Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук_____
(повна назва)

Кафедра _____Програмної інженерії_____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти – другий (магістерський)

Дослідження та аналіз методів відмінності та ідентифікації людини від програми у web застосунках
(тема)

Виконав: студент 2 курсу, групи ІПЗм-20-3___
_____Селіхов Г. Т._____
(прізвище, ініціали)

спеціальності 121 – Інженерія програмного забезпечення
(код і повна назва спеціальності)

_____Освітньо-наукова_____
(тип програми)

__Інженерія програмного забезпечення_____
(повна назва освітньої програми)

Керівник ___проф. Четвериков Г. Г._____
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри, проф. _____ З.В.Дудар

2022 р.

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет_Комп'ютерних наук_____

Кафедра Програмної інженерії_____

Рівень вищої освіти – другий (магістерський)

Спеціальність _121 – Інженерія програмного забезпечення_____
(код і повна назва)

Тип програми освітньо- наукова програма

Освітня програма_Інженерія програмного забезпечення_____

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«_____»_____ 20 ___ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові _____Селіхова Гліба Тарасовича___
(прізвище, ім'я, по батькові)

1. Тема роботи _Дослідження та аналіз методів відмінності та ідентифікації людини від програми у web застосунках затверджена наказом університету від "_24_" _березня_20 _22_ р. № _412 Ст__
заповнюється вручну після отримання наказу

2. Термін подання студентом роботи до екзаменаційної комісії _15 травня_ 2022 р.

3. Вихідні дані до роботи *технічне завдання, календарний план, методичні вказівки*_____

_____

4. Перелік питань, що потрібно опрацювати в роботі *мета роботи, аналіз проблемної галузі, постановка задачі, огляд технологій, аналіз існуючих методів та алгоритмів,*_____

_____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1.	Аналіз предметної галузі	15.02.2022	
2.	Огляд існуючих методів	12.03.2022	
3.	Аналіз існуючих методів та алгоритмів	09.04.2022	
4.	Підготовка пояснювальної записки	25.04.2022	
5.	Спецчастина	30.04.2022	
6.	Підготовка презентації та доповіді	03.05.2022	
7.	Попередній захист	09.05.2022	
8.	Нормоконтроль, рецензування	09.05.2022	
9.	Занесення диплома в електронний архів	09.05.2022	
10.	Допуск до захисту у зав. кафедри	19.05.2022	

Дата видачі завдання _____ _____ 2022 р.

Студент _____
(підпис)

Керівник роботи _____ __проф. Четвериков Г. Г._____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Кваліфікаційна робота магістра містить: 67 с., 22 рис., 29 джерел, 6 додатків.

ДОСЛІДЖЕННЯ, АНАЛІЗ, МЕТОДИ, МЕТОДИ ВІДМІННОСТІ, ІДЕНТИФІКАЦІЯ ЛЮДИНИ, ВЕБ ЗАСТОСУНКИ, ПРОГРАМА, WEB ЗАСТОСУНКИ.

Метою роботи є дослідження та аналіз методи відмінності та ідентифікації людини.

В роботі розглядаються методи відмінності та ідентифікації людини від програми у web застосунках.

В результаті роботи проведено дослідження та аналіз методів розпізнавання людей та ідентифікація програм у web застосунках.

RESEARCH, ANALYSIS, METHODS, DIFFERENCE METHODS, HUMAN IDENTIFICATION, WEB APPLICATIONS, PROGRAM, WEB APPLICATIONS.

The aim of the work is to study and analyze the methods of distinguishing and identifying a person.

The paper considers methods of distinguishing and identifying a person from the program in web applications.

As a result, research and analysis of methods of recognizing people and identifying programs in web applications.

Я, Селіхов Гліб Тарасович, студент групи ІПЗм-20-3, здобувач вищої освіти на другому (магістерському) рівні, кафедра Програмної інженерії, заявляю: моя кваліфікаційна робота на тему «Дослідження та аналіз методів відмінності та ідентифікації людини від програми у web застосунках», що буде представлена до ЕК для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу ElArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

Перелік умовних скорочень 8
Вступ 9
1 Аналіз проблемної галузі 10
1.1 CAPTCHA 10
1.2 Тест Тюрінгу 15
1.3 Інші засоби ідентифікації 16
2 Аналіз існуючих методів та алгоритмів 18
2.1 Існуючі методи ідентифікації людини від програми 18
2.2 Недоліки існуючих систем ідентифікації людини від програми у web застосунках 22
2.3 Необхідність ідентифікації людини від програми 24
3 Постановка задачі 27
4 Різновид перевірок для ідентифікації людини від програми у web застосунках 28
4.1 Види CAPTCHA 29
4.2 Капча Honeypot 37
5 Розробка системи ідентифікації людини від програми у web застосунках 38
5.1 Передумови системи ідентифікації роботів 38
5.2 Модель системи ідентифікації роботів 40
5.3 Рішення для ідентифікації роботів 43
5.4 Проблема ін’єкції JavaScript 45
6 Перспективи систем відмінності людини від програми у web застосунках 47
Висновки 50
Перелік джерел посилання 51
Додаток А Перелік джерел посилання за науковими напрямами керівника та науковців кафедри програмної інженерії 54
Додаток Б Звіт результатів перевірки на унікальність тексту 55

Додаток В Слайди презентації ……57
Додаток Г Апробація роботи…….....… 65
Додаток Д Експертний висновок результатів перевірки кваліфікаційної роботи на відповідність оформлення вимогам ДСТУ …….. 66

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ПЗ – програмне забезпечення

ШІ – штучний інтелект

WEB – World Wide Web

API – Application Programming Interface

IP – Internet Protocol

WAF – web application framework

BAAS – Bot as a Service

ML – Machine Learning

RD – Reachability distance

LRD – Local reachability density

LOF – Local Outlier Factor

CAPTCHA – Completely Automated Public Turing test to tell Computers and Humans Apart

ВСТУП

У перекладі англійська абревіатура CAPTCHA означає «цілком автоматизований публічний тест Тюрінгу для розрізнення комп'ютерів і людей». Спочатку тест Тюрінгу – експеримент, у якому судді пропонується визначити, хто із двох його співрозмовників людина, хто програма. Суддя, звичайно, не бачить і не чує їх, спілкування відбувається у вигляді листування. Якщо суддя за результатами бесіди, в якій використовується якнайбільше «людських», імовірно незрозумілих комп'ютеру питань, не може визначити, де хтось вважається, що штучний інтелект пройшов тест.

Капча – той самий тест у мініатюрі [1]. Щоб отримати доступ до якихось можливостей інтернет-ресурсу, користувач повинен вирішити завдання. Зазвичай дуже просту, але просту лише для людини. І нездійсненну (в ідеалі) для штучного інтелекту. Загалом, треба, щоб ніякі програми цей тест не пройшли, а будь-яка (в ідеалі, знову ж таки) жива людина, будь вона хоч неписьменною, хоч дитиною, пройшла без жодних проблем.

Проте боти чудово навчилися розшифровувати капчу. Першу reCAPTCHA з першого разу вирішували 98% ботів, а людей – лише 40. Саме тому у 2012 році було запущено другу версію системи, де використовувався метод розпізнання об'єктів на Street View. Полегшена версія цієї системи: користувачеві потрібно тільки натиснути на чекбокс, після чого система робила швидку перевірку і видавала остаточний вердикт: бот ви чи ні. За допомогою такої капчі система збирала всі cookies від Google, а разом з тим інформацію про натискання кнопок, дату, мову пристрою, а також про його браузери та плагіни.

Метою даної атестаційної роботи є дослідження та аналіз методів відмінності та ідентифікації людини від програми у web застосунках.

1 АНАЛІЗ ПРОБЛЕМНОЇ ГАЛУЗІ

1.1 CAPTCHA

Капча з'явилася 20 років тому і спочатку показувала високу ефективність у боротьбі з роботами. З розвитком штучного інтелекту метод почав старіти: тепер лише капчею не врятуватися навіть від спаму, не кажучи вже про більш серйозні проблеми. На даний момент існують різні технології та браузерні розширення, що дозволяють її оминати. Якщо завдання дуже складні для робота, їх виконання можна доручити реальним людям через спеціалізовані сервіси. Якщо дуже потрібно щось отримати, CAPTCHA – не перешкода.

Але важливо відзначити, що розробники капчі теж не сидять склавши руки та вдосконалюють технологію. Багато компаній виробляють капчі власними силами: наприклад, можна найняти спеціалістів на біржі для програмістів. Але більшість користується готовими рішеннями. Найчастіше використовується варіант – reCAPTCHA від Google.

У 2014 році пошуковик випустив удосконалену версію – reCAPTCHA v2. Її й досі активно використовують власники сайтів. Якщо користувач не викликає підозри у програми, йому потрібно лише поставити галочку для підтвердження своєї людяності. В іншому випадку з'являється інша задача, яка може завдати чимало клопоту реальному користувачеві, якщо програма помилиться (а це часто відбувається).

Так як ця версія капчі від Google все ж таки виявилася не такою зручною для користувачів, пошуковик запропонував удосконалений варіант – reCAPTCHA v3. Вона невидима для користувачів і не змушує їх вирішувати завдання. Натомість вона постійно відстежує їхні дії, дає їм оцінку, щоб потім винести вердикт, робот перед нею чи людину.

Капчу все ще використовують багато сайтів. Незважаючи на те, що її ефективність є сумнівною, цей метод боротьби з фейковим трафіком залишається простим і безкоштовним.

1.1.1 Технологія CAPTCHA

CAPTCHA – це тип тесту виклик-відповідь, який використовується в обчислювальній техніці для визначення того, чи є користувач людиною. Також дане поняття являє собою надуману абревіатуру від «Повністю автоматизований публічний тест Тюрінга, щоб розрізняти комп’ютери та людей»

Термін був введений у 2003 році Луїсом фон Аном, Мануелем Блумом, Ніколасом Дж. Хоппером та Джоном Ленгфордом [2]. Найпоширеніший тип CAPTCHA (відображається як версія 1.0) вперше був винайдений у 1997 році двома групами, які працювали паралельно. Ця форма CAPTCHA вимагає, щоб хтось правильно оцінив і ввів послідовність букв або цифр, які відображаються на спотвореному зображенні, що відображається на його екрані. Оскільки тест виконується комп’ютером, на відміну від стандартного тесту Тюрінга, який виконується людиною, CAPTCHA іноді описується як зворотний тест Тюрінга.

Ця процедура ідентифікації користувачів зазнала багато критики, особливо з боку людей з обмеженими можливостями, а також інших людей, які відчувають, що їхню повсякденну роботу гальмують спотворені слова, які важко прочитати. У середньому людині потрібно приблизно 10 секунд, щоб розгадати типову CAPTCHA.

CAPTCHA, за визначенням, повністю автоматизовані, вимагають незначного обслуговування або втручання людини для адміністрування, що дає переваги у вартості та надійності.

Алгоритм, який використовується для створення CAPTCHA, має бути оприлюдненим, хоча він може бути покритий патентом. Це робиться для того, щоб продемонструвати, що для його зламу потрібне рішення складної проблеми в області штучного інтелекту, а не просто відкриття (секретного) алгоритму, який можна отримати за допомогою зворотної інженерії або іншим способом.

Сучасні текстові CAPTCHA розроблені таким чином, що вони вимагають одночасного використання трьох окремих здібностей – інваріантного

розпізнавання, сегментації та розбору – для правильного виконання завдання з будь-якою послідовністю:

– інваріантне розпізнавання відноситься до здатності розпізнавати велику кількість варіацій у формі букв. Існує надзвичайно велика кількість версій кожного символу, які людський мозок може успішно ідентифікувати. Те ж саме не стосується комп'ютера, і навчити його розпізнавати всі ці різні утворення є складним завданням;

– сегментація або можливість відокремлювати одну літеру від іншої також ускладнена в CAPTCHA, оскільки символи скупчені разом без пробілів між ними;

– контекст також важливий. Щоб правильно ідентифікувати кожен символ, CAPTCHA потрібно розуміти цілісно. Наприклад, в одному сегменті CAPTCHA літера може виглядати як «m». Тільки коли все слово враховано в контексті, стає зрозуміло, що це a і n;

– кожна з цих проблем становить значну проблему для комп'ютера, навіть ізольовано. Наявність усіх трьох одночасно – це те, що ускладнює розв'язання CAPTCHA.

