Automatic Identification of CAPTCHA Schemes

M.A. Asim K. Jalwana, Muhammad Murtaza Khan, and Muhammad U. Ilyas

National University of Sciences and Technology (NUST), Islamabad, Pakistan
{11mseemjalwana,muhammad.murtaza,usman.ilyas}@seecs.nust.edu.pk

Abstract. Text based CAPTCHAs are ubiquitous on the Internet since they are easily generated by machines, easily solvable by humans and yet not easily defeated by state-of-the-art computer algorithms. Over the years, several attacks have been designed by researchers to solve different types of CAPTCHAs. These attacks always assume that the type of CAPTCHA is known. However, in order to devise a common frame work, comprising of different attacks that can be launched automatically, the first prime step is to recognize the CAPTCHA scheme. In this paper we present a method based on geometric features to automatically identify text based CAPTCHA schemes. The proposed method is verified on a data set comprising of 25 different types of CAPTCHA (1,000 samples per type). We achieve an identification / classification accuracy of up to approximately 99%.

1 Introduction

A Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) is a standard test for differentiating between human users and bot-based attackers while gaining access to online services. The idea was introduced in an unpublished work by Naor [1]. However, the term CAPTCHA was coined in 2000 by Blum and von Ahn [2]. Over the years, hard AI problems have been used to develop different types of CAPTCHAs [2].[1] The properties that make a problem hard to solve, and thus a CAPTCHA more robust to bot-attacks, were discussed in detail by Basso et al. in [3]. A CAPTCHA is considered robust to attacks if the success rate of attacks is less than 0.01% [4–6]. However, it is also desired that the CAPTCHA is usable, *i.e.* the human success rate should be at least 90% [4]. Recently, Ellie *et al.* [6] revised the value of robustness of CAPTCHA to bot-attacks from 0.01% to 1%, citing it as more meaningful.

CAPTCHA schemes deployed on the Internet are often classified into one of three broad categories, *i.e.* text, audio or image based CAPTCHAs [3, 7]. Text CAPTCHAs are the most widely used type of CAPTCHA, which is why they are the focus of this paper. Text CAPTCHA tests are based on the fact that humans can easily read distorted and / or corrupted text in images, which cannot be solved by state-of-the-art optical character recognition (OCR) softwares [8]. However, researchers have identified different pre-processing steps which

[1] A problem is said to be hard if there is general consensus among community working on it that there are no effective ways to solve it [3].

G. Bebis et al. (Eds.): ISVC 2014, Part II, LNCS 8888, pp. 416–426, 2014.

can remove the effect of distortion or noise from the images thus making the CAPTCHA solvable by OCRs [5, 6, 9–11]. This in turn has led to the evolution and development of more robust CAPTCHAs. Recently, Ellie *et al.* [6] proposed a comprehensive set of schemes for defeating different CAPTCHAs. However, they did not specify how to select a particular attack, from that set to use against a particular CAPTCHA. To the best of our knowledge, no method has been proposed to-date which identifies /classifies a particular text based CAPTCHA scheme to select an attack to maximize the probability of success.

In this paper we propose a method for identification of different CAPTCHA schemes by using geometric features. We explored 31 different features to classify CAPTCHAs and identifed the most significant features among them. We verify our approach using a data set comprising of a total of 25, 000 text CAPTCHAs, 1, 000 samples for each of the 25 different CAPTCHA schemes. For the best case we achieved a CAPTCHA scheme identification accuracy of approximately 99%.

The main contributions of this paper include:

1. Collection of original data set of 25, 000 unique CAPTCHAs consisting of 1, 000 instances of each of the 25 different types (see Section 3).
2. Identification of the most significant features for CAPTCHA scheme identification (see Section 4).
3. Designing of a classifier to detect the type of a CAPTCHA (see Section 4).

2 Proposed Methodology

In this section we describe the methodology adopted to identify the CAPTCHA type of a specific CAPTCHA instance. The proposed methodology is based on a set of 31 candidate geometric features for discriminating among CAPTCHA schemes. These features should be sensitive to inter-class variation and insensitive to intra-class differences. Identification of such features can be a challenging task and requires knowledge of the problem context. In the context of CAPTCHAs, a suitable list of features that can classify the CAPTCHAs is not available publicly. In this paper we evaluate the geometric features listed in Table 1 to classify CAPTCHAs. Next, the ten most significant discriminative features were identified using three different ranking metrics. The rankings of all these features identifies the significance of contribution of features to classification [12]. Finally, five different classifiers are evaluated for their performance to categorize CAPTCHAs into 25 types. This evaluation is done using 10-fold cross validation. These steps are discussed below in detail.

