

Enhancing Security of Urdu Language Websites through Urdu CAPTCHA

Imtiaz Ahmed Dahar[†], Fizza Abbas Alvi^{††}, and Ubaidullah Rajput^{†††}

Sukkur IBA University, Pakistan[†],

Quaid-e-awam UEST, Pakistan^{††}

Quaid-e-awam UEST, Pakistan^{†††}

Abstract

Nowadays almost every daily activity such as in education, entertainment, communication, e-commerce (to name a few) are done through Internet. For secure login to such activities, user have to sign-up to access those resource over the Internet. However, a bot (web robot) can sign-up automatically by entering fake information to access those resources. To prevent from automated-bots attack, CAPTCHAs are used to secure websites. CAPTCHA stands for Completely Automated Public Turing tests to tell Computers and Humans Apart and it is a Turing test to confirm that the end user is human or a robot (bot). CAPTCHA was introduced by gang of four (Luis Von Ahn, Manuel Blum, Nicholas J. Hooper) at CMU in 2000. There are many types of CAPTCHAs including text based, basic questions, mathematical questions and selecting images, whereas scrambled text-based CAPTCHAs are mostly used. However, nowadays bots are very intelligent to break alpha-numeric scrambled text-based CAPTCHAs. Most of CAPTCHAs are in English, European and in east Asian languages but unfortunately Urdu language has less focus. According to the literature, CAPTCHAs should be designed in local and regional languages including Urdu and Sindhi to improve the security, accessibility and understandability of our regional websites. This research aims to propose an Urdu language-based CAPTCHA for regional URDU websites. This research highlights the limitations of exiting available CAPTCHAs and compares the efficiency of our proposed CAPTCHA with exiting work. The results show that our proposed CAPTCHA provides robustness and efficiency in terms of complexity.

Key words: Urdu, CAPTCHA, security, efficiency.

1. Introduction

In fact, almost every daily activity like education, entertainment, communication, e-commerce and other tasks are done through internet. To perform such type of activities, users have to sign-up to access those resource over the Internet. A bot can sign-up automatically by entering fake information to access those resources. Bots (web robot) are automated programs that abuse the websites, for example, automatically formfilling, signing up into multiple accounts, make registrations with fake data and sending junk emails. Most of the bots are used for web crawling in which data and information is fetched from web servers and half of web traffic is made by bots. There are various types of bots

including Social bots, Commercial bots, Monitoring bots, search engine bots and Malicious bots (known as bad bots) [1-3]. Whereas some good bots are there as well, which are helpful for Internet users like chat bots, auto reply bots for emails, lowest price finding bots (to name a few). CAPTCHAs are used to defend the websites from bots' attack. CAPTCHA is Turing test used to distinguish between human and computer. This test ensures that no computer programs (bots) are accessible to access such type of sensitive data provide by web sites and portals. It was introduced by Luis Von Ahn, Manuel Blum, Nicholas J. Hooper at CMU in 2000 [4]. These days, CAPTCHA is standardized as a security mechanism to secure the websites from automatic bots over the internet. It is small program which is used to generate various types of tests that could be difficult for computer to pass. On other hand, CAPTCHAs are proposed in different forms including scrambled text-based, selecting images, mathematical questions and basic questions. Whereas text-based CAPTCHAs are highly appreciated by companies because they are easy in implementation and it is still widely used by web applications. But nowadays bots are becoming intelligent every day with the help Artificial Intelligence (AI) and they easily break text-based CAPTCHAs [4]-[7]. In the past, many CAPTCHA breaking techniques (also known as anti-CAPTCHA) were proposed and some of them got higher success ratio by using powerful machine learning algorithms including support vector machine (SVM), K-nearest neighbor and convolutional neural networks (CNN). Two methods used to break the CAPTCHAs by using machine learning, first method needs large number of data set of real CAPTCHAs to train the breaking system and they can be downloaded from the website which is aimed to be broken. However, other one method does not need large dataset to break the CAPTCHAs and this type of CAPTCHA solver system uses Generative Adversarial Network (GAN) [8]. The research provides new CAPTCHA challenges for Optical character recognition (OCRs) because it is based on multiple features (mentioned in next sections). OCR is technique to convert image-based data into text-based form which could be helpful for searching, sorting, filtering and editing the data. These days, OCRs are also used to break the CAPTCHAs














Type	Example	Source	Features
Solid CAPTCHA		Discuz!	Character independent, texture background, some interference
		Slashdot	A large number of interference lines and noise point
		Gimpy	Multiple strings, overlap, distortion
		Google	Unfixed length, distortion, adhesion
Hollow CAPTCHA		Microsoft	Double string, unfixed length, uneven thickness, tilting, adhesion
		QQ	Hollow, shadows, interference shapes
		Sina	Hollow, adhesion, interference lines
Three-dimensional CAPTCHA		Yandex	Hollow, virtual contours, distortion, adhesion, interference lines
		Scihub	Hollow, shadows, interference lines, noise points
		Teabag	Grids, protrusion, distortion, background and character blending
Animation CAPTCHA		Parc	Colorful, shadow, rotation, zoom
		Program generating	Multiple characters jumping
		Hcaptcha	Multilayer character images blinking transformation

Fig 1. Image-based CAPTCHA Sample.

by which using three main steps including preprocessing, segmentation and recognition [9]. Text-based CAPTCHAs are easiest and simple form of CAPTCHA to secure websites from bot attacks. Early days simple text-based CAPTCHAs were used but nowadays bots are very intelligent, and they can easily break this type of CAPTCHA. Text-based CAPTCHA can be made of simple alphabets, numbers, alphanumeric, symbols and mathematical equations along with noisy background and lines. Almost every website implements text-based CAPTCHA because it is easy to implement with low cost [10, 11]. Apart from text-based CAPTCHAs, another type of CAPTCHA is image-based CAPTCHAs that uses images of people, animals and objects rather than scrambled alphanumeric text to verify that user is human or robot. Whereas image bases CAPTCHA system provides number of images and asked a user to recognize image according to given word. Figure 2 shows image-based CAPTCHA [7, 12]. Finally, some CAPTCHAs are audio based or sound-bases that contains small auto-clips. This type of CAPTCHA is developed for visually disabled people. Firstly, user has to listen the auto-clip before submitting the CAPTCHA. Audio-clips can be random words or a sequence of alpha-numeric letters with some distorted noise which can be only recognized by human [7, 13].