На відміну від комп'ютерів, люди відмінно справляються з цим типом завдань. Хоча сегментація та розпізнавання є двома різними процесами, необхідними для розуміння зображення для комп'ютера, вони є частиною одного процесу для людини. Наприклад, коли людина розуміє, що перша літера CAPTCHA є a, ця особа також розуміє, де знаходяться контури цієї букви, а також де вона зливається з контурами наступної літери. Крім того, людський мозок здатний до динамічного мислення на основі контексту. Він здатний підтримувати в живих кілька пояснень, а потім вибирати те, яке є найкращим поясненням для всього введення на основі контекстних підказок. Це також означає, що його не обдурять варіанти літер.

1.1.2 Технологія reCAPTCHA

reCAPTCHA – це система CAPTCHA, яка дає змогу веб-хостам розрізняти людський і автоматичний доступ до веб-сайтів [3]. Оригінальна версія пропонувала користувачам розшифрувати важкий до читання текст або зіставити зображення. Друга версія також запропонувала користувачам розшифрувати текст або зіставити зображення, якщо аналіз файлів cookie та рендеринг полотна показав, що сторінка завантажується автоматично. Починаючи з версії 3, reCAPTCHA ніколи не перериває користувачів і призначений для автоматичного запуску, коли користувачі завантажують сторінки або натискають кнопки.

Початкова версія сервісу була платформою для масової співпраці, розробленою для оцифрування книг, особливо тих, які були занадто нерозбірливі для сканування комп'ютером. Перевірка підказує використані пари слів із відсканованих сторінок, причому одне відоме слово використовується як контроль для перевірки, а друге використовується для читання невизначеного слова. reCAPTCHA спочатку була розроблена Луїсом фон Ан, Девідом Абрахамом, Мануелем Блумом, Майклом Кроуфордом, Беном Маурером, Коліном Макмілленом та Едісоном Таном в головному кампусі Університету Карнегі-Меллона в Піттсбурзі. Його придбала Google у вересні 2009 року. Система допомогла оцифрувати архіви The New York Times, і згодом була використана Google Books для подібних цілей.

Повідомлялося, що система щодня відображає понад 100 мільйонів CAPTCHA на таких сайтах, як Facebook, TicketMaster, Twitter, 4chan, CNN.com, StumbleUpon, Craigslist (з червня 2008 року) і в США. Веб-сайт програми купонів для конвертерів цифрового телебачення Національного управління телекомунікацій та інформації (як частина переходу США на DTV).

У 2014 році Google відкинув службу від її початкової концепції, зосередившись на зменшенні кількості взаємодії з користувачем, необхідної для підтвердження користувача, і представляючи проблеми розпізнавання людини (наприклад, визначення зображень у наборі, які задовольняють певну підказку), якщо поведінковий аналіз підозрює, що користувач може бути ботом.

Розподілені коректори були першим проектом, який добровольцем присвятив свій час розшифровці відсканованого тексту, який не можна було прочитати програмами оптичного розпізнавання символів (OCR). Він співпрацює з Project Gutenberg для оцифровки матеріалів, які є загальнодоступними, і використовує методи, що сильно відрізняються від reCAPTCHA.

Програма reCAPTCHA створена з гватемальського вченого-комп'ютерщика Луїса фон Ана, а за підтримки стипендії Макартура. Ранній розробник CAPTCHA зрозумів, що «він мимоволі створив систему, яка витрачала на десять секунд мільйони годин найціннішого ресурсу: циклів людського мозку».

1.1.3 Технологія No-CAPTCHA

Після попередніх версій капчі (що описані вище) корпорація Google представила нову систему захисту від спамерів та ботів, яка відрізняється від того, що було розглянути раніше.

Ця технологія двофакторної (хоча скоріше трифакторної) оцінки якості користувача.

Спеціальний скрипт оцінює поведінковий фактор, і якщо галочку ставить людина, то вона може перейти до користування сайтом. Роботи діють по-іншому, тому навіть якщо спамерський скрипт спробує поставити галочку, No-CAPTCHA зрозуміє, що це не людина.

Проте одразу така людина не буде викинута з сайту, бо у Google розуміють, що трапляються помилки. Тому якщо перший етап не пройдено, No-CAPTCHA запропонує пройти другий етап перевірки. Тут також можливі варіанти. Перший з них – звична капча. Але Google тестує і оновлений варіант, більш дружній для користувача.

У такому разі пропонується вибрати зображення, наприклад, тварини з низки фотографій.

Що стосується поведінкових факторів, відомо, що використовують інформацію про час, проведений на сторінці та IP користувача. Інші критерії оцінки не розкриваються, оскільки спамери швидко створять робота, що обманює No-CAPTCHA.

1.2 Тест Тюрінга

Неможливо, піднімаючи питання розпізнавання людини від машини, не підняти питання машині Тюрінгу [4].

У 1950 році Алан Тюрінг сформулював те, що пізніше було відоме як тест Тюрінга, ідею про те, як можна визначити, чи буде комп'ютер, тобто машина, мати здатність мислення, еквівалентну людській. Він сам спочатку назвав цю тестову імітаційну гру, яка спочатку була лише теоретичним начерком. Це було сформульовано більш точно і конкретно пізніше (тобто після самогубства Тюрінга в 1954 році), після того, як штучний інтелект став самостійним академічним предметом як частиною інформатики. Відтоді цей тест був у всіх на вустах у дискусії про штучний інтелект і неодноразово служив для відродження міфу про мислячу машину для комп'ютерної епохи.

Під час цього тесту людина, яка запитує, веде розмову з двома невідомими співрозмовниками за допомогою клавіатури та екрана без візуального чи слухового контакту. Один співрозмовник – людина, інший – машина. Якщо після інтенсивного опитування запитувач не може сказати, хто з двох є машиною, машина пройшла тест Тюрінга, і вважається, що машина має здатність мислення, рівну людській.

Було висунуто ряд аргументів, які захищають думку, що тест Тюрінга невідповідний для вимірювання інтелекту:

– тест Тюрінга перевіряє лише функціональність, а не наявність навмисності чи свідомості. Цей аргумент був розроблений Джоном Сірлом, серед

інших, у своєму розумовому експерименті «Китайська кімната». Тюрінг уже знав про цю проблему, коли формулював свій тест, але вважав, що його також можна використовувати як доказ свідомості. Серл, з іншого боку, відкидає це;

– тест Тюрінга «в першу чергу стосується обману». Він перевіряє «людську довірливість, а не справжній штучний інтелект». Виноградський виклик краще перевіряє «здоровий глузд» і «знання реальності».

Тюрінг припустив, що до 2000 року можна буде запрограмувати комп'ютери таким чином, щоб середній користувач мав не більше 70 відсотків шансів успішно ідентифікувати людину і машину після «розмови» з ними протягом п'яти хвилин. Те, що це передбачення досі не здійснилося, багато хто вважає доказом того, що складність природного інтелекту недооцінена.

1.3 Інші засоби ідентифікації

Якщо вже мова зайшла про інші засоби ідентифікації людини від програми, можна коротко розглянути, що можна використовувати на сайті крім або замість капчі.

Для початку необхідно визначити, проти чого необхідно боротися.

Якщо сайт страждає від брутфорс-атак (таке найчастіше відбувається з сайтами на безкоштовних CMS), є сенс встановити фаєрвол, використовувати двофакторну автентифікацію, змінити URL-адресу сторінки входу і так далі. Варіантів у цьому випадку може бути багато.

Якщо проблема в отриманні тонни спаму в коментарях та контактних формах, можна використовувати різні плагіни, які за допомогою спеціальних фільтрів відсівають спам.

Боротися зі спамом у формах можна по-іншому. Наприклад, задавати полям унікальні назви, до яких боти не звикли, або вбудовувати додаткові невидимі для

користувачів поля. Останній метод роботи обходити вже навчилися, тому його ефективність перебуває під сумнівом.

Якщо йдеться про інтернет-магазини, тут не обходиться без реєстрації з підтвердженням e-mail або телефонного номера. Реєстрація електронної скриньки для ботів не така велика проблема, а телефонний номер для них отримати більш проблематично. Авторизація за допомогою соцмереж теж допомагає відсіяти фейки при реєстрації.

Нарешті, ботів можна блокувати у файлі .htaccess або фільтрувати за допомогою спеціалізованих сервісів.

2 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ТА АЛГОРИТМІВ

2.1 Існуючі методи ідентифікації людини від програми

Мабуть, немає людини, які б не стикалася з проханням Google довести, що вона реальна людина. Все частіше за простою кнопкою «Я не робот» почали з'являтися вимоги довести це – обравши всі світлофори, переходи чи вітрини у сітці зображень. Незабаром світлофори почали ховатися в листі, переходи спотворюватися і йти за кут, а вивіски магазинів стали розмитими і перейшли на корейську мову.

Ці тести називаються CAPTCHA – акронім від «повністю автоматичного публічного тесту Тюрінгу, призначеного для розрізнення людей і комп'ютерів», і колись вони вже доходили до такої міри нерозбірливості. На початку 2000-х простих зображень із текстом було достатньо, щоб зупинити більшість спам-ботів. Пройшло десять років, і після того, як компанія Google купила програму у дослідників з Університету Карнегі-Меллона і використовувала її для оцифровки в проекті Google Books, тексти доводилося все сильніше спотворювати і приховувати, щоб обганяти оптичні розпізнавання символів, що поліпшуються, ті самі програми, які допомагали покращувати ті самі люди, кому доводилося розгадувати всі ці капчі.

Оскільки CAPTCHA –інструмент для тренувань штучного інтелекту, будь-який придуманий тест може протриматися лише деякий час, що визнають і його винахідники. З усіма цими дослідниками, шахраями, і найпростішими людьми, вирішальними мільярди завдань межі можливого для штучного інтелекту, рано чи пізно машини просто мали нас обігнати. У 2014-му Google стравила між собою свій найкращий алгоритм з розгадування найбільш спотворених текстів і людей: комп'ютер правильно розпізнав текст у 99,8% випадків, а люди лише у 33%.

Після цього Google перейшов на NoCaptcha ReCaptcha, що спостерігає за поведінкою людей і збирає їх дані, що дозволяє деяким з них пройти далі просто

натисканням кнопки «Я не робот», а іншим видає завдання на пошук зображень, які можна сьогодні спостерігати. Але машини знову наздоганяють людей. Всі навіси, які можуть бути чи не бути вітринами магазинів – це остання стадія перегонів озброєнь людей та машин.

Язон Полакіс, професор інформатики в Університеті Іллінойсу в Чикаго, особисто відповідає за недавнє ускладнення капчі. У 2016 році він опублікував роботу, в якій використовував готові програми розпізнавання зображень, включаючи власний пошук за зображеннями від Google, щоб вирішувати капчі з точністю в 70%. Інші дослідники справлялися з розпізнаванням аудіо капчі від Google за допомогою програм з розпізнавання мови від самої компанії.

Машинне навчання вже не гірше за людей справляється з розпізнаванням простих текстів, зображень та голосу. Алгоритми, можливо, навіть роблять це вже краще: «Ми дійшли до того моменту, коли ускладнення завдань для софту призводить до того, що завдання стають надмірно складними для людей. Нам потрібна альтернатива, але чіткого плану поки що немає» [5].

Література по капчах переповнена фальстартами та дивними спробами знайти щось окрім тексту та зображень, з чим добре справляються всі люди та погано справляються машини. Дослідники намагалися пропонувати користувачам сортувати зображення людей за виразом осіб, підлоги та етнічної приналежності. Були пропозиції організувати капчі з вікторинами, капчі на основі колискових, поширених у тих місцях, де, як передбачається, виріс користувач. Такі капчі з культурною прив'язкою націлені як на роботів, а й у людей інших країн, що вирішують капчі за копійки. Люди намагалися загнати алгоритми розпізнавання зображень у глухий кут, пропонуючи користувачеві впізнати, наприклад, свиню, але при цьому мальовану і в сонячних окулярах. Дослідники вивчали такі варіанти, як запропонувати користувачам розпізнати об'єкти у мішанині калейдоскопа. В одному з найцікавіших варіантів у 2010-му дослідники запропонували використовувати капчу для сортування стародавніх петрогліфів – комп'ютери погано справляються з розпізнаванням скетчів або зображень оленів на стінах печер.

