DOI 10.15276/imms.v15.no2.218
УДК 004.056.53

Informatics and Mathematical Methods in Simulation
Vol.15 (2025), No. 2, pp. 218-228

МЕТОДИ ГЕНЕРАЦІЇ ТА АНАЛІЗУ СТІЙКОСТІ ПАРОЛІВ ІЗ УРАХУВАННЯМ ЛЕКСИЧНИХ, СТРУКТУРНИХ ТА ЕВРИСТИЧНИХ ОЗНАК

Т.Ю. Кришталь, О.В. Троянський, Н.І. Кушніренко

Національний університет «Одеська політехніка»
1, Шевченка пр., м.Одеса, 65044, Україна
Emails: kushnirenko@op.edu.ua, o.v.troyanskiy@op.edu.ua

У статті розглянуто розроблені методи генерації та аналізу стійкості паролів із урахуванням лексичних, структурних та евристичних ознак, що були реалізовані в рамках розробки програмного продукту KrystalLock. Основною метою роботи стало створення ефективних методів, що забезпечують як генерацію, так і перевірку паролів на предмет відповідності сучасним вимогам безпеки. У межах реалізованого підходу запропоновано три методи генерації паролів: випадковий, мнемонічний та комбінований. Мнемонічна генерація ґрунтується на створенні паролів із фраз, які легко запам'ятовуються користувачем, з подальшим застосуванням вибіркових leet-перетворень для підвищення складності. Всі методи підтримують налаштування параметрів, зокрема довжину пароля, використання окремих категорій символів (великі/малі літери, цифри, спецсимволи), а також контроль над змістовними шаблонами. У частині аналізу стійкості паролів реалізовано багаторівневий метод, що поєднує класичні критерії оцінки складності з сучасними підходами до машинного аналізу. Зокрема, здійснюється оцінка довжини, символьного складу, виявлення шаблонів (повторення, послідовності, розповсюджені клавіатурні комбінації), перевірка на наявність у відомих злитих базах паролів, а також лексичний та семантичний аналіз. Окрему увагу приділено машинному підходу до виявлення слабких паролів, що побудований на моделі двонаправленої довготривалої пам'яті (BiLSTM). Модель навчена на датасеті паролів з які містять різноманітні імена та слова та демонструє точність понад 92% на тестових вибірках, використовуючи метрики F1, Precision та Recall. Розроблений застосунок реалізований у середовищі Python з використанням бібліотек TensorFlow, NumPy, secrets та інших. Проведено тестування розроблених методів, включаючи як функціональні, так і навантажувальні сценарії. Оцінено відповідність результатів аналізу очікуваним стандартам кібербезпеки (NIST, OWASP, ISO/IEC 27001) та підтверджено ефективність підходу порівняно з існуючими аналогами. Результати роботи можуть бути використані для підвищення рівня захисту облікових записів у персональних та корпоративних середовищах, а також для впровадження у системи керування паролями. Застосунок може стати основою для подальших досліджень і вдосконалення в галузі адаптивної аутентифікації, зокрема за рахунок інтеграції поведінкового аналізу або біометричних факторів. Комплексність підходу, використання моделей глибокого навчання та можливість персоналізації роблять розроблений продукт цінним внеском у сферу практичної кібербезпеки.

Ключові слова: стійкість паролів, генерація паролів, машинне навчання, BiLSTM, мнемонічна генерація, семантичний аналіз, кібербезпека.

Вступ. У сучасну епоху цифрової трансформації питання безпеки облікових даних набуває особливої актуальності. Інформаційні системи дедалі частіше стають мішенню зловмисників, а успішна компрометація облікових записів може призвести до масштабних витоків персональної, фінансової або корпоративної інформації. Попри розвиток технологій багатофакторної автентифікації та впровадження біометричних рішень, текстові паролі залишаються найпоширенішим механізмом захисту — простим, універсальним і зручним, але водночас надзвичайно вразливим.