1.1 Problem Statement

With exponential growth of web users, web applications are expanding their services in vast categories and they are also increasing their security mechanism for offered services. Web services are provided to clients and sometime websites may not require user to login to access those services but at the same other services may be required to login from being misused. A plethora of algorithms have been proposed for CAPTCHAs and most of them are in English, European and east Asian languages [14-16]. However, a small amount of research has been done to make regional/local CAPTCHAs, unfortunately Urdu has less focus. It has been observed by



Fig 2. Text-based CAPTCHA Samples.

researchers that non-English speakers' performance is less accurate and slower on solving the English CAPTCHA test. Therefore, it is desirable to make regional/local CAPTCHAs for regional users. Most of internet users in Pakistan are Urdu speaker and they feel difficulty while visiting English based websites. Therefore, most of government and non-profit websites are made in Urdu language for the easiness of users, but the main issue in those Urdu websites are that they still use Latin based CAPTCHAs, likely, it rises usability, accessibility and understand-ability issues because of non-native speakers [15]. On the other hand, Latin script-based CAPTCHAs are vulnerable as compared to Arabic scripted CAPTCHAs [10]. Arabic script like languages are complex as well as they have many versions of same characters including isolated character, initial, middle, and final with different dots, position of dots, cursive and non-cursive [14]. Therefore, it has many advantages being used in CAPTCHAs because it will be hard for computer to break those CAPTCHAs. According to authors [17, 18], CAPTCHAs used in E-Commerce website are vulnerable and successfully broke the CAPTCHAs with an overall precision of up to 82.4%. Arain et al. suggested CAPTCHAs should be designed in local languages including Urdu and Sindhi to improve the security, accessibility and understand-ability of our regional websites [15]. Therefore, the aim of this research is to propose Urdu CAPTCHA for regional websites.

2. Related Work

This section explains the related work of existing CAPTCHA and techniques to secure websites.

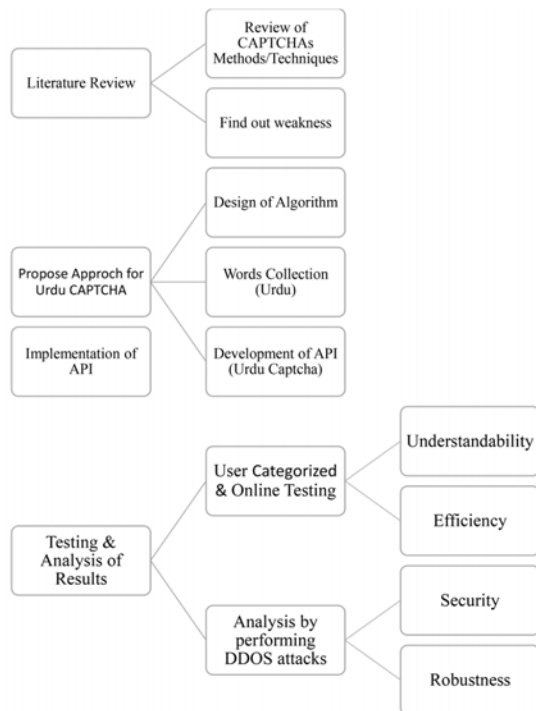


Fig 3. Main architecture of proposed approach

In this section, study of CAPTCHA and its kinds are provided. CAPTCHAs are not new in field of image processing, it was proposed by Reshef et al. in 1997 and developed first CAPTCHA based on bi-colors black and white with English capital characters. The aim of their research was to design a system which could find difference between human and computer. After that, their method was implemented that into well-known website AltaVista search engine to avoid the bots from adding URLs into website [19].

The term CAPTCHA also known as (Completely Automated Public Turing test to tell Computers and Humans Apart) was introduced by Luis von Ahn at CMU in 2000 and claimed the invention of CAPTCHA. Initially, CAPTCHA was developed in various forms including text-based, GIMPY and OCR based. In early years, it was observed that text-based CAPTCHAs were suitable solutions for web security. At the present time, bots became more intelligent and they use different OCRs techniques to break text-based CAPTCHAs [4].

Chellapilla et al. analyzed that CAPTCHAs must be understandable so human can solve it easily while they should too hard or robust for computer. HIPs have also major role in protecting the services from attacks of automatic bots or scripts. Examples of services could be online registrations, spam emails, Denial-of-service attack, blogs and chat rooms [20].

Since first version of CAPTCHA, researchers and companies tried to implement CAPTCHA in different form



Fig 5. Meaningful Urdu CAPTCHAs

(like text, image and audio) to increase the security and ensure that services are being used by human, not a computer robot. Ishfaq et al. discussed different kinds of CAPTCHAs in his paper namely Scrambled Text Based, Picture Identification Based, 3DSuper CAPTCHA, Mathematical Problems Based, Ad-injected CAPTCHA and Social-media. Moreover, authors discussed that Socio Captcha would be suitable solution to generate CAPTCHAs based on user's social profile so user can easily solve the CAPTCHAs because he/she is already familiar of their personal data and it would make full of surprises and fun for the users [21].

