3. Пособие по методике преподавания русского языка как иностранного. – М.: Русский язык, 1984. – 185с.

Получено 18.05.16

УДК 004.9

О. М. Чорний
Національний технічний університет України «Київський політехнічний інститут»

Виявлення form-ботів із використанням методів машинного навчання

У даній роботі розглядається метод виявлення ботів, які виконують автоматичне заповнення і відправку веб-форм, що базується на використанні методів машинного навчання.

виявлення ботів, машинне навчання, навчання з учителем, класифікація

В наш час глобальна мережа Інтернет набула широкого поширення. Люди використовують її для спілкування, отримання останніх новин, пошуку необхідної інформації, інтернет-банкінгу, здійснення покупок в інтернет-магазинах тощо. Одним із основних способів вводу інформації та реєстрації на сайтах є *форми* – набори полів вводу чи вибору інформації, відповідних їм текстових міток та кнопки, при натисненню на яку введена інформація передається для подальшої обробки сервером.

Форми є надзвичайно привабливою мішенню для спамерів. У даній роботі ми будемо вважати за *спам* будь-яку небажану дію або введення даних на сайті, будь то щось шкідливе, що приносить дохід спамеру або ж взагалі не відповідає призначенню і тематиці сайту.

Специфіка роботи веб-застосунків передбачає можливість створення *ботів* – застосунків, що здатні заповнювати форми та виконувати їх відправку автоматично. Будь-яка автоматизація,

очевидно, приводить до надзвичайної швидкості і ефективності розповсюдження спаму. Власники веб-ресурсів не спроможні власноруч перевіряти та фільтрувати все, що надходить від користувачів, а значить необхідною є наявність систем, здатних автоматично виявляти ботів при спробі відправки форми на веб-сервер. Таким чином, метою даної роботи є побудова системи автоматичного розпізнавання form-ботів.

1. Методи захисту

Найбільш розповсюдженим на сьогодні методом боротьби із form-ботами є CAPTCHA [1] та її різновиди, наприклад reCAPTCHA [2]. Зазвичай вона являє собою спотворене зображення рядка, що складається із алфавітно-цифрових символів, котрий людина повинна розпізнати й відправити на перевірку серверу. На рис. 1 зображено приклад такої CAPTHA.

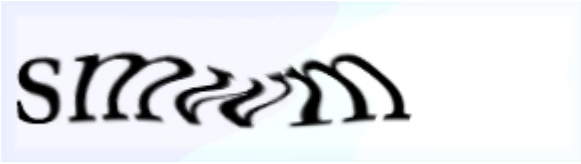

Рисунок 1 – Текстова CAPTCHA

Такий вид CAPTCHA має ряд недоліків:

• Досить часто текст спотворюється наскільки, що розпізнати його не під силу навіть людині. В зв’язку із цим тест доводиться проходити кілька разів, витрачаючи досить багато часу.

• З іншого боку, на сьогоднішній день галузь OCR розвинута достатньо для того, щоб розпізнавати текст найбільш розповсюджених реалізацій текстової CAPTCHA автоматично.

Однією із можливих альтернатив є система розпізнавання побудована на основі вимірювання певних біометричних характеристик людської поведінки. Незважаючи на певні успіхи сучасної науки у спробах створення штучного інтелекту та вирішення деяких суміжних задач, імітація людської поведінки досі є надзвичайно складною задачею для комп’ютерів та роботів. Це говорить про те, що система виявлення form-ботів із використанням

біометричних характеристик людини має забезпечувати достатній рівень захисту.

Враховуючи, що розроблювана система повинна працювати у веб-середовищі і має бути пристосована для роботи із будь-яким користувачем було вирішено використовувати характеристики руху рук користувачів при користуванні комп'ютерною мишкою.

2. Архітектура системи

Система розпізнавання може бути вбудованою в системи реєстрації, відправки повідомлень, оформлення замовлень та інші системи більшості веб-ресурсів. Складається вона із двох підсистем: клієнтської та серверної.

Клієнтська частина реалізується мовою програмування JavaScript. Основне її призначення це збір даних про рух руки користувача та їх відправка для подальшої обробки та класифікації на сервер.