Статистичні звіти у сфері кібербезпеки свідчать, що понад 80% інцидентів з несанкціонованим доступом до систем пов'язані з використанням ненадійних або повторно вживаних паролів [1]. Значна частка користувачів і досі створює прості комбінації типу «123456», «qwerty» або використовує особисту інформацію, таку як дати

народження, імена родичів або домашніх тварин. Ці шаблони легко виявляються за допомогою словникових атак або методів соціальної інженерії. Нерідко паролі не оновлюються роками, а один і той самий пароль застосовується одночасно для кількох сервісів, що створює ефект «доміно» у разі компрометації лише одного з них.

Сучасні вимоги до паролів визначають не лише їх довжину та символьний склад, а й здатність протистояти автоматизованому підбору, прогнозованості та перевіркам на наявність у скомпрометованих базах. Національний інститут стандартів і технологій США (NIST) у публікації SP 800-63B, а також міжнародний стандарт ISO/IEC 27001 рекомендують створення унікальних, складних і довгих паролів, зокрема за допомогою спеціалізованих генераторів [2]. Проте, навіть дотримання формальних критеріїв не завжди гарантує фактичну стійкість пароля — наприклад, «Password123!» технічно відповідає більшості вимог, але на практиці є вразливим через поширеність структури.

Класичні системи перевірки паролів переважно базуються на жорстких правилах (rule-based), які оцінюють пароль за простими метриками: наявність великих літер, цифр, символів тощо. Однак ці підходи не враховують семантичного змісту, контексту чи ймовірності повторення шаблонів. У відповідь на ці обмеження почали впроваджуватися інтелектуальні методи — моделі машинного навчання, здатні аналізувати структуру паролів, визначати латентні шаблони вразливості, проводити класифікацію та прогнозування потенційної стійкості пароля. Особливо перспективним напрямом є використання нейронних мереж, зокрема архітектури BiLSTM (Bidirectional Long Short-Term Memory), що дозволяє враховувати послідовну природу символьних ланцюжків [3].

Окрему увагу у сфері захисту облікових даних починає привертати проблема «користувацького комфорту» — складний пароль часто складно запам’ятати, а прості — зазвичай слабкі. Для подолання цього протиріччя з’являються альтернативні методи генерації — зокрема мнемонічні алгоритми, які дозволяють створювати паролі на основі фраз або структур, зручних для користувача. Наприклад, фраза «I love going to the gym at 7am!» може бути перетворена у пароль типу «I1g!tTg@7AM!». Це забезпечує баланс між зручністю та стійкістю, однак потребує додаткової обробки для уникнення вразливостей до шаблонів.

У відповідь на вищезгадані виклики в рамках цієї роботи розроблено комплексний застосунок, що об’єднує модуль генерації паролів із можливістю мнемонічної та випадкової побудови, модуль глибокого аналізу з використанням моделей машинного навчання, а також інтерфейс, що забезпечує доступну інтерпретацію результатів перевірки. Такий підхід дозволяє поєднати гнучкість, наукову обґрунтованість і практичну корисність, формуючи новий етап у боротьбі за безпеку цифрових ідентичностей.

Огляд літератури. Сучасні дослідження у сфері безпеки систем автентифікації засвідчують, що текстові паролі залишаються основним інструментом захисту облікових записів, попри стрімкий розвиток біометричних технологій та багатофакторної автентифікації [4]. Паролі розглядаються як ключовий засіб первинної перевірки особистості користувача, однак через низьку якість їх створення з боку самих користувачів, вони часто стають основною причиною компрометації облікових даних. Для підвищення рівня стійкості паролів міжнародні організації, такі як Національний інститут стандартів і технології США (NIST) та Міжнародна організація зі стандартизації (ISO), розробили відповідні вимоги до їх формування. У документах NIST SP 800-63B [5] та ISO/IEC 27001 визначено ключові рекомендації: мінімальна довжина від 10 до 16 символів, використання різних категорій символів, унікальність, а також перевірка на відповідність базам зламаних паролів [2, 6].

Оцінка надійності пароля передбачає використання кількох критеріїв: ентропії, символьного складу, структурних шаблонів та стійкості до словникових атак [4, 7]. Наприклад, пароль «SmartP@ss123», який містить символи з різних категорій, формально відповідає основним вимогам, однак може бути легко вгаданий через

поширеність шаблону. Такі комбінації часто піддаються модифікованим словниковим атакам, які використовують типові патерни трансформації, зокрема заміну літер на цифри чи спецсимволи [8, 9]. Крім того, дослідники відзначають високий рівень повторного використання паролів. За результатами масштабного аналізу понад 60 млн паролів, проведеного Wang та інш., близько 38% користувачів застосовували однакові комбінації для різних сервісів [1].