Khan et al. proposed Arabic CAPTCHA and claimed that their proposed system is more robust than Persian CAPTCHA. Authors used VB.net programming to developed Arabic CAPTCHA and used 50 variations of fonts with different background and foreground color. Furthermore, Khan also implemented algorithm to avoid from brute force attacks in which IP addresses were recorded to ensure that CAPTCHA generation requests are not from same IP. In that condition, CAPTCHA's complexity level increases from easy to medium and medium to more complex [22].

Hassan et al. develop Persian CAPTCHA by using well-known programming language JAVA programming and embedded that program in Java Applet website format. While, random meaning less words were used to generate Persian/Arabic CAPTCHAs with random fonts (including Shekasteh, Nasta'liq and Naskh) and back-ground lines. Additionally, proposed system was tested by two commercial OCRs ReadIris and Omnipage. However, OCRs could not break the CAPTCHA [16].

As we know, everyone is talking about digital currency and many financial transactions are done through this currency. In this regard, CAPTCHAs work as Man-in-the-Middle to protect the websites from being stolen the customer's money. However, E-banking CAPTCHAs are also vulnerable, and authors broke these CAPTCHAs with success ratio of 100% or close to. Authors used image processing and pattern recognition methods to break existing e-banking CAPTCHAs and tried these techniques into 41 different e-banking websites. Finally, Li et al suggested that e-banking CAPTCHAs should be replaced by another suitable solution [23].

Megaupload is largest file sharing company which provides different services including file storage for files, images and videos. They have more than 10 websites (mega-video.com, megapix.com megapay.com and others) to provide the services. Megaupload also uses CAPTCHAs to secure their services from unauthorized users (bots). The

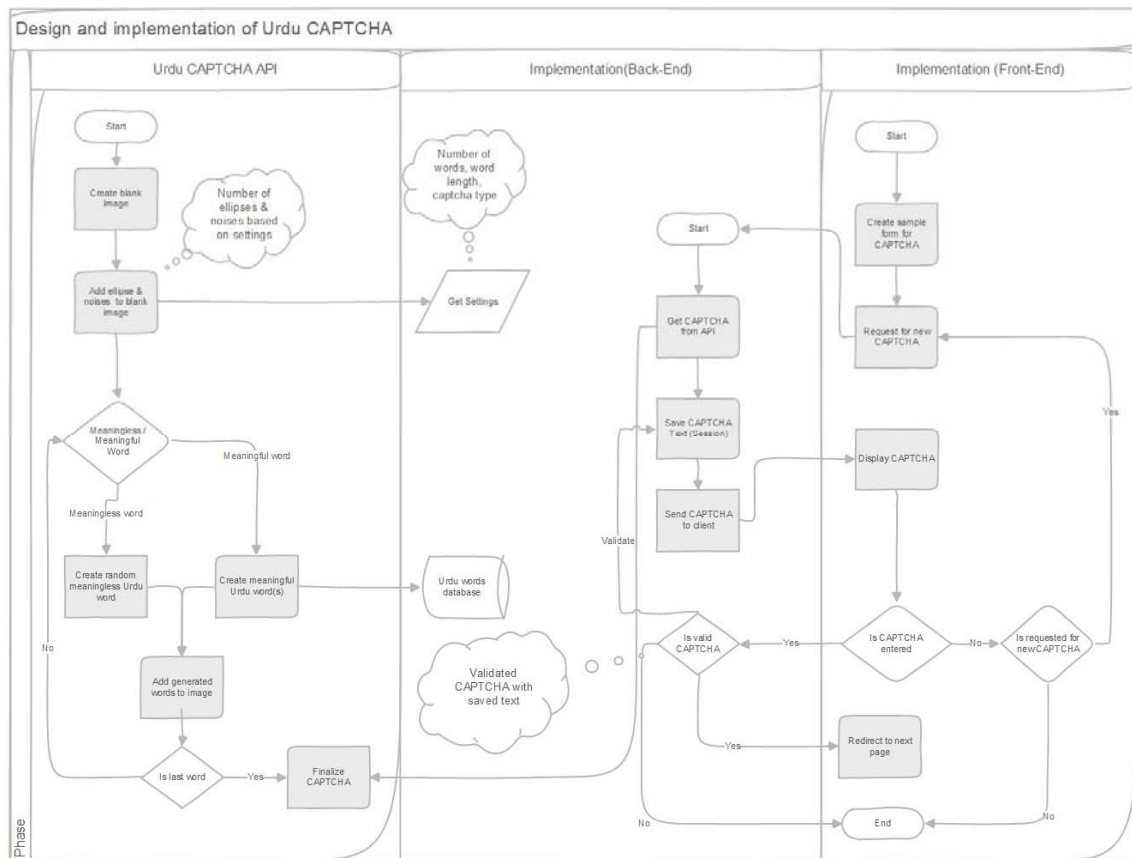


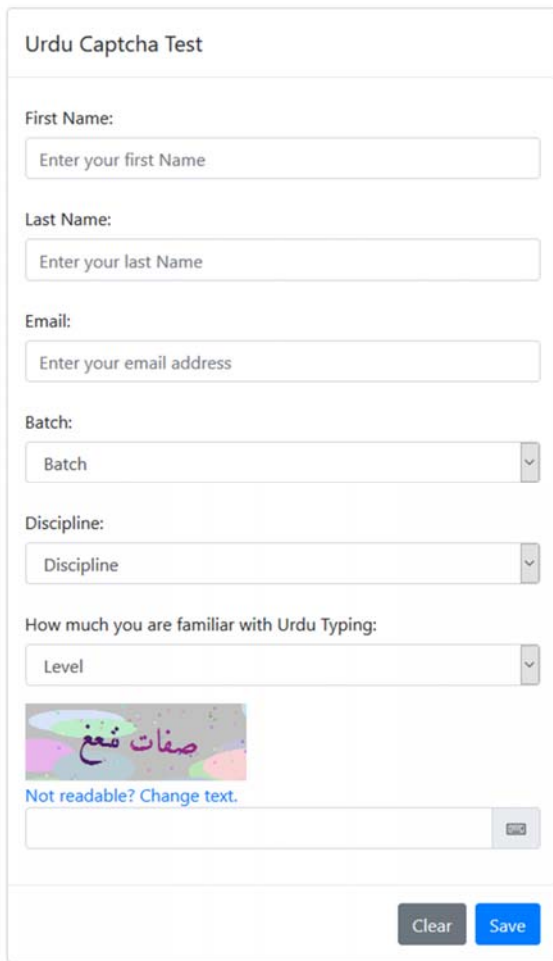
Fig 4. Internal architecture of proposed approach