Серверна частина приймає на вхід зібрані дані та приймає рішення про віднесення користувача до одного з класів – «людина» або «бот».

Підсистема логування буде активуватись в той момент, коли користувач натискає на кнопку відправлення даних на сервер. В цей момент перед ним на екрані з'явиться спеціальне завдання, під час виконання якого буде здійснюватись запис всіх його рухів для подальшої класифікації. Дане завдання повинно мати певний елемент випадковості аби унеможливити ситуацію, коли спамер один раз проходить його власноруч, записує всі свої рухи за допомогою спеціального програмного забезпечення, після чого налаштовує бота на їх відтворення.

3. Метод вирішення задачі

Введемо деякі позначення.

Точкою p_i будемо називати трійку $\langle x_i, y_i, t_i \rangle$, де $x_i \in Z$ – координата *i*-ої точки по осі абсцис, $t_i \in Z$ – координата *i*-ої точки по осі ординат, $t_i \in R^+$ – час реєстрації *i*-ої точки.

Визначення 1. Шляхом π довжини N будемо називати скінченну послідовність точок $(p_i)_{i=1}^{N}$ для якої виконується наступна властивість:

$$\forall i = \overline{2, N} : t_{i-1} < t_i .$$

Довжину шляху π позначимо $|\pi|$.

Множину усіх шляхів довжини $n \in N$ позначимо Π_n. Тоді $\Pi_* = \bigcup_{n=1}^{\infty} \Pi_n$ – множина усіх шляхів довільної довжини. Позначимо $\Pi^* = \bigcup_{n=1}^{\infty} \Pi_*^n$ – множина усіх кортежів довільної довжини, елементами яких є шляхи довільної довжини.

Для зручності, позначимо класи користувачів наступним чином: $+1$ – «людина», -1 – «робот».

Нехай, існує невідома нам залежність $y(\pi^*)$ між набором шляхів, що надійшли від користувача та класом до якого він належить:

$$y : \Pi^* \to \{-1, +1\}$$

Тоді, для побудови системи розпізнавання нам необхідно побудувати алгоритм $a(\pi^*)$, який за скінченну кількість кроків міг би обчислити значення $y(\pi^*) \forall \pi^* \in \Pi^*$. Тепер необхідно вирішити, яким чином ми можемо побудувати алгоритм $a(\pi^*)$. Очевидно, що існує нескінченна кількість кортежів $\pi^* \in \Pi^*$, які характерні як для бота так і для людини. До того ж, структура такого об'єкту є занадто складною для побудови явного і точного алгоритму, який міг би безпомилково віднести його до одного із класів. Тому оберемо іншу стратегію: сформуємо множину $\Pi_{train} \subset \Pi^*$ яка буде складатись як із даних характерних для людей, так і з даних характерних для ботів і спробуємо із її допомогою побудувати такий алгоритм $a(\pi^*)$, який би якомога точніше наближував істинну залежність $y(\pi^*)$ на всій множині Π^* при цьому для оцінки якості отриманого на певному етапі побудови алгоритму будемо використовувати множину Π_{train}, для якої правильні відповіді нам відомі. Комплекс математичних засобів, здатних розв'язувати дану задачу називається *Машинним навчанням*, а наша задача найбільш точно відповідає класу задач, що називається *навчання з учителем*.

Для того щоб звести задачу виявлення ботів до задачі машинного навчання із учителем необхідно виконати наступні кроки:

• зібрати навчальну вибірку, яка буде складатись як із людських даних так і з даних ботів;

• скласти ознаковий опис об’єктів нашої задачі;

• обрати алгоритм класифікації, який би давав найкращі результати;

• обрати метрику якості і оцінити якість отриманої моделі.