Нещодавно були спроби розробити ігрові капчі, де користувачеві потрібно обертати об'єкти на певні кути або пересувати шматочки головоломки, причому інструкції за рішенням капчі давалися не у вигляді тексту, а у вигляді символів, або малися на увазі по контексту ігрового поля. Надія була на те, що люди зрозуміють логіку загадки, а комп'ютери без чітких інструкцій спіткнуться. Інші дослідники намагалися використати факт наявності у людей тіл, та застосовували камери пристроїв або доповнену реальність для інтерактивного підтвердження наявності людини.

З багатьма з цих тестів проблема не в тому, що роботи дуже розумні, а в тому, що люди погано з ними справляються. І справа не в тому, що людям бракує знань, вони просто дуже сильно різняться за мовою, культурою та досвідом. Позбувшись всього цього, щоб зробити тест, який може пройти будь-яка людина без тренування та довгих роздумів, ми залишаємося з такими грубими завданнями, як розпізнавання зображень – а це саме те, з чим добре впорається спеціально заточений під це штучний інтелект.

«Тести обмежені можливостями людини, – каже Полакіс. – Справа не лише у фізичних можливостях – треба знайти щось міжкультурне, міжмовне. Потрібне завдання, яке добре працює з людиною з Греції, з людиною з Чикаго, з людиною з Південної Африки, Ірану та Австралії одночасно. І вона не повинна залежати від культурних нюансів та відмінностей. Потрібне завдання, з яким добре справляється середня людина, вона не повинна бути обмежена певною підгрупою людей, і вона повинна бути складною для комп'ютера. Все це обмежує вибір варіантів. А ще це має бути щось, із чим люди справляються швидко, і що не дуже дратує» [6].

Спроби вирішення цих загадок з розмитими картинками швидко переводять людину на філософські рейки: чи є якась універсальна людська якість, яку можна продемонструвати машині, і яку машина не може імітувати? Що означає бути людиною?

Може, людяність вимірюється не тим, як люди виконують завдання, а тим, як вони поводяться, просуваючись крізь світ – або, в даному випадку, крізь

інтернет. Ігрові капчі, відеокапчі, будь-які капчі, які можна придумати, в результаті будуть зламані, каже Шуман Госмахумдер, який займався в Google боротьбою з автоматизацією кліків, а потім технологічним директором компанії з розпізнавання роботів Shape Security. Він схиляється у бік «постійної авторизації» замість окремих тестів – до нагляду за поведінкою користувача та пошуку ознак автоматизації. «Реальна людина не дуже добре контролює моторику, і не може рухати мишу однаково багато разів під час кількох взаємодій, навіть якщо намагатиметься зробити це», — говорить Госмахумдер. Робот взаємодіятиме зі сторінкою, не рухаючи мишею, або рухаючи її дуже точно, а в діях людини спостерігатиметься «ентропія», яку складно підробити, каже Держмахумдер.

Власна команда Google, що займається капчею, працює у подібному напрямку. Версія reCaptcha v3 використовує "адаптивний аналіз ризиків" для оцінки трафіку за підозрілістю; власники сайтів можуть пропонувати підозрілим користувачам завдання на кшталт введення пароля або двофакторної авторизації. У Google не повідомляють, які фактори враховуються при оцінках, крім того, що компанія оцінює, як виглядає на сайті «хороший трафік», та використовує цю інформацію для фільтрації «поганого трафіку», згідно Сай Кормаї, менеджеру продукту з команди CAPTCHA. Дослідники в галузі безпеки кажуть, що це, ймовірно, суміш куків, атрибутів браузера, закономірностей трафіку та інших факторів. Один недолік нової моделі розпізнавання роботів полягає в тому, що навігація в Інтернеті при спробах мінімізації спостережень за користувачем може стати трохи дратівливою, оскільки такі речі, як VPN і розширення, що ускладнюють відстеження користувача, можуть відзначити користувача як підозрілого.

Аарон Маленфант, провідний інженер команди CAPTCHA в Google, каже, що зрушення у бік від тестів Тюрінга має допомогти обійти змагання, яке люди постійно програють. «Чим більше ми вкладатимемося в машинне навчання, тим складніше ці завдання ставатимуть для людей, і, зокрема, тому ми запустили CAPTCHA V3 – щоб випередити цю криву» [7]. Маленфант каже, що через 5-10

років завдання у капчі взагалі не матимуть сенсу. Більшість Інтернету залежатиме від постійного прихованого тесту Тьюринга, що працює на тлі.

У своїй книзі «Найлюдяніша людина» Брайан Крістіан бере участь у тесті Тюрінга і розуміє, що дуже складно довести свою людяність у бесідах. З іншого боку розробники роботів виявили, що ці тести легко пройти, не вдаючи красномовним або інтелектуальним співрозмовником, а відповідаючи на запитання за допомогою нелогічних жартів, роблячи друкарські помилки, або, як у випадку бота, що виграв змагання Тюрінга в 2014-му, заявляючи, що ти – 13-річний український хлопчик, який погано розмовляє англійською. Адже людині властиво помилятися. Можливо, що таке майбутнє чекає і капчу, найпоширеніший тест Тюрінга у світі – нова гонка озброєнь буде створювати не роботів, що перевершують людей у сортуванні зображень і розборі тексту, а роботів, які роблять помилки, що промахуються по кнопках, що відволікаються і перемикають вкладки.

Капчі можуть зберегтися і в цьому світі. У 2017-му Amazon зареєструвала патент на схему, в якій використовуються оптичні ілюзії та логічні завдання, з якими важко справлятися людям. Цей тест називається «тест Тюрінга через помилку», і єдиний спосіб пройти його – дати неправильну відповідь [8].

2.2 Недоліки існуючих систем ідентифікації людини від програми у web застосунках

Користувачі в мережі не полюбляють капчу. І на це є певні причини. Нижче перераховано основні з них.

Капча забирає багато часу. Так, кожен тест забирає всього кілька секунд. Але, якщо порахувати, скільки головоломок доводиться вирішувати за тиждень або місяць, то виходить велика цифра. До того ж, часто система змушує

підтвердити свою особистість два або три рази поспіль, щоб на всі 100% переконатися, що на сайті дійсно людина.

Капчі бувають надто складними. Часом буває дуже важко розрізнити цифри та літери – особливо, якщо символи надто сильно спотворені. Можна переплутати букву «О» та цифру «0». Крім того, розробники постійно збільшують складність тестів, оскільки зловмисники вдосконалюють роботів.

Наприклад, на рисунку 2.1 приведено зображення капчі, яку пропонується розібрати користувачу.

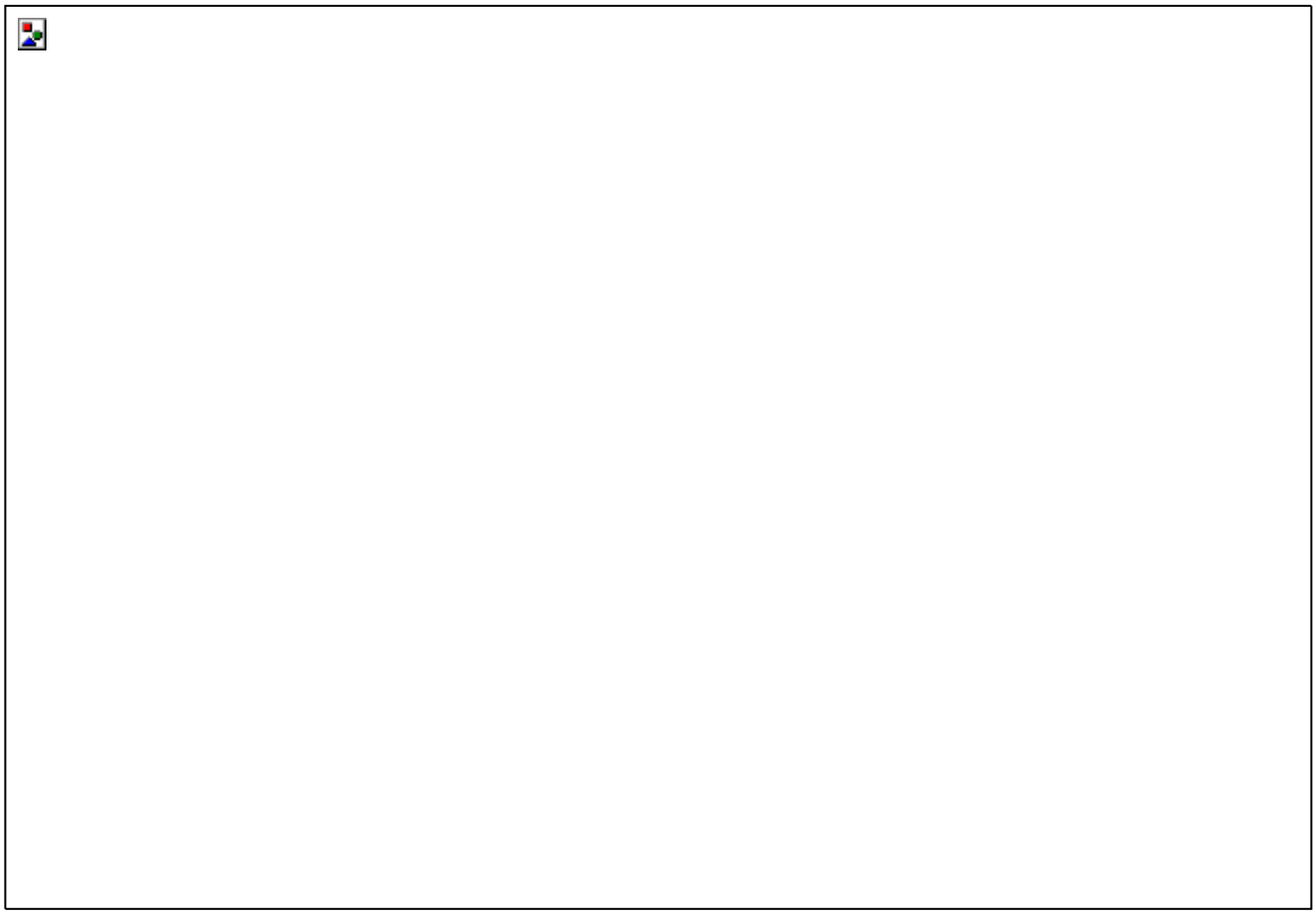


Рисунок 2.1 – Приклад занадто складної капчі

Капча заважає користувачам із особливостями [9]. Вона часто буває надто складною для людей, які мають проблеми із зором. Деякі системи включають можливість озвучувати символи, проте це не завжди рятує.

Капча псує інтерфейс сервісу. Часто спотворені шрифти та різні колірні рішення не вписуються у загальну форму інтернет-порталу, через що порушується загальний візуал.

Капча навіть може знижувати відвідуваність. Побачивши черговий тест, користувачі можуть залишити сайт, щоб зберегти свої нерви. Аудиторія старша за 40, як показують дослідження, взагалі рідко розуміє, навіщо потрібні якісь перевірки [10]. Як результат, веб-сайти втрачають потенційних клієнтів. Чим складніший захист, тим менше відвідувачів.

2.3. Необхідність ідентифікації людини від програми

Існують програми, які можуть автоматично виконувати дії в інтернеті. Наприклад, розсилати рекламу, публікувати фальшиві відгуки. І ось, щоб їх заблокувати, власники сайтів ставлять капчу.

Наприклад, є сторінка в інтернеті, де відвідувачі залишають відгуки. Будь-яка людина може заповнити кілька полів (ім'я, адресу пошти) та надіслати на сайт своє повідомлення, де воно буде опубліковано.

Ці дії легко зімітувати програмою. Її можна «навчити» заповнювати поля та публікувати потрібні відгуки, чим і користуються шахраї [11]. Тому і була придумана капча – адже її програма не запровадить, а отже, не зможе опублікувати повідомлення.

Але капча з'являється і в інших випадках. Наприклад, вона може виникати коли люди просто спілкуються в інтернету. В таких випадках на сайті не можна буде виконати більше жодної дії, поки людина не пройде капчу.