aim of authors [5] was to break newly developed CAPTCHA (Megaupload) and broke with the success rate of 78%. They used “segmentation-resistant” techniques with 120 ms average time in segmentation challenge [5]. Datta et al. proposed IMAGINATION image-based user-friendly CAPTCHA in which composite image is created by tiles of 8 images. The purpose of their research to increase the robustness of CAPTCHAs and users are asked to click on center of any tile (included in composite image) then write the word matched to image, this methodology could difficult for automated system while human can clearly recognize the images [24]. Fidas et al. worked on localized and Latin-based CAPTCHAs and discussed various issues that localized CAPTCHA will increase the usability and user experience according to User Centered Design (UCD). Thus, authors involved the real users (junior and senior students from two academics) and analyzed the results of localized vs Latin-based CAPTCHAs and presented 440 different challenges to students while the age of students were average 23. As a result, authors also suggested that localized CAPTCHAs must be used for regional and local websites because more

than 60% internet users are not known of English or Latin alphabets [18].

3. Proposed Urdu Captcha Framework

Proposed framework has multiple phases like designing of algorithm for Urdu CAPTCHA, data collection and development of API. It is observed that CAPTCHA breaking techniques are based on width of CAPTCHA, font size, color, style, number of characters, feature and overlapping the characters. Therefore, we have kept those things in mind before designing of the API so we can make our proposed CAPTCHA more secure and understandable as possible. Designing of algorithm for Urdu CAPTCHA is most important phase of our proposed system that is used into API for generating random Urdu words for our CAPTCHAs. First, we have designed our algorithm that how our proposed framework will look like, how it will be implemented in any web application (ASP.Net) and which type of Urdu words (meaningless/meaningful) will be used. This phase is heart of our proposed system and it is developed in C# as Urdu CAPTCHA API to implement in any of ASP.net application as shown in Figure 4. C# is one



Urdu Captcha Test

First Name:

Last Name:

Email:

Batch:

Discipline:

How much you are familiar with Urdu Typing:



 Not readable? [Change text.](#)

Fig 6. Sign-up Web Form

of best programming language in .Net framework and it is developed by Microsoft [25]. The output of our API is in Dynamic Link Library (DLL) format which can easily be implemented without installing any 3rd party softwares. It can also be deployed in nuget packages so users can easily implement into their web application. Proposed framework is made flexible that would easily be extended according to user's requirement. Configurations of API is based on user's requirements and user can decide how many numbers of words should be used in Urdu CAPTCHA, number of characters per word, meaningful and meaningless words as well. Moreover, it can also be configured that how many ellipses and noises should be used in CAPTCHA to make it simple or harder. Proposed approach has many interfaces for configuration of API just by setting-up the values in API. Figure 4 shows that API is getting configuration values from out-side. As long as, API has default settings if configuration values are not provided by user. Default settings of API are Single meaningful word with 4 to 8 character length of Urdu word and word could be



Urdu Captcha Test


 Not readable? [Change text.](#)

Fig 7. CAPTCHA Testing Web Form

meaningful or meaning less. Whereas words are randomly selected from database based on server's time to make sure that each word is random and it's length is between 4 to 8 characters. Additionally, for meaningless Urdu CAPTCHAs, each character is selected randomly to make sure that every word is different and some of isolated Urdu letters are removed to make joint Urdu word. Similarly, if words are greater than one, then each word's color will be different to increase the robustness of our proposed framework. Color of words are also selected randomly. The rotation of words can increase the complexity of CAPTCHA so it will be hard for bots to break, bots are trained for specific format of CAPTCHA. If we rotate some angles of our CAPTCHAs, bots will be supposed to retrained for new CAPTCHA sets. Therefore, we are also rotating our words in different angle for single and double words. Finally, various fonts are used to make our CAPTCHAs more unique. We have used more than 20 (Table 3) different fonts, which are also free available at softlay.net [26]. API is developed in different stages as shown in Figure 4. First, blank image is created with specified width and height, default width and height are 200, 80 pixels respectively. Second, Noises and ellipses are added to blank image according to user's requirements. However, we have also set default values for ellipses and noises so user do not need to provide those values and 20s are used for ellipses and 50s are for noises. Noises and ellipses has also different colors which are randomly selected. Third, API will add Urdu words based on user's settings like single word, double word, meaningful words, meanings less and length of each word. The rotation of words are supposed be used in different angle so OCRs could not find the base line of words for recognizing. Additionally, if user has selected single meaningful word option in settings, API will fetch a word from database (Urdu words) with particular length of word. Words will be fetched from database based on length of word provided by user. We have large number of words in our database and those words are fetched whose length is between minimum