4. Збір даних

Для того, щоб було можливо залучити якомога більше людей для збору даних було вирішено створити спеціальну веб-сторінку, що була б доступна через глобальну мережу Інтернет. Аби спонукати відвідувачів сторінки до виконання якомога більшої кількості рухів мишкою їм пропонувалось виконати спеціальне завдання. На сторінці у спеціальному HTML-елементі <canvas> було створено зображення прямокутника, яке відвідувачі повинні були «затерти» використовуючи свою мишку. Після того як завдання виконано користувач повинен був відправити зібрані дані натиснувши на кнопку «Send». При цьому відбувалась перевірка того, чи дійсно відвідувач виконав завдання і не намагається відправити порожній набір із даними. На рис. 2 зображено сторінку під час виконання завдання. Всього вдалось зібрати 390 наборів даних від близько 90 користувачів.

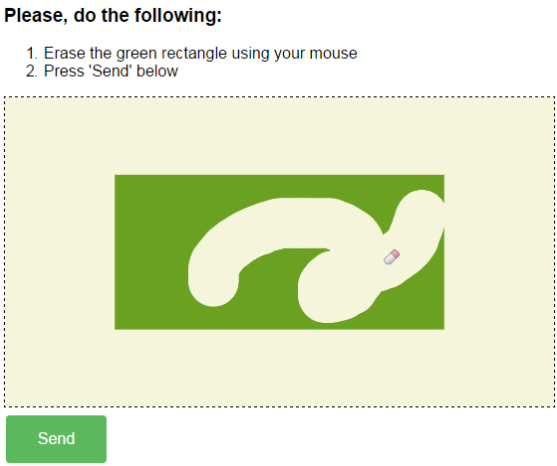

Рисунок 2 – Сторінка для збору даних під час виконання завдання

Для отримання даних, характерних для ботів необхідно було змоделювати бота, який би намагався імітувати людський рух. Під ботом $B(n)$ ми будемо розуміти деякий алгоритм, який приймає на вхід довжину шляху n і повертає шлях $\pi = (p_1, p_2, \ldots, p_n)$ відповідної довжини. Шлях тут розуміється у сенсі визначення 1.

Всі моделі ботів можна розділити на 2 класи:

- моделі, що використовують для генерації точок статистичні дані про рух справжніх людей;
- моделі, що не використовують таких даних.

Якщо розглядати рух руки людини як послідовність точок, які реєструються операційною системою, то одною із основних його властивостей є те, що положення кожної наступної точки залежить від попередніх. Ця залежність проявляється у двох головних характеристиках руху: напрямі та швидкості. Зважаючи на це, моделювати бота першого типу було вирішено як Марківський процес [5] із дискретним часом, де кожен стан відповідного Марківського ланцюга був представлений у вигляді пари (φ_i, v_i), де φ_i – кут між вектором $\overrightarrow{p_{i-1}, p_i}$ та додатнім напрямом осі абсцис, а v_i – швидкість руху від точки p_{i-1} до точки p_i. Оцінювання початкових та перехідних імовірностей Марківського ланцюга виконувалось із використанням даних про рух рук користувачів, зібраних описаним вище способом.

Поведінка бота другого типу була значно простішою – точки генерувались випадковим чином із використанням рівномірного, нормального та експоненційного законів розподілу для отримання значень швидкості та кута повороту, подібно до бота першого типу.

5. Ознаковий опис об'єктів

Використані у даній роботі ознаки об'єктів базуються на фундаментальних фізичних характеристиках руху: швидкості, прискоренні, кутовій швидкості та кутовому прискоренні. Нехай $p_i, p_{i+1}, p_{i+2}, p_{i+3}$ послідовні точки деякого шляху π. У табл. 1 наведено формули для обчислення перелічених вище характеристик на основі даних точок.

Таблиця 1 – Фізичні характеристики руху адаптовані до нашої задачі

Опис	Формула
Швидкість руху	$v_i = \frac{\sqrt{(x_{i+1} - x_i)^2 - (y_{i+1} - y_i)^2}}{t_{i+1} - t_i}$
Прискорення	$a_i = \frac{v_{i+1} - v_i}{\frac{t_{i+2} - t_i}{2}}$
Кутова швидкість	$\omega_i = \frac{\Delta\varphi_i}{\frac{t_{i+2} - t_i}{2}}$
Кутове прискорення	$\varepsilon_i = \frac{\omega_{i+1} - \omega_i}{\frac{t_{i+3} + t_{i+2} - t_{i-1} - t_i}{4}}$

Тут $\Delta\varphi_i$ є кутом повороту вектору $\overrightarrow{p_{i+1}, p_{i+2}}$ відносно вектору $\overrightarrow{p_i, p_{i+1}}$.