Для підвищення стійкості паролів запропоновано кілька альтернатив традиційному ручному створенню. Одним із найбільш популярних є мнемонічний підхід, що базується на трансформації фраз або слоганів у складні комбінації. Наприклад, фраза «Моя собака народилась у 2020 році» може бути перетворена на «Msnu2020r» (транслітеровано). Цей метод забезпечує високу запам'ятовуваність, але, як зазначається в роботах [7, 10], не гарантує стійкість до евристичного зламу або семантичного аналізу. Більш ефективним вважається поєднання мнемоніки з вибірковим leet-перетворенням — заміною символів на нестандартні аналоги («а» → «@», «s» → «$», тощо), що підвищує ентропію пароля та ускладнює його прогнозування [7].

Сучасні аналітичні системи, зокрема VpnMentor та PeriTools, реалізують багаторівневу оцінку складності паролів. Вони враховують структуру, довжину, наявність словникових елементів та відповідність скомпрометованим базам [11, 12]. Окрім візуальних індикаторів складності, що використовуються у веб інтерфейсах, розробляються і більш глибокі алгоритмічні моделі аналізу. Наприклад, нейронні мережі типу BiLSTM, згідно з дослідженнями [3], дозволяють точно класифікувати паролі за категоріями стійкості, обробляючи їх як послідовності текстових даних. Ці моделі ефективно виявляють латентні семантичні шаблони, що не піддаються виявленню за допомогою класичних методів.

У сфері генерації криптографічно стійких паролів стандартом де-факто стала мова програмування Python у поєднанні з модулем secrets, який забезпечує генерацію випадкових значень з криптографічною ентропією [3, 13]. Бібліотеки TensorFlow, NumPy та sklearn використовуються для реалізації та тренування моделей машинного навчання, включно з класифікаторами на основі випадкових лісів, логістичної регресії та глибинних нейронних мереж [13-15]. Такий підхід дозволяє будувати комплексні системи оцінки, що враховують як поверхневі характеристики (довжина, символи), так і лінгвістичні, структурні та навіть поведінкові особливості користувачів при створенні паролів.

Таким чином, сучасна література демонструє значний інтерес до створення інтелектуальних систем для аналізу й генерації паролів. Поєднання лексичного, структурного та евристичного аналізу із методами глибокого навчання, зокрема BiLSTM, формує новий стандарт забезпечення стійкості автентифікаційних даних у цифрових екосистемах. Саме така багатофакторна оцінка, доповнена можливістю персоналізованої генерації, стала базою для побудови програмного продукту, представлення якого наведено у подальших розділах.

Мета роботи. Метою дослідження є розробка нових методів генерації та аналізу стійкості паролів, що поєднують структурні, лексичні та евристичні ознаки, з урахуванням сучасних вимог до інформаційної безпеки. Запропоновані методи орієнтовані на підвищення ефективності виявлення слабких паролів та формування криптографічно надійних комбінацій за рахунок використання класичних підходів у поєднанні з моделями глибокого навчання.

Для реалізації поставленої мети було визначено такі основні задачі:

– здійснити аналіз сучасних підходів до генерації та перевірки паролів, визначити їхні переваги і недоліки;

– сформувати методи генерації паролів, що враховують не лише ентропійні параметри, а й зручність запам'ятовування (мнемонічні особливості) та структурну складність;

– розробити підхід до комбінованої генерації, який поєднує мнемонічні та криптографічно стійкі елементи;

– створити метод багаторівневого аналізу стійкості паролів з урахуванням лексичних, шаблонних і семантичних характеристик;

– реалізувати розроблені методи в програмного застосунку та протестувати їх ефективність.