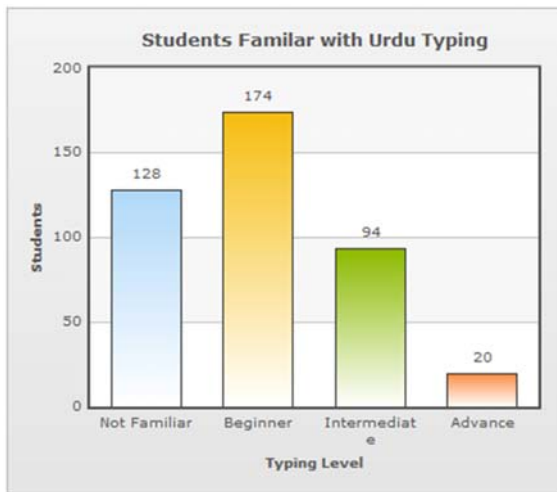


Fig 8. Number of students familiar with Urdu typing

and maximum length according to user's need. For example, we are setting length of word is between 4 to 8 characters, therefore, Urdu words will be fetched from database whose length is between mentioned characters. Furthermore, If user has decided for meaningless word then API will choose random letters based on length of word and it will keep going on until number of CAPTCHA words are not completed. We can increase the width of CAPTCHA so we can use more than 2 words, for now, our default settings are maximum 2 words per CAPTCHA because width of image is 200 pixel, if we add more words into image they will cross the length of image and will be half-cut CAPTCHA. Therefore, before adding more than 2 words in CAPTCHA we have to increase the width of image as well. Finally, word will be added to CAPTCHA with different and random font. We have as already discussed that font style has also big role in CAPTCHAs to create complexity for OCRs., color and angle after that, word will be stored in RAM for temporary bases to verify with entered CAPTCHA. Additionally, user can also configure the database settings that API will use internal or external Urdu word's database for generating Urdu meaningful words. Finally, we compiled our API into DLL file so we can implement it in our test web application project. In the end, we have collected Urdu words from different sources including Android Urdu dictionaries (Offline Urdu Lughat - Urdu to Urdu Dictionary, English to Urdu Dictionary), pdf books and on-line websites (<https://www.rekhta.org>, <http://www.urduword.com>, <https://www.urdupoint.com>). We developed a custom soft- ware to extract words from different source and saved those Urdu words into our SQL database. As we known vocabulary is compulsory for meaningful Urdu CAPTCHAs. For development and testing purpose we have used MS SQL Server 2012 Express Edition, which is free and lightweight to implement our CAPTCHA, and it also easy to use in ASP.Net websites.

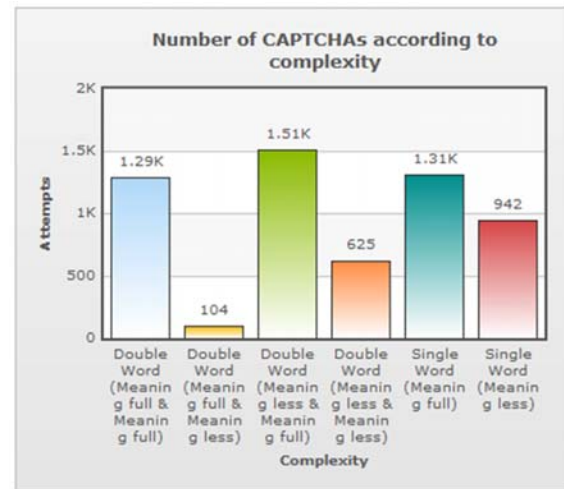


Fig 9. CAPTCHA attempts according to complexity

We have extracted more than 1,16,000 unique Urdu words for our Urdu CAPTCHA. The implementation of our proposed Urdu CAPTCHA's is done with C# language because of it is open source and cross platform that can be used in any platform including Mac OS, Linux and Windows as well. Additionally, we have removed special characters like (Zer, Zabar, Pais) [27] from Urdu Word's database to make our CAPTCHA more understandable and easier for users. Words were extract into form of paragraphs and divided into sentence, after that sentences were divided into words while special character were also removed from that word. Finally, Words were checked into our database before saving, if extracted word was not found in database then it was flagged a new word and saved into database otherwise words were skipped as a duplicate word.

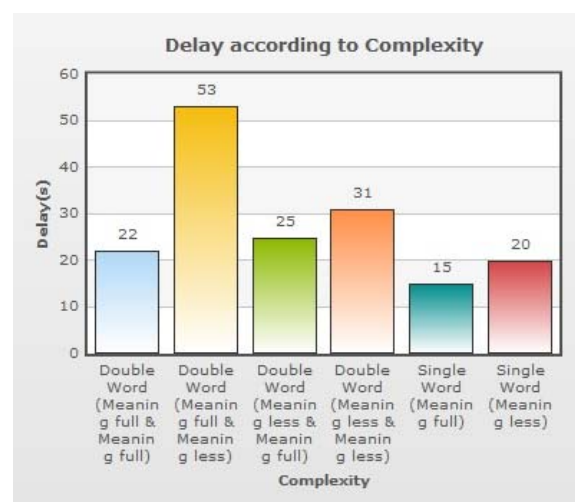


Fig 10. Response time according to complexity

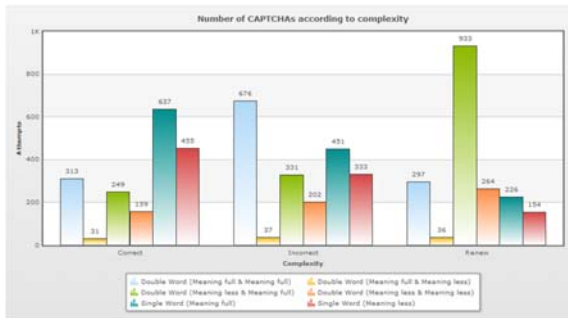


Fig 11. Comparison of CAPTCHAs according to complexity

After developing API, we have added developed Urdu CAPTCHA's API into our project by adding reference of DLL file. Proposed system is developed in two projects, first one for API named "Urdu Captcha" and second one for implementation of API in front-end web application named "CAPTCHA Web Test". Additionally, we can see different properties of our API which can be configured according to user/client's requirement. After that we have set the configuration settings for CAPTCHAs including number of

Table 1. Urdu CAPTCHAs

	Single Word	Double Words
Meaningful		
Meaningless		
Meaningful & Meaningless		
Meaningless & Meaningful		

words per CAPTCHA, range of characters per word, meaningless/meaningful words, number of ellipses and noises to make more difficult or easy CAPTCHA according to our requirement. Another point is, we can decide which Urdu word's database (internal/external) should be used for our CAPTCHAs by providing database connection string (SQL server's username and password) to access the internal/external database. API is implemented in ASP.Net Web forms application, it is early version and most successful product of .Net framework for web development. ASP.net supports C# language which is now open source and cross platform. After completing all configurations, API is ready to generate Urdu CAPTCHAs and it has public methods which will give us CAPTCHA image and text which are used in image so we can compare the result text got by user with the text generated by our API.