Таким чином, кожна характеристика шляху є не окремим числом, а послідовністю дійсних чисел. У якості ознак брались наступні статистичні показники даних послідовностей:

- Середнє значення
- Середньоквадратичне відхилення
- 0.25-квантиль
- 0.75-квантиль

Загалом же, 4 характеристики та 4 статистичні показники дають нам 16 ознак.

6. Використані алгоритми

Для класифікації об’єктів було використано насту-пні алгоритми:

- *Логістична регресія* [3], яка є прикладом лінійного класифікатора.
- *Градієнтний бустинг* [4], що є композицією більш простих алгоритмів – дерев прийняття рішень.
- *Зважене голосування*, що також є композиційним методом, яке повертає у якості відповіді зважену суму відповідей кількох інших алгоритмів.

Тестування виконувалось за методом *крос-валідації*. При використанні даного методу вся навчальна вибірка розбивається на K частин, після чого відбувається K ітерацій навчання, в кожній із яких одна із частин використовується в якості тестової вибірки для обчислення якості класифікації, а решта K-1 частин беруться в якості навчальної вибірки. Отримані на кожній ітерації результати усереднюються, в результаті чого і отримується фінальна оцінка якості навчання.

В даній роботі K було обрано рівним 5, а у якості метрики якості використовувалась *точність класифікації*, тобто відношення кількості правильно класифікованих об'єктів до загальної кількості об'єктів

7. Отримані результати

Для проведення навчання алгоритмів та аналізу отриманих результатів було сформовано навчальну вибірку, яка мала наступний склад:

- людські дані – всього 350 об'єктів;
- дані ботів – по 500 об'єктів від кожної із розроблених моделей.

У табл. 2 наведено результати крос-валідації обраних алгоритмів.

Таблиця 2 – Результати навчання класифікаторів

Алгоритм	№ ітерації				Середнє значення
	1	2	3	4	
Логістична регресія	0.967	0.982	0.982	0.982	0.978
Градієнтний бустинг	0.994	0.994	1.000	0.994	0.995
Зважене голосування	0.997	0.994	1.000	1.000	0.998

Виходячи з результатів можна зробити висновок про те, що всі використані алгоритми забезпечують дуже високу якісь класифікації і можуть бути включені до складу системи розпізнавання ботів. Як і прогнозувалось, композиційні методи показали кращий результат ніж лінійні, хоча перевага і не є досить суттєвою.

Додатково, метод градієнтного бустингу дозволяє оцінити відносну важливість кожної ознаки. Виявилось, що для побудови кожного дерева прийняття рішень, що увійшло до підсумкової композиції було використано тільки 5 із 16 ознак, важливість яких була оцінена наступним чином:

- Середньоквадратичне відхилення кутового прискорення: 0.906;
- 0.25-квантиль кутової швидкості: 0.083;
- 0.75-квантиль швидкості: 0.006;
- Середнє значення швидкості: 0.003;
- 0.25-квантиль швидкості: 0.002.

Можна побачити, що для використання методу градієнтного бустингу достатньо обчислювати тільки 2 ознаки, оскільки їм було надано сумарну важливість близько 99%. Це дозволяє економити обчислювальні ресурси без зниження якості класифікації. Цікавості цьому факту додає ще і те, що дані у двовимірному просторі дуже легко візуалізувати. На рис. 3 зображено графік, де світлі точки є людськими даними, а темні є даними ботів.

З даного графіку можна зробити наступні висновки:

- Дані ботів добре відділяються від даних людей навіть найпростішими лінійними класифікаторами. Використання більш складних методів дозволяє звести помилку практично до нуля.
- Дані від обох моделей ботів розташовані дуже близько одне до одного, незважаючи на принципіально різну внутрішню будову. Це дозволяє припустити, що деякі характеристики руху ботів є досить сталими незалежно від конкретної реалізації, а також підтверджує те, що створення бота який би досить точно імітував рух справжньої людини є дуже складною задачею, витрати на створення якого, швидше за все, перевищать прибуток який він зможе принести спамерам.