Це відбувається тому, що системі здалося щось у діях користувача підозрілим. Наприклад, він надто швидко відповідав на повідомлення, і сайт запідозрив, що це робить не людина, а програма.

Тому доводиться вводити у віконце те, що намальовано на картинці. Воно зазвичай не дуже зручне, проте захищає сторінки від злому.

Слід зазначити, що застосування капчі все одно не рятує сайти від реклами, зломів та інших речей. Але все-таки позбавляє сервіси зайвих наполегливих спроб злому.

Часто для зручності у капч є кнопка, натиснувши яку варіант капчі зміниться. Це потрібно, якщо літери складно розібрати. Або коли попри правильно введені дані система не пропускає далі та видає помилку.

Також на текстових капчах зачасту буває кнопка, що озвучує те, що написано [12]. Натиснувши на неї, користувач почує голос, який промовить літери/цифри з зображення.

Капча потрібна насамперед власникам сайту – для простих користувачів вона лише створює зайві проблеми. Захисна програма ставиться, щоб відсіяти ботів, які хочуть потрапити на сайт для різних цілей. Так наприклад капча необхідна для захисту від наступних проблем:

– захист від спаму – боти можуть розсилати рекламу, залишати спам-коментарі та відгуки. Якщо мова йде про великий сайт і його володар не бажає вичищати все вручну, капча стає в нагоді;

– захист від DDoS-атак – коли на сайт одночасно надсилається безліч запитів, які сервер фізично не може обробити, то сайт впаде [13]. Капча може стримати цей потік;

– захист від брутфорсингу, або підбору логінів та паролів [14]. Людині рано чи пізно набридне підбирати логіни та паролі, щоб зламати сайт. А робот може робити це нескінченно і рано чи пізно згенерувати потрібні, якщо його вчасно не зупинити за допомогою капчі;

– захист від перехоплення товарів у інтернет-магазинах. Таке часто практикується в період розпродажів та акцій на сайті: боти імітують дії користувачів, «скуповують» усі товари, перехоплюючи їх у реальних користувачів;

– захист від парсингу даних [15]. Проте в цьому випадку капча не завжди добре працює – з'явилися хороші послуги парсингу. Для сайту критичної шкоди від парсингу немає, але конкуренти отримають важливу і не призначену для їх очей інформацію.

Проте володар сайту завжди повинен пам’ятати, що це незручно для користувача. Незручно, коли доводиться вдивлятися в нерозбірливий набір літер та цифр, а при помилці починати спочатку; що помилитися просто: складно відрізнити букву «О» від цифри «0»; що капча не може запам'ятати, що користувач не робот і з'являється знову і знову.

Те, що незручно для користувача, неминуче відбивається і на власника сайту. Введення капчі – це завжди зайва дія, а взаємодія користувача з сайтом має зводитися до мінімуму кліків. Відповідно, капча на сайті погіршує юзабіліті, а заразом і конверсію.

Наочним прикладом буде сайт інтернет-магазин, де користувач вже обрав товар, додав його в кошик, та готовий оплатити, проте перед оплатою він отримує капчу. Якщо капча буде занадто складною, користувач може не дійти до кінця, та назавжди покинути сайт.

3 ПОСТАНОВКА ЗАДАЧІ

Метою атестаційної роботи є дослідження та аналіз методів відмінності та ідентифікаціх людини від програми у web застосунках.

В ході роботи слід розглянути наявні методи ідентифікації, засновані на тесту Тюрінгу, такі як:

– технологія CAPTCHA;

– технологія reCAPTCHA;

– технологія NoCAPTCHA.

Атестаційна робота повинна містити дані щодо наявних видів ідентифікації людини від програми. Слід розглянути переваги та недоліки технологій, загальну необхідність ідентифікації людини у web застосунках.

В заключенні роботи слід переглянути аналоги технології CAPTCHA, та перспективи даної технології.

Атестаційна роботи проводить дослідження методів ідентифікації та відмінності людини від програми у web застосунках, та містить зроблені в ході виконання роботи висновки.

4 РІЗНОВИД ПЕРЕВІРОК ДЛЯ ІДЕНТИФІКАЦІЇ ЛЮДИНИ ВІД ПРОГРАМИ У WEB ЗАСТОСУНКАХ

В двадцять першому сторіччі користувачу щоразу при створенні нового облікового запису в інтернеті користувачеві доводиться доводити, що він не робот. Цей процес заснований на так званій Капчі – повністю автоматизованому загальнодоступному тесті Тюрінга, який визначає, хто перед ним: людина або робот. Цю технологію веб-сервіси широко використовують для захисту від зломів та спаму з боку роботів. Капча – спрощений варіант абревіатури CAPTCHA (Completely Automated Public Turing Test to Tell Computers and Humans Apart), яку вигадали між 1997 і 2000 роками.

Вперше капчу для захисту від спам-запитів, які отримують кожен день, використовували в неіснуючій пошуковій системі AltaVista [16]. Термін командою розробників з Університету Карнегі-Меллона був офіційно введений 2000-го. У цей час користувачі вперше побачили набридливі спливаючі вікна з проханням підтвердити, що вони не є роботами. З кожним роком капчі ставали дедалі більш спотвореними та важкими для читання – усьому виною зростання «кмітливості» ботів, яких вчили розшифровувати їх.

Капча пропонує тест, який легко вирішить людина, але не зможе розпізнати машина. На екрані з'являється головоломка у вигляді зашифрованого тексту, який потрібно ввести у відповідне поле або щось витонченіше. Капча є незамінною для компаній – особливо для тих, чия діяльність пов'язана з фінансовими операціями. Наприклад, коли авіакомпанії відкриваються продаж квитків, технологія відсікає ботів, які миттєво скуповують усі місця.

Капча повинна знизити можливість присутності на сайті роботів. Для цього сервіс аналізує ризики і часто може вибрати один із наведених вище типів тесту. Але на конкретній сторінці може використовуватися і одне конкретне завдання [17].

4.1 Види CAPTCHA

4.1.1 Розпізнавання тексту

Традиційний тип капчі, який вимагає від користувача ввести ряд цифр та букв [18]. Найчастіше текстовий рядок спотворений різними кольоровими та шумовими фільтрами, а символи в ньому перекреслені або мають нахил (див. рис. 4.1).

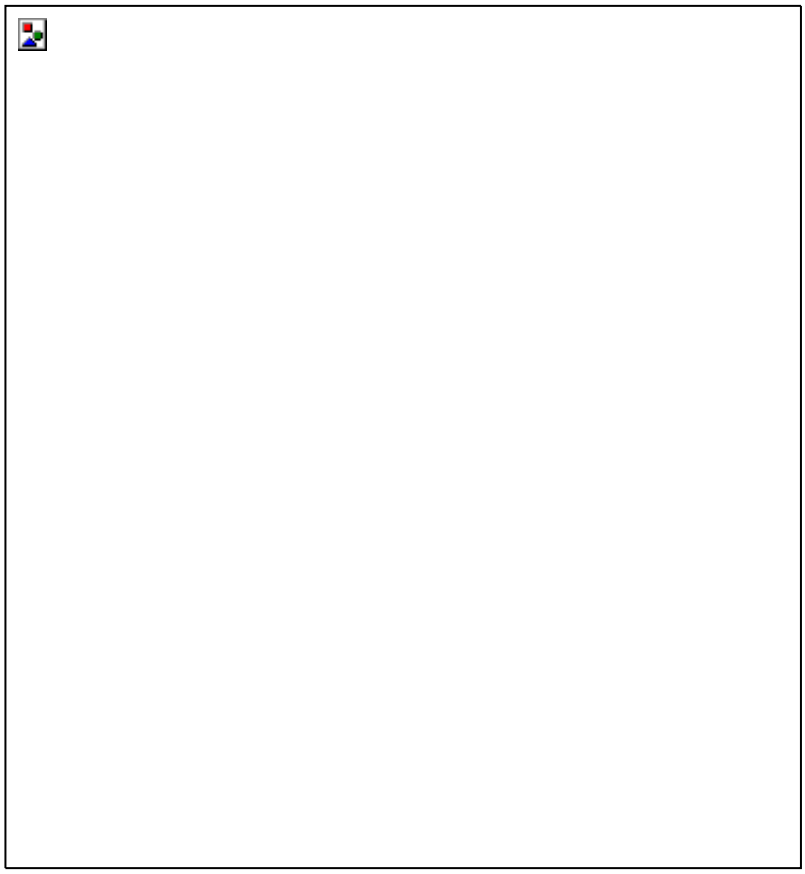


Рисунок 4.1 – Приклад текстової капчі

На рисунку вище приведено розповсюджений варіант даного методу капчі, який можна зустріти на багатьох сервісах.

4.1.2 Вибір зображень

Капча, що вимагає від користувача вибрати потрібне зображення, спонукає користувачів ідентифікувати набір фотографій (наприклад, вказати всі зображення з гідрантами). Її приклад приведено на рисунку 4.2.

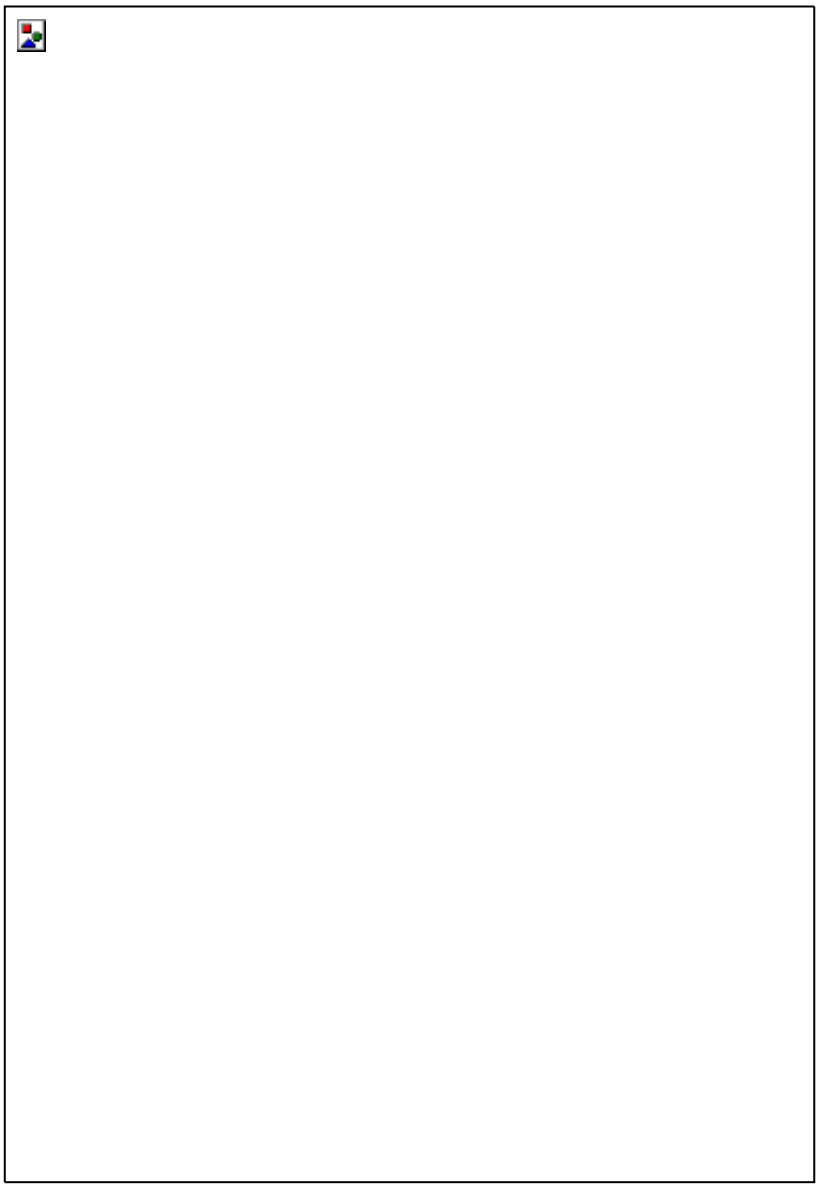


Рисунок 4.2 – Приклад капчі з вибором зображень

Зазвичай картинки захищені від розпізнавання ботами з допомогою спеціального шуму, що вони може відсіяти (див. рис. 4.3).