On other hand, for testing purpose we had opened our web application deployed in server and visited our main sign-up

Invalid Captcha	Valid Captcha	Recaptcha	Captcha	Text	Complexity	Response Time
	✓	✓	ن	غنائی	2	0
	✓	✓	سکون	صنحوار	2	64
	✓	✓	مہارت	طیاعت	2	48
✓			لال	لال	2	14
✓			نہایت	نہایت	2	136

Fig 12. CAPTCHA results of a student

form Figure 6. While visiting website, HTTP request was sent from front-end to back-end (ASP.Net server) and then, proposed system gets the Urdu CAPTCHA from API based on system's configuration, after that CAPTCHA image is sent back (response) to front-end as well CAPTCHA text stored in session (to verify with entered text) as shown in Figure 4. Whenever, user enters the CAPTCHA text and submit the form, text will be submitted to back-end for verification of CAPTCHA that is it correct or not by comparing both strings generated by CAPTCHA and user's entered string. Based on verification, response will be sent to front-end based to show that entered CAPTCHA is true or false. However, if CAPTCHA entered by user is correct then sign-up form will be submitted to create user's account for furthermore testing of our API so we can track the information entered by users (students). For testing purpose, we are collecting basic information of students like their first, last name, email address, year of study (batch), discipline (in which department they are studying) and how familiar they are with Urdu typing so we can calculate the average response time of users'.

On top of that, front-End implementation is done with basic web technologies including HTML5, CSS3, Bootstrap and JQuery. HTML stands for Hyper Text Markup Language, the idea was coined by Tim Berners-Lee in 1990 as a standard markup language for web applications. Whereas CSS is Cascading Style Sheet and is an ornament of web page that describes the presentation of documents. Moreover, Bootstrap is CSS framework which helps to create responsive web pages that support all types of devices' resolutions including mobile, tablet, laptop and desktop computers. Finally, JQuery is java-script framework used for client-side validation, it is easy to use as compared to vanilla java-script and supports almost all browsers. To illustrate, we have developed a sample web form to measure the response time, delay and robustness of our CAPTCHA by conducting different types of tests from university students and tests was conducted from two universities (Sukkur IBA University and QUEST University Nawab-shah). Figure 6 shows the web form, where students have to sign-up with simple data like first name, last name, email address, year of admission (batch),

Table 3. Urdu fonts used in proposed approach

Font Name	Preview	Font Name	Preview	Font Name	Preview
Ahmed	ہمہ روز بولتے ہیں	Aldhabi	ہمہ روز بولتے ہیں	AlQalam Taj Nastaleeq	ہمہ روز بولتے ہیں
AlQalam Tehreeri	ہمہ روز بولتے ہیں	AlQalam Ubaid	ہمہ روز بولتے ہیں	Asad	ہمہ روز بولتے ہیں
Attari Salees	ہمہ روز بولتے ہیں	Electron Unicode	ہمہ روز بولتے ہیں	Farsi Simple Outline	ہمہ روز بولتے ہیں
Gandhara Suls	ہمہ روز بولتے ہیں	Irqam Unicode	ہمہ روز بولتے ہیں	Jameel Noori Nastaleeq	ہمہ روز بولتے ہیں
Kumail	ہمہ روز بولتے ہیں	Riyaz Unicode	ہمہ روز بولتے ہیں	Sadaf Unicode	ہمہ روز بولتے ہیں
UL Amjad Outline	ہمہ روز بولتے ہیں	UL Amjad Shaded	ہمہ روز بولتے ہیں	UL Amjad	ہمہ روز بولتے ہیں
Urdu Naskh Unicode	ہمہ روز بولتے ہیں	PDMS Bukhari	ہمہ روز بولتے ہیں		

discipline and Urdu typing level. While visiting the website, CAPTCHA Request was sent to back-end server which returns the response to front-end as a Urdu CAPTCHA image based on API's configuration and saved text in session for verification of CAPTCHA. When students signed-up, passwords were sent to their email address for sign-in. After all process, students signed-in and entered the CAPTCHAs (Figure 7). The tests were conducted 20 times by each student with different configurations including single, double, meaningless and meaningful Urdu CAPTCHA. Hundreds of students participated in test from different batches of both universities, we conducted tests from department of computer science students.

4. Results & Discussion

This section describes the detail results & discussions of proposed Urdu CAPTCHA.

4.1 Results

Finally, Urdu CAPTCHA API is developed and implemented in ASP.net Web Form application for testing. Overall, it is clear that complexity is inversely proportional of response time, when we increased the complexity of CAPTCHA, response time was decreased. Furthermore, we have analyzed that single meaningful word is suitable for all types of regional website because users can easily understand and solve CAPTCHA in less time. To analyze the effectiveness of Urdu CAPTCHA, the API was tested by two universities' students and 416 students participated. Figure 8. shows that how much users (students) were

familiar with Urdu typing. There were four types of students who were familiar of Urdu typing including Not Familiar, Beginner, Intermediate and Advanced. Students participated in testing were from first year to final year of bachelor's degree. It is clear that most of users were beginner of Urdu typing, 174 users they don't know much about Urdu typing or Urdu keyboard. On other hand, only 20 users were good at Urdu typing (Advanced) while 94 were average (intermediate) and 128 were Not Familiar. 5784 CAPTCHAs were entered by 416 students and average 14 CAPTCHAs were entered by each student.