Most text based CAPTCHA schemes add distortion to the text or background of CAPTCHA in the form of: (i) geometric transformation of text (deformed text) (ii) background noise (lines, speckle etc.), and (iii) closest packing (difficulty in identification of alphabets or characters due to contact or occlusion). We propose to use the geometric measures presented in Table 1 as candidate features to classify CAPTCHAs. The features listed in Table 1 are used to quantify different aspects of these distortions, *e.g.* frequency of holes can be used to detect artificial holes generated due to characters that are closely packed together; The greater characters are spaced apart, the smaller the number of holes. It can be observed in CAPTCHA schemes, like Google CAPTCHA, that when characters are placed so close to each other that they touch, false holes may be created. Holes are also generally associated with characters like *a, o etc.* However, a large number is indicative of characters that may be closely packed instead of just presence of characters *a, o etc.*

Similarly, the Euler number represents the difference in number of connected components. The smaller the number of connected components the more characters are touching / overlapping each other thus, providing supplementary confirmation of the of holes measure. The fifth feature in Table 1 named *Steps* is the number of steps required to erode the text in a CAPTCHA image. This is a measure of the thickness of foreground text. The thicker the text the greater the number of times the erosion operation will have to be used to scrub the text. 'Error from line fitting,' the ninth feature, identifies the degree of waviness of the foreground text. 'Orientation,' the 14th feature, helps identify the average direction of text which is different for different schemes.

Table 1. List of candidate features and their descriptions

S/N	**Feature**	**Description**
1	Projection length	Length of projection of foreground on x-axis.
2	Branch points	Number of branch points left after applying skeletonization and pruning algorithm.
3	Branch point density	Ratio of branch points and projection Length.
4	Perimeter (of foreground)	Evaluated by summing Euclidean distance between adjacent boundary pixels.
5	Steps	Number of iterations to erode the image with a 3x3 structuring element.
6	Connected components	Number of connected components in binary image.
7	Major axis length	Length of major axis of an ellipse having the same normalized second central moments as foreground.
8	Eccentricity	Ratio of distance between the foci of foreground and its major axis length.
9	Error from line fitting	Least square error of fitting a line on the lower envelope of foreground text.
10	Frequency of holes	Set of background pixels surrounded by foreground pixels.
11	Euler number	Difference between number of connected components and holes.
12	Compactness	Ratio between $perimeter^2$ and Area.
13	Elongatedness	Ratio between area and $Steps^2$.
14	Orientation	Angle in degrees between the x-axis and major axis of an ellipse (fitting foreground).
15-29	M*ab* Only used up to fourth order (15 values).	Moment of order a on x-axis and order b on y-axis.
30	Compactness by major axis length	Ratio between compactness and major axis length.
31	Euler number by steps	Ratio between Euler number and steps.

Before extracting these features we binarized CAPTCHA images using Otsu's thresholding and resized them to a uniform resolution of 60 × 200 pixels. The purpose of resizing is to remove any bias in the recognition process that may occur due to variations in sizes of CAPTCHAs schemes. This transformation from CAPTCHA image to binary image yields two advantages: (1) Binarization automatically removes noise pixels, *e.g.* in the case of Wyoming Community Bank CAPTCHA, binarization removes the water mark ('Wyoming') from the image. (2) It is computationally more efficient to work on binary images as opposed to color images. To differentiate between background and foreground we assumed that the majority of pixels in the binarized CAPTCHA image belong to the background. So, the most frequently occurring binary level is considered as the background and others as the foreground.

From the set of training images we extracted their features given in Table 1 and 1) ranked the features and 2) trained various classifiers using the most significant ones. After we trained the classifiers, we applied them to the remaining CAPTCHA images that constitute the testing set. We used Matlab for feature computation and WEKA v3.6.9 implementations of all classification algorithms. After trying different classification algorithms we determined that the random forest trees classifier gave us the highest accuracy. The above mentioned steps were repeated for 10-fold cross validation. The implementation details are further discussed in next section.

