.** *ICCCN*, 2003.
19. G. Kochanski, D. Lopresti, and C. Shih. **A reverse turing test using speech. *Proc.*** *International Conference on Spoken Language Processing*, September 2002.
20. Y. Rui and Z. Liu. **Artifacial: Automated reverse turing test using facial features. *Proc***. *The 11th ACM international con- ference on Multimedia*, November 2003.
21. Yan, J.; El Ahmad, A.S. **A Low-cost Attack on a Microsoft CAPTCHA. In Proceedings of the 15th ACM Conference on Computer and Communications Security**, Alexandria, VA, USA, 27–31 October 2008; pp. 543–554.
22. Gao, H.; Yan, J.; Cao, F.; Zhang, Z.; Lei, L.; Tang, M.; Zhang, P.; Zhou, X.; Wang, X.; Li, J. A **Simple Generic Attack on Text Captchas. In Proceedings of the Network and Distributed System Security** Symposium 2016, San Diego, CA, USA, 21–24 February 2016.
23. Chen, J.; Luo, X.; Guo, Y.; Zhang, Y.; Gong, D. A **Survey on Breaking Technique of Text-Based CAPTCHA. *Secur***. *Commun. Netw.* 2017, *2017*, 6898617.
24. Chen, J.; Luo, X.; Liu, Y.; Wang, J.; Ma, Y. **Selective Learning Confusion Class for Text-Based CAPTCHA Recognition**. *IEEE Access* 2019, *7*, 22246–22259.
25. Stark,F.;Hazırbas¸,C.;Triebel,R.;Cremers,D. **CAPTCHA Recognitionwith Active Deep Learning.** In Proceedings of the German Conference on Pattern Recognition Workshop, 2015.
26. Sivakorn, S.; Polakis, I.; Keromytis, A.D. **I am Robot: (Deep) Learning to Break Semantic Image CAPTCHAs. In Proceedings** of the 2016 IEEE European Symposium on Security and Privacy (EuroS P), Saarbrucken, Germany, 21–24 March 2016; pp. 388–403.
27. Bostik, O.; Klecka, J. Recognition of **CAPTCHA Characters by Supervised Machine Learning** Algorithms. *IFAC-PapersOnLine*2018, *51*, 208–213.

28. Kopp, M.; Nikl, M.; Holeña, M. Breaking **CAPTCHAs with Convolutional Neural Networks**. *CEUR Workshop Proc.* 2017, *1885*, 93–99.
29. Jaderberg, M.; Vedaldi, A.; Zisserman, **A. Deep Features for Text Spotting. In Proceedings of the European Conference on Computer Vision**, Zurich, Switzerland, 6–12 September 2014.
30. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. **Gradient-based learning applied to document recognition.** ***Proc.*** *IEEE* 1998, *86*, 2278–2324.
31. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.B.; Guadarrama, S.; Darrell, T. Caffe: **Convolutional Architecture for Fast Feature Embedding. In Proceedings** of the 22nd ACM international conference on Multimedia, Orlando, FL, USA, 3–7 November 2014.
32. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., **Thirion, B., Grisel, O., Dubourg, V., "Scikit-learn: Machine learning in Python",** *The Journal of Machine Learning Research, 2011, vol. 12*, pp. 2825-2830.
33. J. Xu, R. Lipton, I. Essa, M. Sung, and Y. Zhu. Mandatory **human participation: A new authentication scheme for building secure systems**. *ICCCN*, 2003.