4.2 Discussions

Table 1 illustrates the CAPTCHA samples generated by proposed system. It can be seen that all generated CAPTCHAs are single, double, meaningless and meaningful words with 4 to 8 characters limit. Furthermore, Table 1 also shows the double words Urdu CAPTCHA including first meaningful and second meaningless word. Additionally, other two words are meaningless and meaningful. However, all these CAPTCHA were captured during conducting the tests from students. Figure 12 provides the information about invalid, valid, renew and entered (text) CAPTCHAs with different complexity and response time. Whereas complexity 2 means single meaningful word. Additionally, it can be seen that entered (text) CAPTCHAs are compared with image-based CAPTCHAs and response time is also measured in seconds while ReCaptcha (renew) has zero response time. The bar chart (Figure 9) provides the information about CAPTCHAs entered by different students with different

Table 2. Comparison of proposed API with early CAPTCHA

Feature	Authors [15]	Proposed
Number of Words	English and Urdu	Both Urdu
Characters Nature	Joint and Separated	Joint
Virtual Keyboard	Yes	Yes (Auto Switched)
Users	50	416
Average Response Time (seconds)	24.5	15
Fonts	1	20
Colors	Bicolor (Black & White)	Multi Colors

complexities. It can be seen that students were less interested in complex CAPTCHAs and they didn't try to solve complex CAPTCHAs and only 104 CAPTCHAs were entered in double words meaningful & meaningless complexity due to large words as shown in Table 1. While 31 attempts were correct, 37 and 36 CAPTCHAs were incorrect and renewed, respectively. Moreover, second least attempts were tried in double meaningless words complexity and 625 CAPTCHAs were entered due to ambiguity of meaningless Urdu words. However, 159 were correct, 202 were incorrect and 264 were renewed (Table 1). Third, as we know that meaningless words are ambiguous than meaningful words. Therefore, renew ratio was at high peak around 1000 CAPTCHAs were renewed in double words meaningless & meaningful complexity while only 249 were correct and 331 incorrect, and total 1510 CAPTCHAs were tried. Table 3 shows the list of fonts used in proposed system. As we know, OCRs can break CAPTCHAs easily when fonts are simple, so we have increased number of fonts to make our CAPTCHA more robust. However, human can easily understand these fonts at one glance while it will be hard for a bot to break the CAPTCHA. Overall, it is clear that all double words with meaningless CAPTCHAs were too much complex to solve them in less time. Moreover, they were also ambiguous to understand therefore correct ratio was less than incorrect and renewed. On other hand, it can also be seen that all meaningful Urdu CAPTCHAs were easier than meaningless either they were single word or double words. Figure 11 shows that single word Urdu CAPTCHAs has highest correct ratio and 637 CAPTCHAs were correct while 451 incorrect and 226 renewed out of 1300 (Figure 9) attempts. Figure 10 gives the information about response time based on different complexities. Response time (delay) of CAPTCHAs are measured in seconds (average). It is clear that single meaningful word CAPTCHAs were easily solved in less time about 15 seconds (average). Second least,

single word meaningless CAPTCHAs were solved in 20 seconds while double meaningful words were solved in 22 seconds. Overall, it is clear that all single and meaningful CAPTCHAs were easy and solved in less time than 3 other types of CAPTCHAs. On other hand, it can be seen that all double words were too ambiguous to understand. Therefore, the response time (delay) was high as compared to single words.

4.3 Comparison of Results

Table 2 illustrates the comparison of proposed API with early developed Urdu CAPTCHA. However, it can be seen that proposed system is better than other one. Proposed system has 20 different fonts with multi colors while early system has used only 1 font with bi-colors. Moreover, response time of proposed API has 15 seconds as compared to others 24.5 seconds. Additionally, nature of words is joint in our system while older one has used joint and separated technique. Finally, our system is tested by 416 students of two universities to get more accurate results.

4. Conclusion and Future Work

This paper highlights the limitations of existing Captcha for regional URDU websites. Furthermore, the paper proposed a robust Urdu Captcha framework for regional websites. The results show that the proposed Captcha is efficient in terms of response time, number of words, character nature and fonts. The proposed Urdu Captcha is robust and secure for regional websites. In future, we aim to extend our work to other regional languages such as Sindhi and Punjabi.

References

- [1] K. Dunham and J. Melnick, *Malicious bots: an inside look into the cyber- criminal underground of the internet*. Auerbach Publications, 2008.
- [2] I. Zeifman, "Bot Traffic Report 2016," Jan. 2017. [Online]. Available: <https://www.incapsula.com/blog/bot-traffic-report-2016.html>
- [3] <https://www.globaldots.com>, "Bad Bot Report 2018 | GlobalDots." [Online]. Available: <https://www.globaldots.com/bad-bot-report-2018/>
- [4] L. V. Ahn, M. Blum, and J. Langford, "Telling humans and computers apart automatically," *Communications of the ACM*, vol. 47, pp. 56–60, 2004.
- [5] A. S. E. Ahmad, J. Yan, and L. Marshall, "The robustness of a new captcha," in *Proceedings of the Third European Workshop on System Security. Proceedings of the Third European Workshop on System Security, 2010*, pp. 36–41.
- [6] P. N. Vidya and S. Naika, "Simple text based captcha for the security in web applications," *International Journal of Computer Science and Mobile Computing*, 2015.
- [7] V. P. Singh and P. Pal, "Survey of different types of captcha," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 2, pp. 2242–2245, 2014.