3 Data Set Acquisition

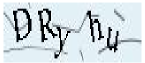
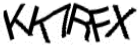
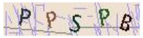
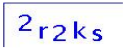
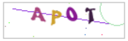
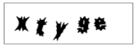
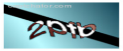
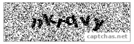
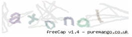
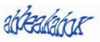
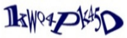
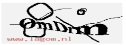
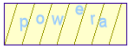
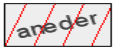
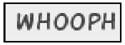
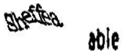
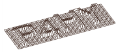
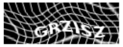
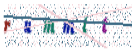
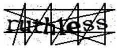
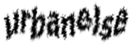
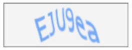
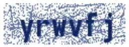
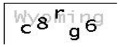
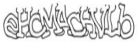
No standardized public database of CAPTCHAs is available, so we developed a crawler (in C#) to crawl and download CAPTCHAs from a number of different websites [13]. The CAPTCHAs were downloaded from websites where they were deployed. The collected data set comprises of a total of 25, 000 CAPTCHA images, with 1, 000 samples each of 25 different classes / types. These classes / types are listed in Table 2.

4 Experimental Results

4.1 Feature Ranking

A total of 31 features were extracted from all 25, 000 CAPTCHA images. Thus, following feature extraction, each CAPTCHA in the data set is represented by a feature vector of length 31. Note that Yousra *et al.* [14] proposed and explored the use of some geometric measures for usability analysis of CAPTCHAs by classifying any CAPTCHA as hard or easy to solve for human users, regardless of its type. We explored a much more comprehensive list of features in this paper than Yousra *et al.* [14], which are then ranked in order of significance to classification. To determine if all of these measures are necessary for correct identification of a CAPTCHA class we employed 3 different feature ranking algorithms. The aim was to identify the most significant or discriminative features: The ranking criteria used are a) information gain, b) χ^2 ranker and c) gain ratio.

Table 2. CAPTCHA types contained in our data set

CAPTCHA Scheme	**Sample Image**	**CAPTCHA Scheme**	**Sample Image**	**CAPTCHA Scheme**	**Sample Image**
Agricultural Bank of China		Amazon		Bank of Communications	
Bank of Poynette		Crytographp		CAPTCHA Creator	
Captchator		Free CAPTCHA Service		Freecap	
Google		Hotmail		HK CAPTCHA	
Protect webform Type 1		Protect webform Type 2		Protect webform Type 3	
ReCAPTCHA		Rediff		Reddit	
Simple Machines Forum		Slashdot		Wikipedia	
Web Spam Protect		White Hat		Wyoming Community Bank	
Yahoo					

The graphs in Figure 1 plot the normalized feature ranking metrics (information gain, χ^2 ranker, and gain ratio) against each feature. The top 10 features were selected based on their median rank index by a) information gain, b) χ^2 ranker and c) gain ratio. In the graph in Figure 1 the horizontal axis lists features and the vertical axis represents their contribution as measured by each of the three ranking metrics. Thus, if all ranking metrics were in agreement, all three curves should decrease monotonically. Any increase in the curve represents that a different ranking would have been obtained if each curve would have been plotted separately. The fact that there are only a couple of incidences in which the normalized value of *information gain* or χ^2 increases indicates that there is little variation in the ranking of features.

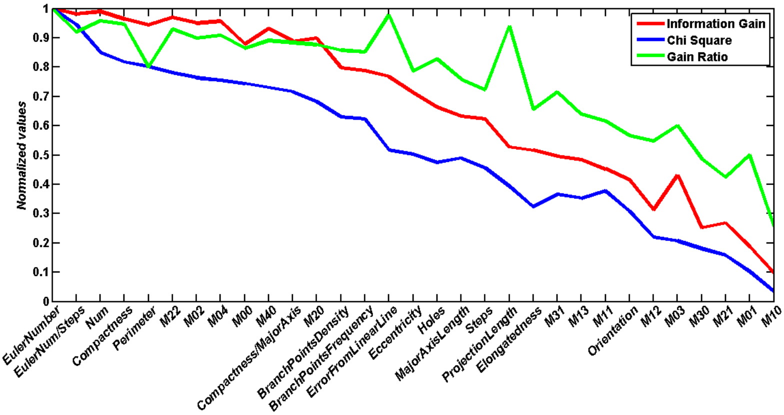


Fig. 1. Features ranked by information gain, gain ratio and χ^2 ranker

4.2 Classifier Design

For comparative analysis we used the features identified by Yousra *et al.* [14] for classification and demonstrate that our feature set outperforms theirs. The results are summarized in Table 3.