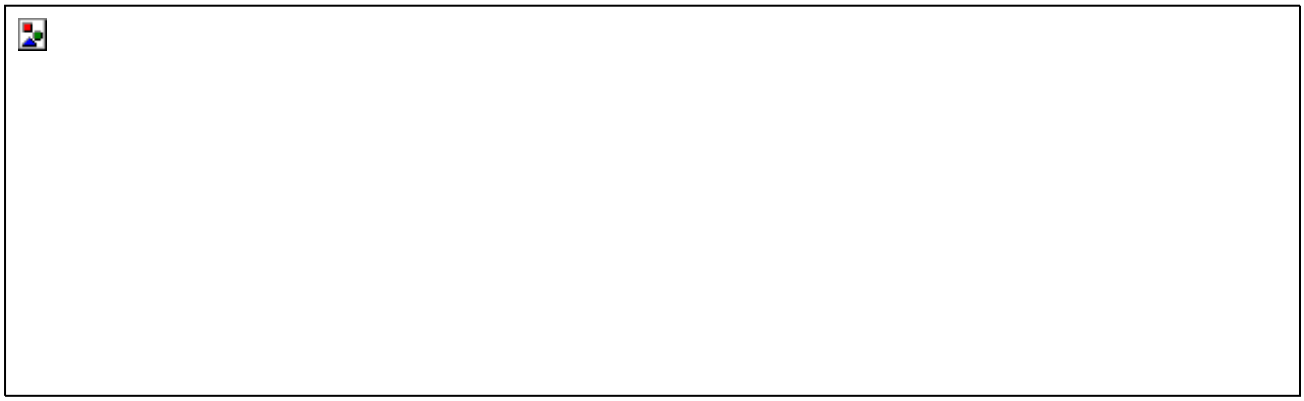


Рисунок 4.3 – Приклад капчі з вибором зображень із шумом

Так на рисунку вище можна побачити накладання шуму поверх приведеної картинки.

Так ми маємо картинку зліва в її вигляді, на якій зображена риба, що розуміє людина, а також розуміє бот. Посередині приведено фрагмент шуму, який надалі накладається на базове зображення. В результаті отримаємо третю картинку (справа). Так на ній бот вже не може побачити рибу, та йому здається, що він бачить зображення краба [19].

Після цих дій з зображенням бот вже не може розпізнати зображення, та не розуміє що саме приведено на рисунку.

4.1.3 Логічна задача

Варіант капчі з логічної задачею перевіряє, чи може умовний користувач перед нею думати. Головоломки можуть бути різними:

– система просить користувача вирішити математичний приклад – як варіант, скласти або відняти пару чисел [20];

– капча просить вибрати певний об'єкт – наприклад, знайти фотографію, на якій зображена людина з піднятою рукою;

– завдання вибрати одну цифру з набору – як варіант, потрібно написати другу чи третю цифру з числа 1741505;

– потрібно вибрати слово, яке починається з конкретної літери.

Приклад даної капчі можна побачити на рисунку 4.4.

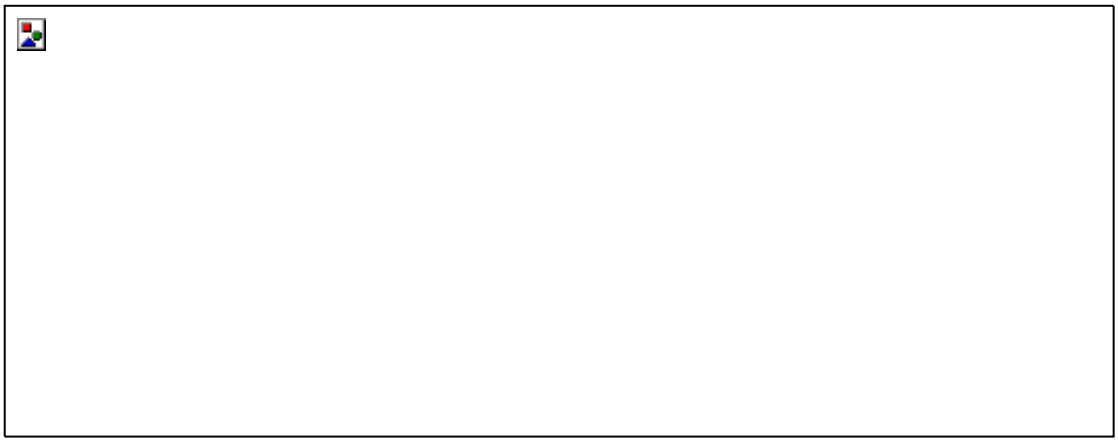


Рисунок 4.4 – Приклад капчі, що вимагає від користувача вирішення логічної задачі

Так як боти здатні розпізнавати цифри на зображенні, дані капчі в своїй більшості перекриваються лініями, містять не точно зображені цифри, або ж «розмиваються» задля перешкоджання розпізнавання символів ботами, проте в такому ступіні, щоб звичайний користувач все ще міг розпізнати приведене зображення.

4.1.4 Тривимірна капча

Наступний приклад капчі – тривимірна (див. рис. 4.5). Вона вимагає від користувача ідентифікував зображення, літери або числа, які відображаються у трьох вимірах. У принципі, це ускладнений варіант будь-якого з попередніх типів.

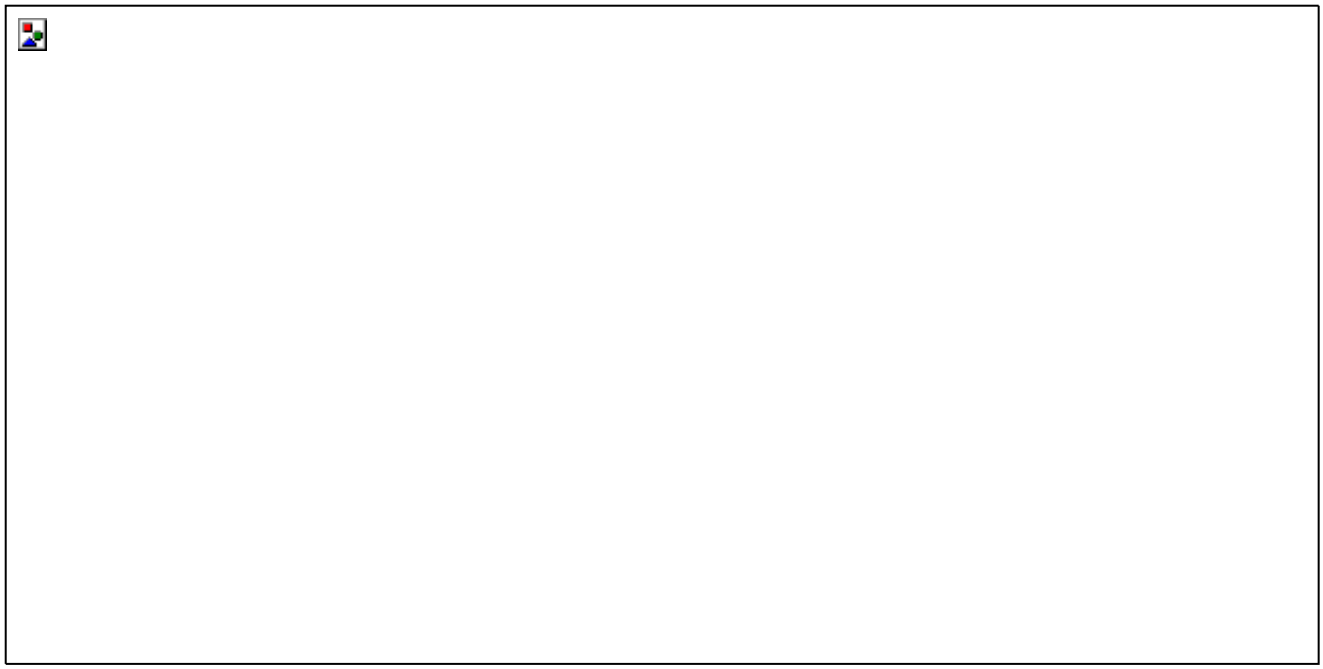


Рисунок 4.5 – Приклад тривимірної капчі

Так на рисунку 4.5 приведено варіант тривимірної капчі. В даному випадку від користувача вимагається ідентифікувати напис, який буде дуже важко розпізнати боту.

4.1.5 Маркетингова капча

Маркетингова капча просить користувача ввести слово чи фразу, що відповідає певному бренду (див. рис. 4.6).

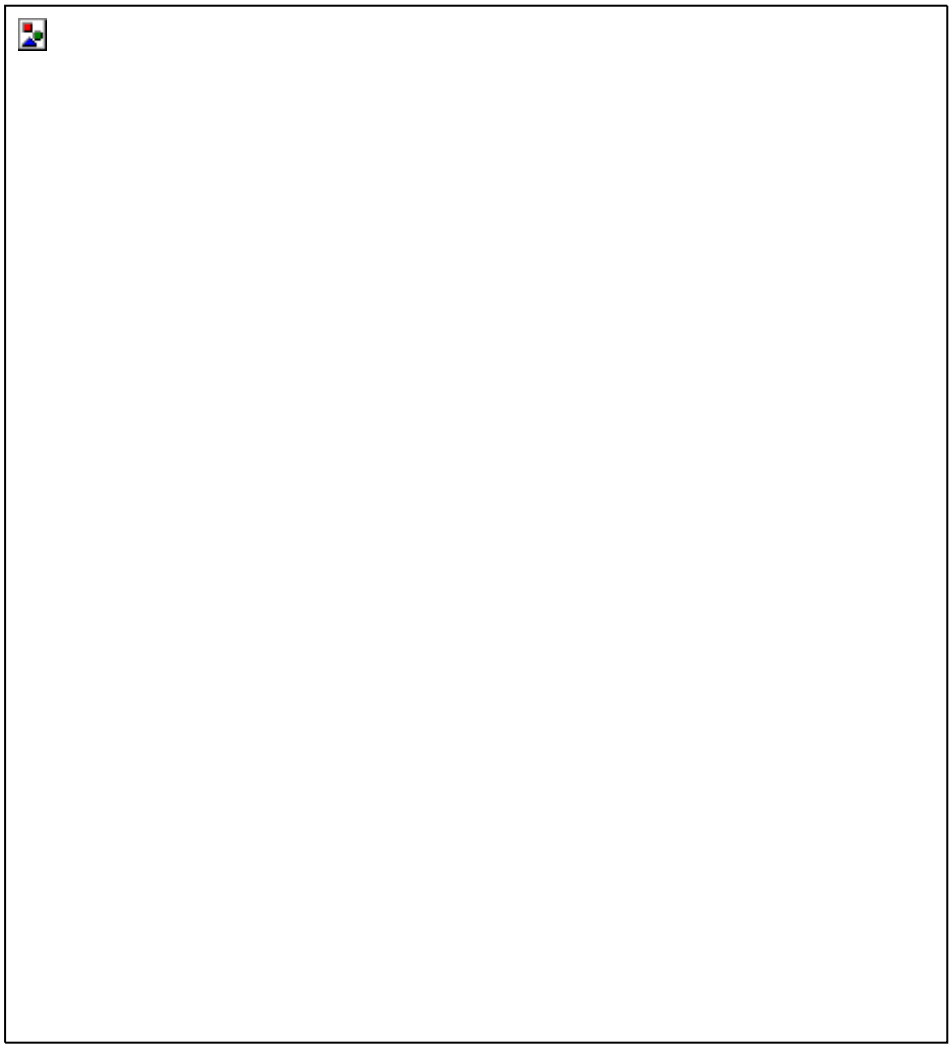


Рисунок 4.6 – Приклад меркетингової капчі

Маркетингова капча в даному випадку просить користувача ввести слоган фірми Volkswagen [21].

4.1.6 Капча «я не робот»

Капча «я не робот» вимагає від користувача встановити прапорець, щоб довести, що він не робот. Правдивість цього твердження визначається секундами виконання та інші складнішими механізмами. Капча фіксує час на комп'ютері та часовий пояс. Вона визначає приблизне місцезнаходження та браузер, який використовується. Вона бере до уваги навіть роздільну здатність екрана, на якому

відображається. Відстежуються також рух миші та інші засоби контролю курсору, натискання [22]. Зазвичай дана капча також сканує кукіси та аналізує історію пошукових запитів.