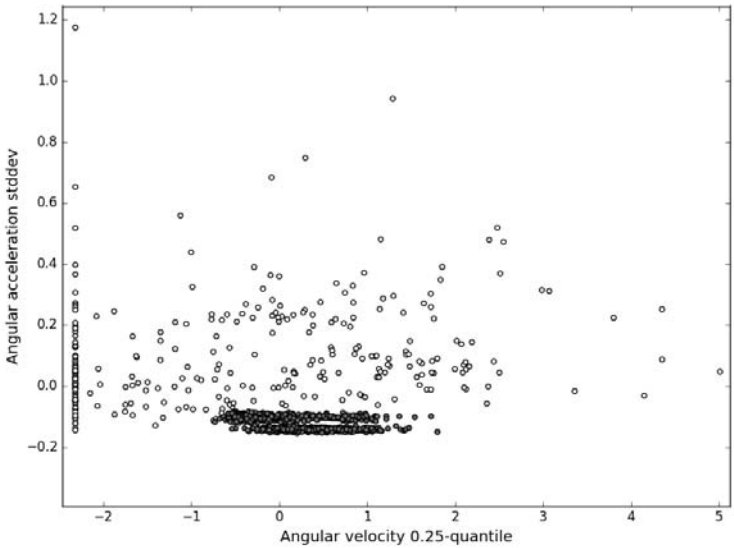

Рисунок 3 – Навчальна вибірка у просторі двох найбільш важливих ознак

Висновки. У даній роботі було запропоновано систему розпізнавання form-ботів на основі біометричних характеристик руху руки людини та використанням методів машинного навчання. Було виконано всі етапи вирішення задачі машинного навчання та отримано модель здатну виявляти ботів із точністю 99%, що є надзвичайно високим результатом.

У якості ознак об'єктів було використано статистичні показники основних характеристик руху, а саме швидкості, прискорення, кутової швидкості та кутового прискорення.

На основі аналізу наданої методом градієнтного бустингу інформації про важливість ознак було встановлено, що для якісної класифікації достатньо 5 із початкових 16 ознак, що дозволяє економити обчислювальні ресурси системи.

Також, було виконано візуалізацію зібраних даних у просторі двох ознак, які мають сумарну важливість 99%. На основі отриманого графіку було виявлено, що об'єкти різних класів насправді досить

добре розділяються навіть простими моделями. Також з'ясовано, що незважаючи на відмінності у внутрішній будові обох моделей ботів, об'єкти, що відносяться до різних моделей лежать у такому просторі дуже близько одне до одного, що може означати приховану подібність усіх можливих моделей ботів, хоч і перевірити на практиці це, швидше за все, неможливо.

Було запропоновано загальну архітектуру системи розпізнавання form-ботів, в основі якої лежатиме розроблена у даній роботі модель. Отримані результати свідчать про те, що така система буде здатна забезпечити високий рівень захисту від ботів і буде простою та зручною у використанні.

Список літератури

1 The Official CAPTCHA Site [Електронний ресурс]. – 2010. – Режим доступу до ресурсу: http://www.captcha.net.

2 reCAPTCHA: Easy on Humans, Hard on Bots [Електронний ресурс] – Режим доступу до ресурсу: http://www.google.com/recaptcha/intro/index.html.

3 Hastie T. The Elements of Statistical Learning Data Mining, Inference, and Prediction / T. Hastie, R. Tibshirani, J. Friedman. – 764 с. – (2).

4 Friedman J. Greedy Function Approximation: A Gradient Boosting Machine. / Jerome Friedman. // Annals of Statistics, Vol. 29. – 2001. – №5. – С. 1189–1232.

5 Марков А. А. Распространение закона больших чисел на величины, зависящие друг от друга / Андрій Андрійович Марков. // Известия физико-математического общества при Казанском университете, Том 15. - 2-а серія. - 1906. – С. 135–156.

Одержано 19.05.16