- [8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.
- [9] M. Tang, H. Gao, Y. Zhang, Y. Liu, P. Zhang, and P. Wang, "Research on deep learning techniques in breaking text-based captchas and designing image-based captcha," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 10, pp. 2522–2537, 2018.
- [10] E. Bursztein, M. Martin, and J. Mitchell, "Text-based captcha strengths and weaknesses," in *Proceedings of the 18th ACM conference on Computer and communications security*. *Proceedings of the 18th ACM conference on Computer and communications security*, 2011, pp. 125–138.
- [11] J. S. Mtebe and A. W. Kondoro, "Accessibility and usability of government websites in tanzania," *The African Journal of Information Systems*, vol. 9, no. 4, p. 3, 2017.
- [12] J. Anil, G. S. Naveli, and S. Bhukya, "Image based captcha generation system," *International Journal of Pure and Applied Mathematics*, vol. 118, no. 24, 2018.
- [13] W. Zhang, "Zhang's captcha architecture based on intelligent interaction via ria," in *2010 2nd International Conference on Computer Engineering and Technology*, vol. 6. IEEE, 2010, pp. V6–57.
- [14] S. A. Alsuhibany, "Generating arabic handwritten captcha for cyber security," *International Journal of Computer Science and Network Security*, vol. 18, pp. 41–47, 2018.
- [15] M. T. Banday and N. A. Shah, "Challenges of captcha in the accessibility of indian regional websites," in *Proceedings of the Fourth Annual ACM Bangalore Conference*. *Proceedings of the Fourth Annual ACM Bangalore Conference*, 2011, p. 31.
- [16] M. H. Shirali-Shahreza and M. Shirali-Shahreza, "Persian/arabic baffletext captcha," *Journal of Universal Computer Science*, vol. 12, pp. 1783–1796, 2006.
- [17] R. H. Arain, R. A. Shaikh, K. Kumar, A. Maitlo, A. Kehar, S. A. Shah, and H. Shiakh, "Verifying the robustness of text-based captchas offered by local e-commerce sites," *International Journal of Computer Science and Network Security*, vol. 18, pp. 79–84, 2018.
- [18] C. Fidas and A. G. Voyiatzis, "On users' preference on localized vs. latin-based captcha challenges," in *IFIP Conference on Human-Computer Interaction*. *IFIP Conference on Human-Computer Interaction*, 2013, pp. 358–365.
- [19] E. Reshef, G. Raanan, and E. Solan, "Method and system for discriminating a human action from a computerized action," *Patent US 2005/0 114 705 A1*, Mar., 1997.
- [20] K. Chellapilla, K. Larson, P. Simard, and M. Czerwinski, "Designing human friendly human interaction proofs (hips)," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2005, pp. 711–720.
- [21] H. Ishfaq, W. Iqbal, and W. B. Shahid, "Attaining accessibility and person-alization with socio-captcha (scap)," in *Applied Sciences and Technology (IBCAST), 2015 12th International Bhurban Conference on Applied Sciences and Technology (IBCAST), 2015 12th International Bhurban Conference on*, 2015, pp. 307–311.
- [22] B. Khan, K. Alghathbar, M. K. Khan, A. M. AIkelabi, and A. Alajaji, "Cyber security using arabic captcha scheme." *Int. Arab J. Inf. Technol.*, vol. 10, pp. 76–84, 2013.
- [23] S. Li, S. Shah, M. Khan, S. A. Khayam, A.-R. Sadeghi, and R. Schmitz, "Breaking e-banking captchas," in *Proceedings of the 26th Annual Computer Security Applications Conference*, 2010, pp. 171–180.
- [24] R. Datta, 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*. ACM, 2005, pp. 331–334.
- [25] Microsoft. (2019, Feb.) Urdu. [Online]. Available: <https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet>
- [26] S. EDITOR. (2018, Feb.) 500 best urdu fonts collection free download for windows 7/8/10. [Online]. Available: <https://softlay.net/graphics-design/fonts/500-best-urdu-fonts.html>
- [27] <https://www.columbia.edu>, "Diacritics: zer, zabar, and pesh." [Online]. Available: <http://www.columbia.edu/itc/mealac/pritchett/00urdu/urducscript/section05.html>



Imtiaz Ahmed Dahar received the bachelor's degree in computer science from the Shah Abdul Latif University (SALU), Pakistan, in 2015, and his master's degree in Software Engineering from Quaid-e-Awam University of Engineering, Science and Technology (QUEST), Pakistan. His research interest is Natural Language Processing (NLP). He has more than 5 years of software development experience and currently working at Sukkur IBA University, Pakistan, as an Assistant Manager IT.



Fizza Abass received the bachelor's degree in computer system engineering from the Quaid-e-Awam University of Engineering, Science and Technology (Quest), Pakistan, in 2007, and the master's degree in communication system and networks from Mehran University, Pakistan, in 2011. She successfully completed her PhD in Computer Engineering from Hanyang University, Korea in 2017. Her research interests are security and privacy in social network services, mobile social networks, cloud computing, mobile computing, and vehicle ad hoc networks. She has 12 years of teaching experience and currently working as Associate Prof. in Quest Pakistan. She has served as a reviewer in many conferences and journals. She is an author of many International and national papers.



Ubaidullah Rajput received his Bachelor's Degree in Computer System Engineering from Quaid-e-Awam University of Engineering, Science and Technology (Quest), Pakistan in 2005. He received his Master's in Computer System Engineering from NUST Islamabad, Pakistan in 2011. He successfully completed his PhD in Computer Engineering from Hanyang University, Korea. His research interests are security and privacy issues in crypto-currency, security and privacy issues in VANETS, Internet of Things (IoT), mobile social networks and cloud computing. He has more than 15 years of teaching and research experience and currently working as Associate Professor. in Quest Pakistan. He has served as a reviewer in many conferences and journals. He is author of many International and national papers.