Приклад такої капчі можна побачити на рисунку 4.7.

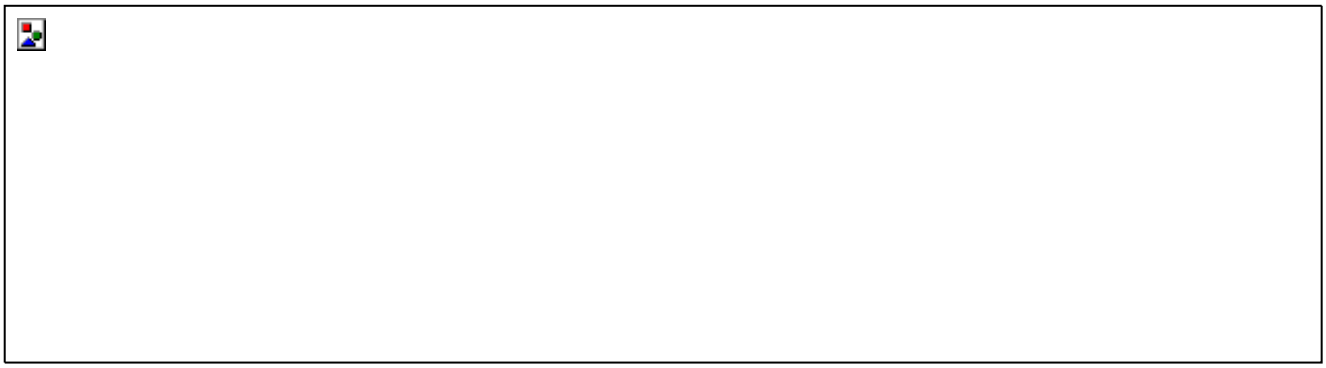


Рисунок 4.7 – Приклад капчі «я не робот»

Все описане вище потрібно для визначення, що за комп'ютером знаходиться людина. Якщо система все ще сумнівається, виводить текстову або іншу капчу, яка посилить впевненість визначення робота.

4.1.7 Звукова капча

Звукова капча представляє користувачеві серію літер або цифр. Часто користувач може запросити текстове відображення. Трапляється і зворотний варіант – коли звук стає альтернативою тексту. Приклад звукової капчі приведено на рисунку 4.8.

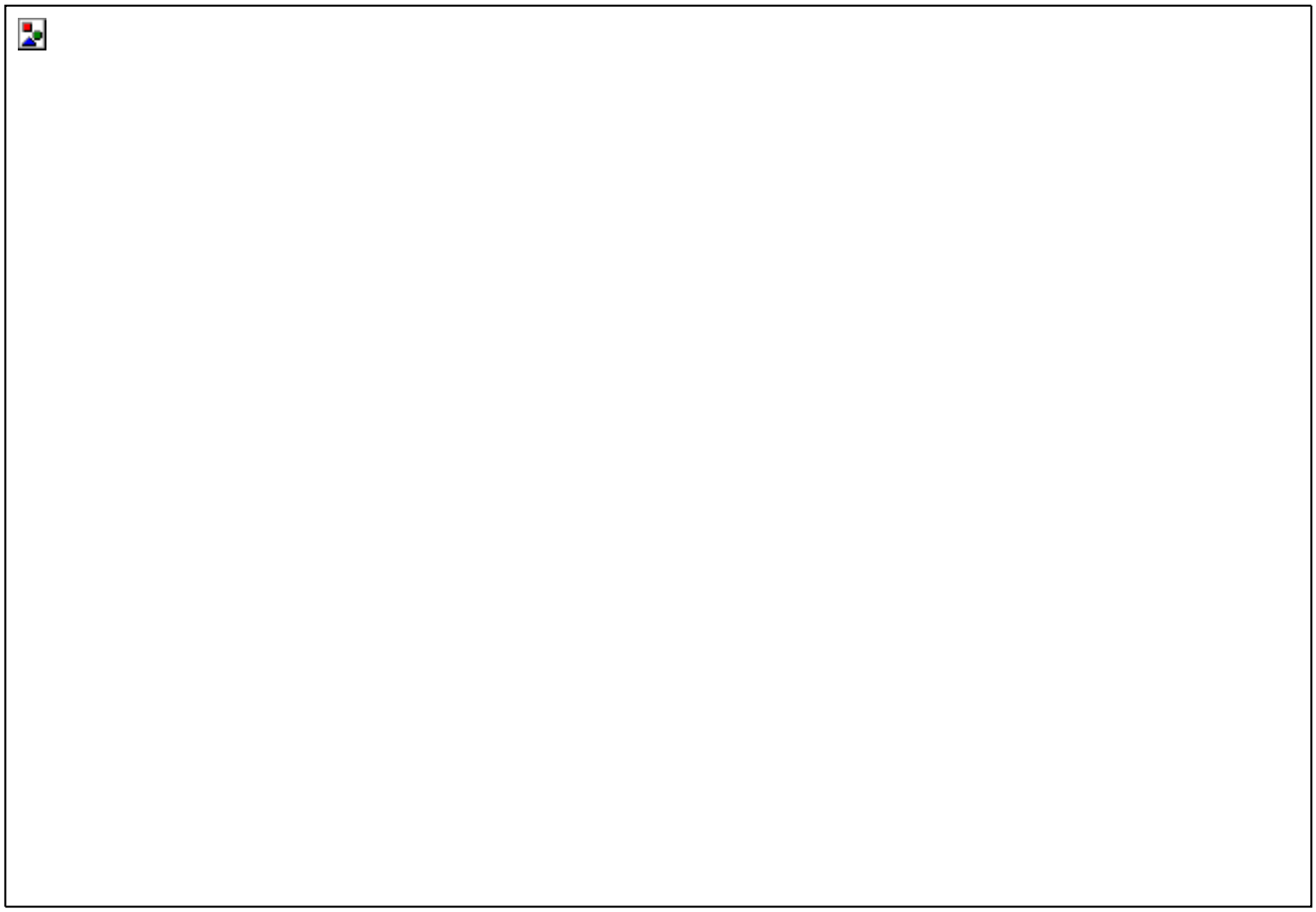


Рисунок 4.8 – Приклад звукової капчі

Проте, як видно з прикладу зображеного на рисунку вище, звукова капча зазвичай являє собою доповнення до текстової капчі, бо працює лише за тісної взаємодії з нею.

4.1.8 Капча «Drag-and-drop»

Так звана капча «Drag-and-drop» може запропонувати зібрати зображення, переміщаючи його частини [23]. Наочний її приклад приведено на рисунку 4.9.

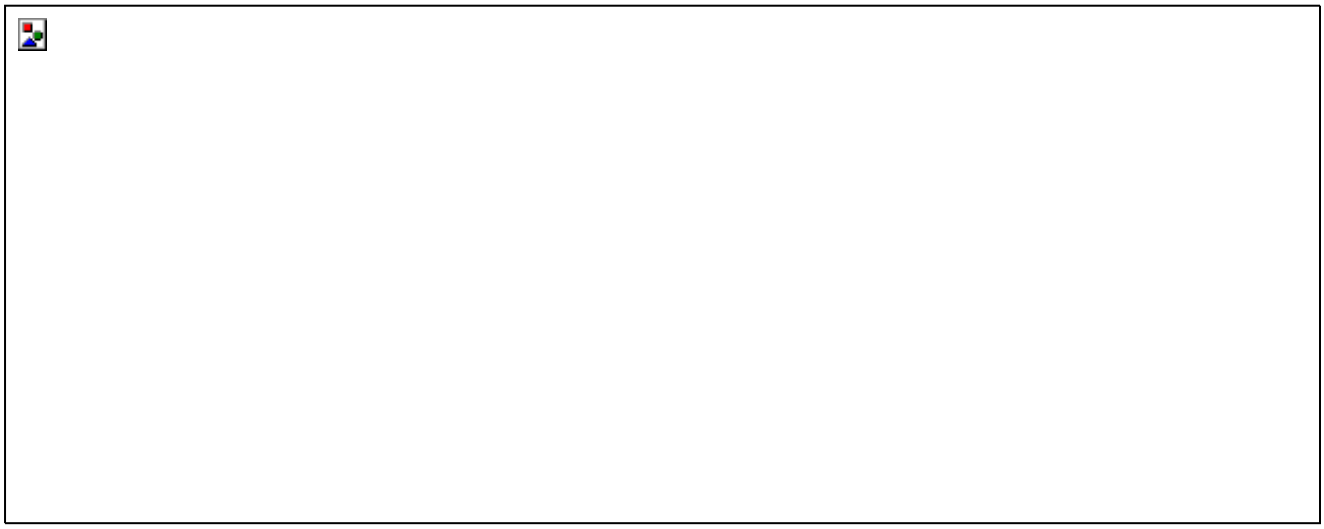


Рисунок 4.9 – Приклад капчі «Drag-and-drop»

В даному випадку система базує свій висновок на швидкості та способі керування мишкою людиною (чи ботом), та на базі аналізу робить припущення щодо того, з ким має справу.

4.2 Капча Honeypot

Капча Honeypot розміщує на екрані приховані поля, які невидимі людини, але зрозумілі для бота. Коли програмне забезпечення починає взаємодіяти з кодом, зрозуміло, що сервіс використовує не людина, а машина, і такі дії необхідно заблокувати [24].

5 РОЗРОБКА СИСТЕМИ ІДЕНТИФІКАЦІЇ ЛЮДИНИ ВІД ПРОГРАМИ У WEB ЗАСТОСУНКАХ

5.1 Передумови системи ідентифікації роботів

Метод CAPTCHA найбільш розпоширений та відомий в світі на сьогоднішній день. Проте він все ще не здатний справлятися з ідентифікацією людини від програми на прийнятному рівні. Технологія постійно ускладнюється, що робить все складнішим проходження перевірки людиною, тоді як програма з легкістю кожен раз вчиться долати ускладнення, не кажучи вже про людей, які проходять капчу на замовлення.

То ж наскільки прийнятно намагатися відрізнити людину від машини? Можливо, слід зайти з іншого боку, та спробувати віднайти машину серед користувачів.

Якщо розглядати питання з даного боку, слід перш за все визначитися з тим, що саме характеризує бота у веб просторі. То ж, якщо зібрати усі показники, краще за все характеризують бота наступні показники:

– аномально високі перегляди сторінок. Певні атаки ботів намагаються перевантажити сервери. Незалежно від того, чи це атака DDOS чи велика кількість скрейперів, це відображатиметься як раптовий, незрозумілий сплеск переглядів сторінок у аналітичному програмному забезпеченні;

– аномально високий показник відмов. У кожного бота є ціль. Як тільки він досягає своєї мети, або коли бачить, що не може досягти своєї мети, він прагне негайно піти. Оскільки боти можуть працювати за мілісекунди замість секунд, це покаже аномально високий і швидкий показник відмов;

– ненормальна тривалість сеансу – сеанси в діапазоні мілісекунд є підозрілими, як і аномально тривалі сеанси. Люди, як правило, залишаються принаймні на кілька секунд, але не часто залишаються на одній сторінці довше кількох хвилин. Варто слідкувати за відхиленнями тривалості сеансу у аналітичному програмному забезпеченні;

– сплески трафіку з невідомих місць. Наприклад, якщо бізнес працює у В’єтнамі, але раптом сайт отримуєте приплив запитів з В’єтнаму, є велика ймовірність, що це атака ботів. Запити, які надходять із країн, які не мають відношення для бізнесу, часто є запитами ботів;

– непотрібні перетворення – отримання повідомлень з контактної форми, які не мають сенсу, чи постійно певні користувачі розміщують товари в кошиках, не купуючи їх, безкоштовний інформаційний бюлетень раптом має велику кількість відмов – всі ці показники є небажаними перетвореннями, які вказують на поведінку бота.