When we use all the 31 features presented in Table 1 together with a random forest classifier, we achieve an average accuracy of around 99.4% over 10-folds. This work focuses on identification of different CAPTCHAs, *i.e.* classification of CAPTCHAs in 25 classes based upon the CAPTCHA scheme. For classification we explored the performance of naïve Bayes, random forest trees, decision trees, support vector machine (SVM) and multi-layer perceptron (MLP) classifiers using their implementations in WEKA [15]. Each classifier's performance was tested using 10-fold cross validation, as well as by randomly splitting data 70:30 into training and testing sets, respectively. The classification accuracy using each features is shown in Figure 2. Figure 3 shows that using only the top 10 most significant features, we can achieve a classification accuracy of up to 98.7%. However, the improvement in accuracy gets smaller as we continue raising the

Table 3. Accuracies of various classifiers using only top 10 features, all 31 features and Yousra *et al.*'s geometric features

No.	Classifier	Top-10	All	Yousra et al. [14]
1	Naïve Bayes - Cross validation	94.8%	96.6%	83.6%
2	Naïve Bayes - Split testing	96.3%	96.5%	83.6%
3	Decision trees- Cross validation	98.1%	98.8%	91.3%
4	Decision trees- Split testing	97.0%	98.1%	89.1%
5	Multi-layer perceptron- Cross validation	96.7%	99.2%	89.1%
6	Multi-layer perceptron- Split testing	96.9%	98.1%	87.7%
7	Random forest- Cross validation	98.7%	99.4%	92.8%
8	Random forest- Split testing	98.3%	98.9%	91.3%
9	Support vector machines - Cross validation	90.9%	97.6%	67.5%
10	Support vector machines - Split testing	83.8%	96.6%	41.5%

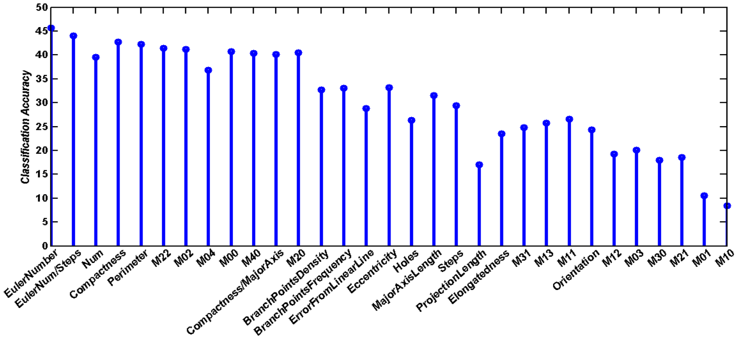


Fig. 2. Classification accuracy of individual features

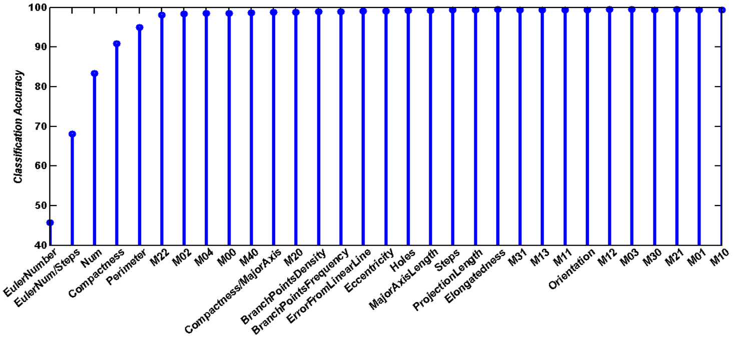


Fig. 3. Classification accuracy of top-*n* features

number of features used for classification. When all features are used by the classifier the accuracy improves by 6.6% to 99.4%, a relative increase of 7.1%. From Table 3 it is clear that using all 31 features results in the best classification accuracy for each classifier followed closely by the case when top 10 selected features are used. They result in better classification than when the measures proposed in [14] were used.