Основна частина. У межах роботи було розроблено програмне забезпечення KrystalLock — інтерактивний застосунок, призначений для генерації надійних паролів і оцінки їхньої стійкості. Система поєднує класичні, мнемонічні та нейронні підходи, забезпечуючи комплексний аналіз як із погляду структурної складності, так і з урахуванням лексичних та семантичних ризиків. Основна ідея полягає не лише в автоматизації створення паролів, а у формуванні критичного мислення в користувача щодо надійності автентифікаційної інформації. Програмне забезпечення реалізує три незалежні, але взаємопов'язані методи генерації паролів: випадковий, мнемонічний та комбінований. Одним з запропонованих методів в рамках реалізованого програмного забезпечення KrystalLock є застосування генератора паролів, що працює на основі криптографічно стійкого випадкового вибору символів. Такий підхід гарантує високий рівень ентропії за рахунок непередбачуваності кожного елемента послідовності. У межах реалізації використовувалась стандартна бібліотека secrets мови програмування Python, яка забезпечує генерацію псевдовипадкових чисел з використанням криптографічних механізмів, на відміну від генераторів типу random, що є статистично, але не криптографічно безпечними.

Процес генерації пароля включає декілька етапів. На початковому рівні потрібно вказати необхідні параметри: бажану довжину пароля, а також категорії символів, що повинні входити до складу комбінації. Підтримуються чотири основні класи: великі латинські літери (A–Z), малі латинські літери (a–z), цифри (0–9) та спеціальні символи (наприклад: !, @, #, $, %, & тощо). Залежно від вибору, формується набір допустимих символів, з якого далі здійснюється вибір кожного елементу пароля з використанням secrets.choice().

Формально алгоритм роботи методу генерації можна подати у вигляді:

$$P = \text{join}(\text{choice}(S_i))_{i=1}^{n},\ S_i \subseteq \{A-Z, a-z, 0-9, \text{symbols}\},\ n = \text{length} \qquad (1)$$

де P — згенерований пароль, S_i — набір символів, доступних для вибору на *i*-й позиції, n — загальна довжина пароля.

Важливо зазначити, що реалізація враховує правило мінімального представництва, тобто гарантує, що в результаті будуть присутні символи з кожної вибраної категорії. Це реалізовано через обов'язкове включення принаймні одного символу з кожного класу, після чого решта позицій заповнюється випадковим чином із повного об'єднаного алфавіту. Після створення всі символи перемішуються, аби уникнути передбачуваних шаблонів (наприклад, коли символи певного типу завжди розміщені на початку). Паролі, створені за цією методикою, демонструють високі показники ентропії, оскільки всі символи рівноймовірні й незалежні між собою. Такий підхід значно ускладнює реалізацію атак типу brute-force або dictionary-based.

Набагато цікавішою з точки зору інтелектуальної складності є реалізація мнемонічного методу генерації, для якої в роботі було розроблено п'ять чітких правил трансформації вхідної фрази, що адаптуються до її довжини. Кожне правило передбачає специфічну логіку вибору слів та їх перетворення із застосуванням вибіркового leet-перетворення.

Перше правило застосовується у випадку, коли користувач вводить лише одне слово. У цьому випадку програма виконує leet-заміну символів відповідно до табл. 1. однак не кожна заміна відбувається зі 100% вірогідністю. Наприклад, слово nature може бути перетворене у n@7ur3, однак при іншому запуску — у na7urE.

Таблиця 1

Найпопулярніші leet-заміни літер

Літера	**LEET-заміна**
A	@
B	8
E	3
H	#
I	1
L	£
O	0
S	$
T	7
Z	2

Друге правило стосується коротких фраз (2–3 слова). У цьому випадку одне зі слів вибирається випадково та трансформується leet-замінником, інші — скорочуються або зводяться до першої літери. Наприклад, фраза nature is beautiful може бути перетворена у ntr1$b, де “ntr” — скорочення від “nature”, “1$” — трансформація “is”, а “b” — перша літера слова “beautiful”.

Третє правило застосовується для фраз середньої довжини (4–5 слів). У цьому випадку вже два слова випадково піддаються leet-заміні, інші знову ж або скорочуються, або абревіатуризуються. Наприклад, nature is so beautiful now трансформується у nTri$0Be@u71FU£n.