Звичайно, визначення комп’ютера є складною задачею. Так виявлення ботів має наступні складнощі:

– боти атакують усі кінцеві точки, вони більше не атакують лише веб-сайти. Вони також атакують мобільні додатки, веб-програми, сервери та API. Тож якщо система має будь-яку з цих кінцевих точок, залишати одну з них незахищеною небезпечно;

– боти використовують ті ж технології, що й люди. Вони використовують браузери, які мають надзвичайно схожі відбитки пальців на людські браузери, і можуть, наприклад, вдатися до ферм мобільних телефонів, щоб використовувати реальні пристрої замість імітованих;

– оператори ботів можуть легко розподіляти свої атаки в часі та просторі. Вони можуть атакувати API мобільного додатку протягом кількох днів у різних країнах, причому все це з дуже незначними зусиллями та за невеликими фінансовими витратами;

– боти можуть обертатися через мільйони чистих домашніх IP-адрес. Часто кожен IP надсилає не більше одного або двох запитів, перш ніж бот перейде на інший IP. Багато рішень безпеки, такі як WAF, покладаються виключно на IP-адреси, щоб відрізняти ботів від людей. Цей трюк робить їх неефективними.

– поява ботів як сервісу (BaaS) тепер дозволяє будь-кому запускати атаку ботом. Ці послуги дають операторам-зловмисникам можливість налаштувати ботнет і надсилати ботів на певний веб-сайт або додаток. Оскільки ці служби

налаштовані таким чином, щоб їхні користувачі платили лише за успішні запити, їх стимулюють зробити своїх ботів якомога більш розвиненими. Усі ці фактори разом роблять виявлення ботів неймовірно складним.

5.2 Модель системи ідентифікації роботів

Тож, підходячи до розробки системи, слід визначитись з тим, що саме буде використано та бажаний кінцевий результат.

Розроблена система включає комплексне рішення, що складається з двох обов’язкових частин. Звичайно, окремо кожна з частин здатна надавати результати, але при комплексний підхід є більш якісним та результативним.

То ж, слід створити скрипт, що буде так званим прошарком між сайтом та сервером, він збиратиме дані усіх запитів (саме всіх запитів, не певну кількість запитів один раз в певний проміжок часу, а усі запити), та аналізувати отримані дані.

Інша частина підходу полягає у використанні штучного інтелекту на стороні користувача, що аналізує аномалії користувача.

Основуючись на обох аналізах, штучний інтелект зможе винести кінцеве рішення щодо того, перед ним людина чи бот.

Звичайно, для штучного інтелекту необхідна величезна кількість даних, які він використає для навчання. Величезну кількість даних створити власними силами неможливо, але можна надати ШІ початкові вхідні дані, та натренувати декількома атаками.

На рисунку 5.1 приведено діаграму розгортання системи.

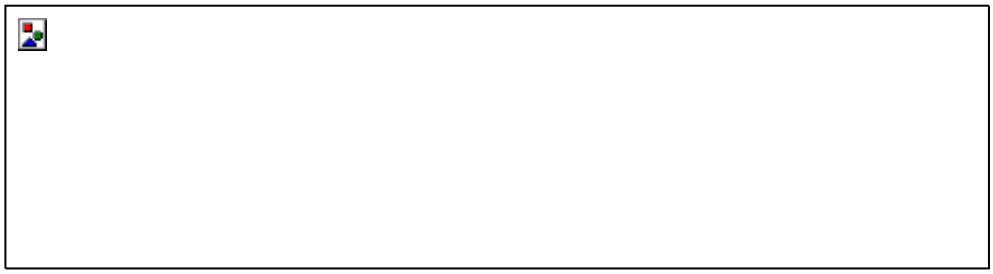


Рисунок 5.1 – Deployment diagram

Аномалії, які згадано в діаграмі в контексті детектора аномалій, – це точки даних, які виділяються серед інших точок даних у наборі даних і не підтверджують нормальну поведінку даних [25]. Ці точки даних або спостереження відхиляються від нормальних моделей поведінки набору даних.

В розроблюваній системі аномаліями вважаються дії ботів, які слід відсіяти та виявити серед усіх даних.

Виявлення аномалій – це неконтрольована методика обробки даних для виявлення аномалій із набору даних. Аномалію можна широко розділити на кілька категорій:

– викиди: короткі/дрібні аномальні моделі, які несистематично з’являються під час збору даних;

– зміна подій: систематична або раптова зміна попередньої нормальної поведінки;

– зміни: повільна, ненаправлена, довготривала зміна даних.

Прості статистичні методи, такі як середнє значення, медіана, квантилі, можна використовувати для виявлення значень ознак одновимірних аномалій у наборі даних. Для виявлення аномалій також можна використовувати різноманітні методи візуалізації та дослідницького аналізу даних.

Для визначення чи являється певне значення аномальним використовується так званий алгоритм Local Outlier Factor (LOF) – неконтрольований метод виявлення аномалій, який обчислює локальне відхилення щільності даної точки даних щодо її сусідів.

Коли точка вважається аномальною її локальними сусідніми точками, її називають локальним викидом.

Використовуючи LOF, ми можемо визначити викид, враховуючи його щільність сусідства. LOF використовується, коли набір даних має різну щільність.

Передумови для розуміння LOF:

– K-відстань і K-сусіди;

– відстань досяжності (RD);

– щільність локальної доступності (LRD);

– коефіцієнт локального викиду (LOF).

K-відстань – це відстань між точкою та її найближчим сусідом K. Сукупність точок, які лежать в колі радіуса K-відстані, називається K сусіднім. Ця множина позначається Nk (A).

Відстань досяжності визначається як максимальна K-відстань Xj і відстань між Xi і Xj. Міра відстані залежить від конкретної проблеми (евклідова, Манхеттенська тощо).

Приклад даного алгоритму відображено на рисунку 5.2.

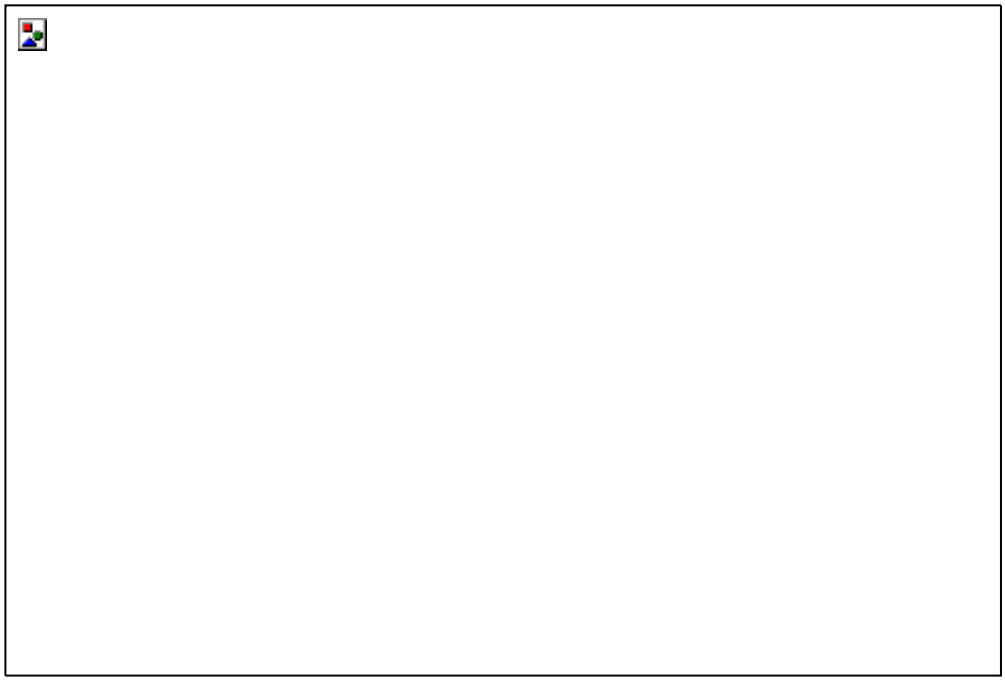
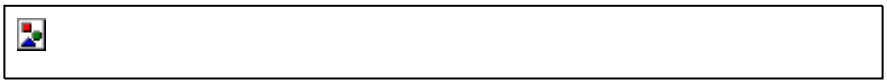


Рисунок 5.2 – Наочне зображення відстані досяжності (RD).

Якщо точка X_i знаходиться в межах K-сусідів X_j, відстань досяжності буде K-відстанню X_j (синя лінія), інакше відстань досяжності буде відстанню між X_i та X_j (помаранчева лінія).

Для відстань досяжності використовується наступна формула:

 (1)

Наступним кроком слід визначити щільність локальної доступності. LRD є зворотним до середньої відстані досяжності точки A від її сусідів, що відображено у його формулі:

 (2)

Інтуїтивно, згідно з формулою LRD, чим більше середня відстань досяжності (тобто сусіди далекі від точки), тим менша щільність точок навколо певної точки. Це говорить про те, наскільки віддалена точка від найближчого скупчення точок. Низькі значення LRD означають, що найближчий кластер знаходиться далеко від точки.

Після визначення LRD можна переходити до LOF.

LRD кожної точки використовується для порівняння із середнім LRD її K сусідів. LOF – це відношення середнього LRD K сусідів A до LRD A. Це можна побачити на приведеній нижче формулі LOF:

 (3)

Інтуїтивно, якщо точка не є викидом (внутрішнім), відношення середніх LRD сусідів приблизно дорівнює LRD точки (оскільки щільність точки та її

сусідів приблизно рівні). У цьому випадку LOF майже дорівнює 1. З іншого боку, якщо точка є викидом, LRD точки менший, ніж середній LRD сусідів. Тоді значення LOF буде високим.

Як правило, якщо LOF > 1, це розглядається як аномалія.

Як вже було описано вище, даний алгоритм оцінює імовірність, що конкретна точка даних є аномалією для кожного значення. В процесі аналізуються наступні фактори:

- показник відмов;
- тривалість сеансу;
- перегляд кукі браузеру;
- місце надходження трафіку;
- часова відмітка країни;
- запити, що відправляє користувач.

Штучний інтелект аналізує усі зібрані характеристики, та видає кінцеве рішення щодо того, чи являється користувач ботом.

У випадку, якщо вірогідність того, що користувач є людиною, йому дозволяється доступ до сервісу. В інакшому випадку користувач більше не може переглядати ресурс.

5.3 Рішення для ідентифікації роботів

Рішення для захисту від роботів поєднує передові технології та надійні алгоритми для виявлення, аналізу та класифікації шаблонів і підписів ботів. Система аналізує комбінацію методологій, включаючи унікальний відбиток пристрою, динамічні тести Тьюринга, аналіз поведінки користувачів і проблеми JavaScript [26]. Механізм виявлення використовує різні форми машинного навчання (ML) для навчання алгоритмів на основі відомих шаблонів і історичних даних, щоб виявляти нові типи ботів і зупиняти їх атаки.

Виклик API і вбудований тег JavaScript збирають і передають кілька параметрів відвідувача для обробки системою виявлення ботів. Механізм виявлення диспетчера ботів працює в режимі реального часу, аналізуючи кожного відвідувача сервісу, а також розробляє унікальний відбиток для кожного відвідувача та бота.

Якщо відвідувачем є людина, пошукова система чи бот-партнер, менеджер ботів надає доступ за кілька мілісекунд, не сповільнюючи роботу користувача. Однак, коли виявлено бота, менеджер ботів може заблокувати його, показати CAPTCHA, надати йому підроблені дані та інші типи відповідей на основі потреб організації.

5.4 Проблема ін’єкції JavaScript

Існує два фундаментальних підходи до захисту веб-додатків – один, який зосереджується на змінах у веб-додатку (або мобільному), а інший, який приховано працює на стороні сервера. Наразі спостерігається перехід продуктів безпеки веб-додатків до прихованого виявлення на стороні сервера в порівнянні з традиційним методом ін’єкції JavaScript.

За іронією долі, як стандартний підхід, як кіберзлочинці, так і багато компаній, що займаються захистом веб-додатків, досі зазвичай використовуються як стандартний підхід. Хоча він може виявляти та пом’якшувати деякі атаки, існує кілька обмежень і внутрішньо-організаційних проблем, які виникають із застосуванням підходу на основі JS.

Модель JS працює, вставляючи скрипт у веб-програму, який «відбиває відбитки» браузера користувача, щоб розрізняти активну активність користувача та активність бота. Коли рішення безпеки намагаються визначити аномалії в цілій групі великих користувачів або тенденції протягом тривалого періоду часу, вони зазвичай використовують комбінацію JS і файлів cookie, щоб спробувати виявити

автоматичну атаку. Цей механізм вимагає, щоб клієнт надав інформацію браузера, щоб він міг оцінити дані про поведінку користувача під час сеансу, наприклад, чи відповідає поведінка миші параметрам «по-людськи», щоб ідентифікувати зловмисну автоматичну активність або небажаний трафік.