5 Related Work

Yousra *et al.* [14] proposed the use of geometric features for usability analysis of different text based schemes. The geometric features used by them included compactness, Euler number, erosion steps, ratio between compactness and foreground width and ratio between Euler number and erosion steps. Our work is different from Yousra *et al.* [14] because it focuses on identifying / recognizing different CAPTCHA schemes instead of categorizing their usability. We are interested in automatically identifying a CAPTCHA as a Google CAPTCHA, Yahoo CAPTCHA, Bank of Taiwan CAPTCHA *etc.* However, Yousra *et al.*'s [14] work focused on identifying whether a given CAPTCHA was difficult to solve or easy to solve by a human, independent of the CAPTCHA type. Their classification scheme required dividing CAPTCHAs into two categories or classes *i.e.* easy or hard, whereas our work divides different CAPTCHAs into their respective class. When our classifier is restricted to use only the set of geometric features identified by Yousra *et al.* [14], we were able to achieve a maximum classification accuracy of only 92.8%. Thus the classification accuracy of 98.7% that we achieved with our features and classifier demonstrates an increase of 5.9% and a relative improvement of approximately 6.4%. The difference in accuracy of the proposed method and by using the features proposed by Yousra *et al.* can be explained in terms of information gain. Since the parameters defined in [14] have their information gain in the top quartile, so the classification accuracy achieved by them is reasonably high.

To the best of our knowledge, no work is available in the public domain that addresses this issue of CAPTCHA scheme / type identification. CAPTCHA types differ from each other based on the size of text, fonts used in them, distortions added to text and the types of foreground and background noise. The most common forms of noise are addition of random lines, warping of text (locally or globally) and addition of circles of various radii [16]. Other distortions include adding waviness to characters (Reddit), collapsed characters (Google) or broken characters (Yahoo). The purpose of these distortions is to prevent bots from automatically segmenting individual characters and perform recognition of the characters using state-of-the-art optical character recognition (OCR) tools to defeat the CAPTCHA.

The CAPTCHAs are formed using hard AI problems, however, it has been identified that they can be solved by performing specific pre-processing steps. This is done to identify the weaknesses in the CAPTCHA so that the next version is less prone to automated attacks. To identify these weaknesses, Mori *et al.* [9] broke EZ-GIMPY CAPTCHAs with an accuracy of 92% using shape context information for recognition. Jeff Yan *et al.* [5] used a projection technique along with color filling segmentation (CFS) algorithm to break previous versions of Microsoft and Yahoo's CAPTCHAs, which featured random arcs of variable thickness which sometime overlapped with the characters. This was done first by segmenting all the foreground objects using CFS and later on by discarding objects based upon their location and projection information. They exploited the observation that arcs were usually either at the top or bottom of an image and usually had flat projection.

Claudia *et al.* [10] attacked previous version of the ReCAPTCHA scheme and reported a success rate of 40.4%. ReCAPTCHAs had joined characters along with rotational transformation, so Claudia *et al.* proposed orientation correction followed by histogram based techniques to segment characters. They finally used SVM for character recognition.

Yan *et al.* [11] successfully solved the previous version of Google CAPTCHA scheme with an accuracy of 46.75%. The authors exploited geometric patterns like loops, dots and crosses for segmentation.

All of the above cited methods have focused on solving a single CAPTCHA. Thus, it is always assumed that the type of CAPTCHA is known before hand, *i.e.* whether it is a Google CAPTHCA or Reddit CAPTCHA *etc.* Currently the focus has shifted on combining these attacks into a unified framework. In this regard Elie *et al.* [6] have recently developed a tool called *Decaptcha* (not publicly available) that is able of solving 13 out of 15 popular CAPTCHA schemes. The two CAPTCHAs it was unable to solve were the current versions of Google CAPTCHA and Recaptcha. Although, the authors have proposed different algorithms for solving different CAPTCHA schemes they have not disclosed how to recognize a CAPTCHA scheme so that an attack can be launched automatically.

Alongside research on attacking different CAPTCHAs scheme there is a significant body of work that studies the usability of CAPTCHAs. In this regard, Yan [7] explored the factors affecting the usability of CAPTCHA schemes. Elie *et al.* [17] evaluated the usability of CAPTCHA schemes by using Amazon's Mechanical Turk and underground CAPTCHA breaking service. The number of correct answers to CAPTCHA was used as indicator of usability of a scheme. Chien [18] developed a game for usability analysis and also compared its performance with Mechanical Turk.