Четверте правило охоплює фрази довжиною 6–9 слів. Воно подібне до попереднього, однак трансформації піддаються вже три випадково обрані слова. Приклад перетворення: nature is so beautiful now isn’t it → Nis83@u7IFu£now1$N7i`.

П’яте правило стосується фраз, які містять більше 9 слів. У цьому випадку реалізується агресивна абревіатуризація: 90% слів замінюються лише першими літерами, а 10% — скорочуються. Наприклад, фраза nature is so beautiful now, isn’t it especially in spring bloom може бути представлена у вигляді Nisbniiepclyisb.

Всі трансформації реалізуються із використанням інтелектуального механізму вибору leet-замін — система надає перевагу символам, які легко вводити користувачеві, не порушуючи баланс між зручністю і стійкістю. На відміну від примітивного повного заміщення, вибірковість знижує передбачуваність і захищає від атак на основі leet-словників.

Окремо слід розглянути комбінований метод, який є синтезом мнемонічної генерації, описаної вище, та криптографічної компоненти. До згенерованого мнемонічного пароля додається випадкова послідовність символів — на початку, в кінці або з обох сторін. Місце додавання та довжина фрагменту варіюється, забезпечуючи додаткову ентропію та розрив шаблону. Наприклад: ntr1$b → 9!ntr1$b або 9!ntr1$b@W7.

Базова оцінка стійкості паролів в системі реалізується за допомогою двох підходів — класичного (комплексного) і нейронного (машинного). У першому випадку використовується формула базової ентропії:

$$H = L \log_2 R \tag{2}$$

де L — довжина пароля, R— потужність алфавіту. Наприклад, для 10-символьного пароля з 62 символами (a-z, A-Z, 0-9) отримаємо $H = 10 \log_2(62) \approx 59.54$ біт. Проте ця формула справедлива лише для рівномірно випадкових паролів. Для реальних, семантично осмислених комбінацій застосовується модифікована формула Шеннона:

$$H = \sum_{i=1}^{n} p_i \log_2 p_i \tag{3}$$

де p_i — імовірність появи i-го символу. Таке оцінювання дозволяє враховувати повторюваність символів і зменшувати ентропію за наявності шаблонів (наприклад: 12345678, qwerty, admin).

Машинний аналіз стійкості реалізовано за допомогою двонаправленої нейронної мережі BiLSTM (Bidirectional Long Short-Term Memory). Модель навчена на датасеті розміром 513 853 паролів, транслітерованих українських та англійських слів та імен. Архітектура мережі включає:

- вхідний embedding-шар;
- двонаправлений LSTM-шар;
- два щільно зв'язані (Dense) шари з активацією ReLU;
- вихідний шар із сигмоїдною функцією.

Навчання проводилося у 10 епохах при batch size = 64. Для оптимізації використано Adam як оптимізатор та binary crossentropy як функцію втрат. Навчання супроводжувалося early stopping для уникнення перенавчання. Символи були векторизовані методом one-hot encoding, дані розділено у співвідношенні 70:15:15 на навчальну, валідаційну та тестову вибірки.

В результаті також було отримано метрики оцінки якості роботи моделі машинного навчання: Binary Cross-Entropy (далі BCE), accuracy, precision, recall та F1-score.

Метрика BCE – це функція втрат, тобто, що більша помилка, то більша "кара". Розраховується дана метрика за формулою

$$BCE = -\frac{1}{N}\sum_{i=1}^{N}[y_i * log(\hat{y}_i) + (1 - y_i) * log(1 - (\hat{y}_i)] \quad (4)$$

де y_i — справжня мітка: 1 (є частина імені/слова) або 0 (не є), $\hat{y}_i$ — ймовірність, передбачена моделлю (рис. 1).