Два основних варіанти використання для виявлення на основі JS – це рішення для запобігання шахрайству та виявлення ботів. У багатьох випадках ці дії часто зводяться до перевірки того, хто відомий користувач, про те, ким він є, і до ідентифікації нового користувача чи нового пристрою.

Метод на основі JS корисний у випадку шахрайства, коли відомі користувач, його пристрій, місцезнаходження та модель використання. Оскільки рішення має журнал історії користувача, помітне відхилення від цієї історії є вагомою підставою для підозри у шахрайстві. Однак, коли справа доходить до випадку використання безпеки, ін’єкція JavaScript не настільки ефективна.

6 ПЕРСПЕКТИВИ СИСТЕМ ВІДМІННОСТІ ЛЮДИНИ ВІД ПРОГРАМИ У WEB ЗАСТОСУНКАХ

Чисто з філософського погляду перспективи капчі та інших засобів, заснованих на розпізнаванні суб'єкта, який виконує захисну дію, і відділенні таким чином «правильних» суб'єктів від «неправильних» – доволі туманні. Ця гонка рано чи пізно призведе до того, що пропускатиме «розумних» людей і «розумні» програми, але відсіюватиме разом з не дуже «розумними» програмами і частину людей [27]. Та й уже сьогодні «тест на людяність» не вирішує проблему найманих розпізнавачів. Тож потрібні інші рішення.

Щоб розробити засіб, що дозволяє мінімізувати негативну інформаційну активність (спам, флуд тощо), потрібно передусім виділити характерні властивості такої активності (а не її безпосередніх виконавців). Усі капчі та подібні до них «тести на людяність» намагаються визначити, хто намагається користуватися системою. В результаті виходить система, яка:

– сприяє дискримінації людей з обмеженими можливостями (наприклад сліпих) – бо не всі власники, що впроваджують візуальну капчу, бажають морочитися зі звуковою;

– позбавляє сумлінних користувачів потенційних додаткових зручностей (можливості автоматизованого доступу в рамках fair use) [28];

– не виконує свою основну функцію – не захищає від негативної інформаційної активності.

Бо розподіл світу на «хороших» суб'єктів, які завжди виконують хороші дії, та «поганих», які завжди виконують погані, м'яко кажучи, не надто відповідають реальності. Відповідно, і рішення у стилі «вбити всіх поганих» просто не працюють. Тим більше, що зловмисник, зрештою, завжди є людиною, а не ботом.

Реально діючим підходом у захисті від спаму та флуду може бути тільки фільтрація самих дій за тими ознаками, які відрізняють зловмисні дії від

сумлінних, незалежно від того, який суб'єкт є їх безпосереднім виконавцем – програмний агент сумлінного користувача чи найманець, який працює на спамера. Відслідковувати можна як суто технічні особливості (надмірна частота або обсяг повідомлень, спроба надсилання множини однакових чи статистично схожих повідомлень тощо), так і реакцію, яку такі повідомлення викликають в учасників. Одночасно з суб'єктивними ідентифікаторами (неймінгом) слід оцінювати об'єктивні (IP) та блокувати їх за необхідності, знову ж таки суворо або за маскою. Ефективними будуть засоби, що дозволяють швидко виявляти відмінні характеристики зловмисної інформаційної активності та швидко усувати її наслідки шляхом бану або повного видалення, наприклад, за допомогою SQL-подібної мови та регулярних виразів.

Політика ресурсу щодо таких дій (що можна, а чого не можна) має бути відкритою, у тому числі й для програмних агентів. Якщо агент намагається виконати дію, несумісну з політикою ресурсу, ресурс повинен повернути йому стандартну негативну відповідь, заховану у мікроформаті десь усередині HTML. Якщо модератор виявить, що заявленої політики недостатньо і зловмисник її оминув – політику буде уточнено та оновлено. У міру обкатки ефективність сервісу підвищуватиметься, а політика – еволюціонуватиме у бік максимальної зручності для власника сервісу та для його користувачів.

А змушувати користувачів виконувати разом із цільовою дією ще й нецільову – це і розтрата ресурсів (часу, уваги), а коли деякі користувачі чисто фізично не можуть виконувати нав'язану дію – ще й дискримінація.

Фахівці підрахували, що в середньому користувач витрачає на виконання капчі 32 секунди.

Враховуючи, що у світі налічується 4,6 мільярда користувачів інтернету, а кожен користувач бачить вимогу заповнити CAPTCHA в середньому один раз на 10 днів, то виходить, що щодня людство витрачає на марне заняття 500 років. Крім того, сучасні завдання, які пропонує CAPTCHA, зробили інтернет менш доступними для людей із фізичними та когнітивними порушеннями.

Позбавити світ капчі намагаються зараз декількома способами. Наприклад, Google вигадав нову систему захисту від ботів No-CAPTCHA. Ця система працює у кілька етапів. На першому досить просто підтвердити галочкою твердження «Я не робот». У цьому випадку спеціальна програма за непрямими ознаками (всі параметри не розкриваються, відомо, що скрипт дивиться IP-адресу та час, проведений на сайті) розрізняє де бот, а де користувач.

Дійсно, в даний час CAPTCHA не може забезпечити необхідний рівень безпеки. Існує велика кількість способів обійти цей захист та забезпечити доступ незаконним користувачам. Проте наразі система капчі є найбільш розповсюдженою та прийнятною.

ВИСНОВКИ

На сьогоднішній день існує наявна потреба в ідентифікації людини від програми в будь-яких застосунках, проте постає тяжка проблема в занадто швидкому навчанні ботів, коли вони стають здатні вирішувати проблеми на порядок краще звичайних людей.

Спроби ускладнити завдання призвели лиш до того, що звичайні люди перестали справлятися с поставленою задачею. Саме тому перед людьми наразі постало складне завдання – ускладнити капчу настільки, щоб всі люди могли без зайвих проблем справитися з задачею, навіть незважаючи на мовні розбіжності та вікові показники.

Звичайно, капча подобається далеко не всім, але на даний момент це найкраща технологія для боротьби з ботами, які використовують системи автоматизації для вирішення завдань своїх власників. Розробники тестів намагаються зробити їх якнайпростішими для користувачів, але зі зростанням активності зловмисників це завдання стає тільки складнішим.

Рішення для виявлення та керування роботами допомагає виявляти й керувати будь-якими видами ботів, як хорошими, так і поганими – на основі конкретних потреб організації. Воно працює в режимі реального часу, щоб ідентифікувати трафік ботів і вжити необхідних заходів – блокування, показ CAPTCHA для вирішення, надання йому підроблених даних або відключення з’єднання. На перспективу планується можливість легкої інтеграції інструментів керування роботами з широким спектром архітектур веб-інфраструктури та додатків, щоб задовольнити унікальні потреби різних користувачів.

Дана атестаційна робота містить в собі аналіз галузі та основні актуальні на сьогоднішній день шляхи вирішення даного питання, дослідження та аналіз методів відмінності та ідентифікації людини від програми у web застосунках, та розробку власного методу ідентифікації людини від боту.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Chen J. et al. A survey on breaking technique of text-based CAPTCHA //Security and communication networks. – 2017. – Т. 2017.

2. Gossweiler R., Kamvar M., Baluja S. What's up CAPTCHA? A CAPTCHA based on image orientation //Proceedings of the 18th international conference on World wide web. – 2009. – С. 841-850.

3. Von Ahn L. et al. recaptcha: Human-based character recognition via web security measures //Science. – 2008. – Т. 321. – №. 5895. – С. 1465-1468.

4. Дідківська С. О. Людяність штучного інтелекту та тест Тюрінга //ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ В ОСВІТІ ТА. – 2018. – С. 85.

5. Кулевський Д. І. Програмна система управління проектами розробки web-додатків. – 2020.

6. Sivakorn S., Polakis I., Keromytis A. D. I am robot:(deep) learning to break semantic image captchas //2016 IEEE European Symposium on Security and Privacy (EuroS&P). – IEEE, 2016. – С. 388-403.

7. Fedor J., Malenfant A., Zennaro M. CAPTCHA using Word Relationships. – 2016.

8. Козлова О. В. Переваги експертних систем над традиційними системами штучного інтелекту //Системи озброєння і військова техніка. – 2011. – №. 1. – С. 104-106.

9. Lopresti D. Leveraging the CAPTCHA problem //International Workshop on Human Interactive Proofs. – Springer, Berlin, Heidelberg, 2005. – С. 97-110.

10. Cui J. S. et al. A CAPTCHA implementation based on moving objects recognition problem //2010 International Conference on E-Business and E-Government. – IEEE, 2010. – С. 1277-1280.

11. Alajmi M. et al. A password-based authentication system based on the CAPTCHA AI problem //IEEE Access. – 2020. – Т. 8. – С. 153914-153928.

12. Бондаренко М. Ф., Четвериков Г. Г. Феноменология мозгоподобных преобразователей информации. – 2013.

13. Singh K. J., De T. DDOS attack detection and mitigation technique based on Http count and verification using CAPTCHA //2015 International Conference on Computational Intelligence and Networks. – IEEE, 2015. – С. 196-197.

14. D'Souza D., Polina P. C., Yampolskiy R. V. Avatar captcha: Telling computers and humans apart via face classification //2012 IEEE International Conference on Electro/Information Technology. – IEEE, 2012. – С. 1-6.

15. Kuppusamy K. S., Aghila G. HuMan: an accessible, polymorphic and personalized CAPTCHA interface with preemption feature tailored for persons with visual impairments //Universal Access in the Information Society. – 2018. – Т. 17. – №. 4. – С. 841-864.

16. Almazyad A. S., Ahmad Y., Kouchay S. A. Multi-modal captcha: A user verification scheme //2011 International Conference on Information Science and Applications. – IEEE, 2011. – С. 1-7.

17. Singh V. P., Pal P. Survey of different types of CAPTCHA //International Journal of Computer Science and Information Technologies. – 2014. – Т. 5. – №. 2. – С. 2242-2245.

18. Chen J. et al. A survey on breaking technique of text-based CAPTCHA //Security and communication networks. – 2017. – Т. 2017.

19. Gossweiler R., Kamvar M., Baluja S. What's up CAPTCHA? A CAPTCHA based on image orientation //Proceedings of the 18th international conference on World wide web. – 2009. – С. 841-850.

20. Aboufadel E., Olsen J., Windle J. Breaking the Holiday Inn priority club CAPTCHA //The College Mathematics Journal. – 2005. – Т. 36. – №. 2. – С. 101-108.

21. Sawada K., Uda R. Effective CAPTCHA with Amodal Completion and Aftereffects //Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication. – 2016. – С. 1-5.

22. Fortunati L. et al. You need to show that you are not a robot //New Media & Society. – 2019. – Т. 21. – №. 8. – С. 1859-1876.

23. Desai A., Patadia P. Drag and drop: A better approach to CAPTCHA //2009 Annual IEEE India Conference. – IEEE, 2009. – С. 1-4.

24. Abdullahi M. A., Aliyu S., Junaidu S. B. An enhanced intrusion detection system using honeypot and CAPTCHA techniques //Fudma Journal of Sciences. – 2019. – Т. 3. – №. 3. – С. 202-209.

25. Четвериков Г. Г. Концепція уніфікації методів та засобів побудови просторових багатозначних структур. – 2010.

26. Ефимова И. А. и др. Синтез бинарных логических сетей и особенности их функционирования. – 2006.

27. Yan J., El Ahmad A. S. Captcha robustness: A security engineering perspective //Computer. – 2010. – Т. 44. – №. 2. – С. 54-60.

28. Banday M. T., Sheikh S. A. Design of Secure Multilingual CAPTCHA Challenge //International Journal of Web Portals (IJWP). – 2015. – Т. 7. – №. 1. – С. 1-27.

29. Chetverikov G., Puzik O., Vechirska I. Multiple-valued structures of intellectual systems //Proceedings of the with Internations Computer Sciences and Information Technologies (CSIT). 2016, 7589907. -pp. 204-207