6 Conclusions

We have developed a novel scheme for the classification of 25 different kinds of CAPTCHAs. To the best of our knowledge this is first work in area of CAPTCHA scheme / type classification. In this paper we have proposed the use of geometric measures for classification based identification of text based CAPTCHA schemes. The results suggest that our parameters can classify schemes with 99.4% accuracy by using Random Forest Trees. This provides the advantage of combining different attacks for various CAPTCHAs in a framework so that attacks can be launched automatically based on CAPTCHA type. Next, we have demonstrated that selecting the 10 most significant features does not affect the classification accuracy by more than 2%, except when using SVM classification. The proposed classification method can be used to effectively identify a CAPTCHA and automatically launch corresponding attack.

References

1. Naor, M.: Verification of a human in the loop or identification via the turing test, `http://www.wisdom.weizmann.ac.il/~naor/PAPERS/humanabs.html` (1996) (unpublished draft)
2. Von Ahn, L., Blum, M., Langford, J.: Telling humans and computers apart automatically. Communications of the ACM 47, 56–60 (2004)
3. Basso, A., Bergadano, F.: Anti-bot strategies based on human interactive proofs. In: Handbook Information and Communication Security, pp. 273–291. Springer (2010)
4. Chellapilla, K., Larson, K., Simard, P.Y., Czerwinski, M.: Building segmentation based human-friendly human interaction proofs (hips). In: Baird, H.S., Lopresti, D.P. (eds.) HIP 2005. LNCS, vol. 3517, pp. 1–26. Springer, Heidelberg (2005)
5. Yan, J., El Ahmad, A.S.: Captcha robustness: A security engineering perspective. Computer 44, 54–60 (2011)
6. Bursztein, E., Martin, M., Mitchell, J.: Text-based captcha strengths and weaknesses. In: Proceedings of 18th ACM Conference on Computer and Communications Security, pp. 125–138. ACM (2011)
7. Yan, J., El Ahmad, A.S.: Usability of captchas or usability issues in captcha design. In: Proceedings of 4th Symposium on Usable Privacy and Security. ACM (2008)
8. Baird, H.S., Coates, A.L., Fateman, R.J.: Pessimalprint: a reverse turing test. International Journal on Document Analysis and Recognition 5, 158–163 (2003)
9. Mori, G., Malik, J.: Recognizing objects in adversarial clutter: Breaking a visual captcha. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. I–134. IEEE (2003)
10. Cruz-Perez, C., Starostenko, O., Uceda-Ponga, F., Alarcon-Aquino, V., Reyes-Cabrera, L.: Breaking reCAPTCHAs with unpredictable collapse: Heuristic character segmentation and recognition. In: Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F., Olvera López, J.A., Boyer, K.L. (eds.) MCPR 2012. LNCS, vol. 7329, pp. 155–165. Springer, Heidelberg (2012)
11. El Ahmad, A.S., Yan, J., Tayara, M.: The Robustness of Google CAPTCHA's. Computing Science, Newcastle University (2011)
12. Gehler, P., Nowozin, S.: On feature combination for multiclass object classification. In: International Conference on Computer Vision (ICCV). IEEE (2009)
13. `www.abchina.com` ,`www.amazon.com`, `www.bankcomm.com`,`www.poynettebank.com`, `www.captcha.fr`, `www.captchacreator.com`, `www.captchator.com`, `www.captchas-asp.co.uk`, `www.puremango.co.uk`, `www.gmail.com`, `www.hotmail.com`, `www.lagom.nl`, `www.protectwebform.com`, `www.google.com/recaptcha`, `www.rediff.com`, `www.reddit.com`, `www.simplemachines.org`, `www.slashdot.org`, `www.en.wikipedia.org`, `www.webspamprotect.com`, `www.white-hat-web-design.co.uk`, `www.wyomingnationalbank.com`, `www.yahoo.com`

14. Nazir, M., Javed, Y., Khan, M.M., Khayam, S.A., Li, S.: Poster: Captchæ cker–automating usability-security evaluation of textual captchas. In: SOUPS (2011)
15. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. ACM SIGKDD Explorations Newsletter 11, 10–18 (2009)
16. Lee, Y.L., Hsu, C.H.: Usability study of text-based captchas. Displays 32, 81–86 (2011)
17. Bursztein, E., Bethard, S., Fabry, C., Mitchell, J.C., Jurafsky, D.: How good are humans at solving captchas? a large scale evaluation. In: 2010 IEEE Symposium on Security and Privacy (SP), pp. 399–413. IEEE (2010)
18. Ho, C.J., Wu, C.C., Chen, K.T., Lei, C.L.: Deviltyper: a game for captcha usability evaluation. Computers in Entertainment (CIE) 9, 3 (2011)