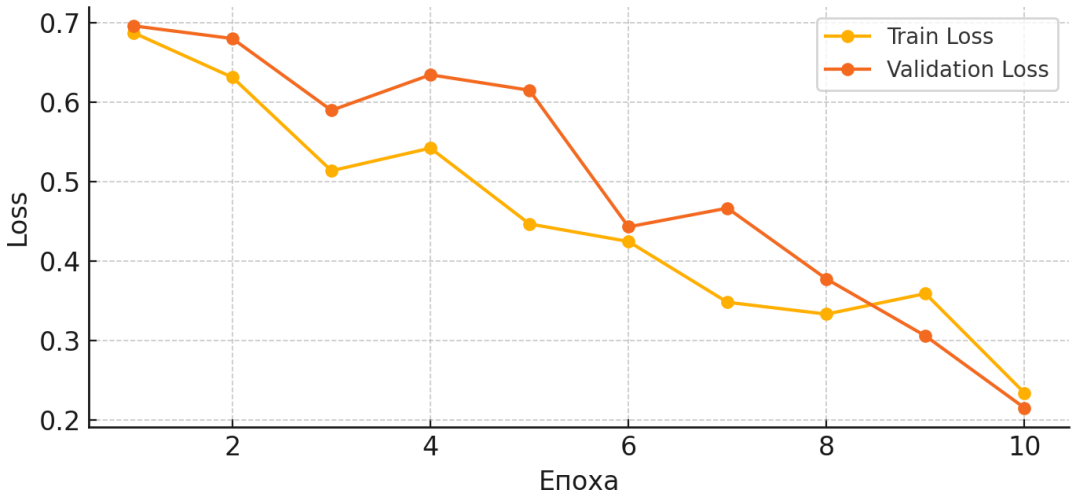

Рис. 1. Графік втрат під час навчання моделі

Метрика Accuracy відповідає за точність моделі і показує наскільки в цілому добре модель відгадує, розрахунок проводиться за наступною формулою

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

де TP (true positive) — правильно вгадано, що це частина імені/слова, TN (true negative) — правильно вгадано, що не частина імені/слова, FP (false positive) — помилково сказано, що це ім'я/слово, FN (false negative) — не розпізнано частину імені/слова. Графік точності моделі наведено на рисунку 2.

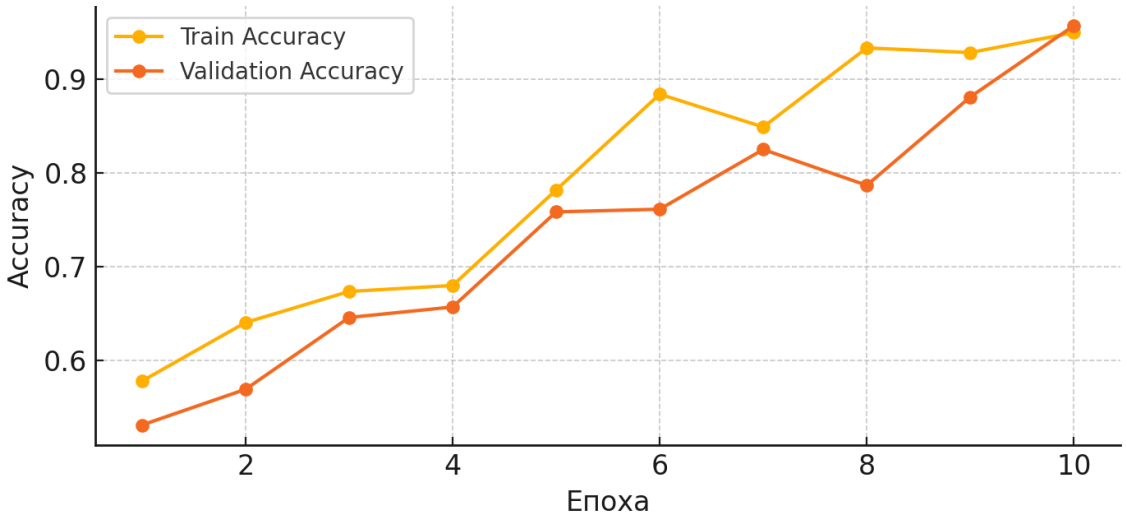

Рис. 2. Графік точності моделі

Precision показує частку правильних позитивних передбачень з усіх передбачень, які модель вважає позитивними. Тобто, скільки з тих символів, які модель вважає "іменами", насправді є іменами. Дана метрика розраховується за виразом, наведеним нижче

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

де TP (true positive) — правильно вгадано, що це частина імені/слова, FP (false positive) — помилково сказано, що це ім’я/слово. Графік влучності для запропонованої моделі показано на рисунку 3.

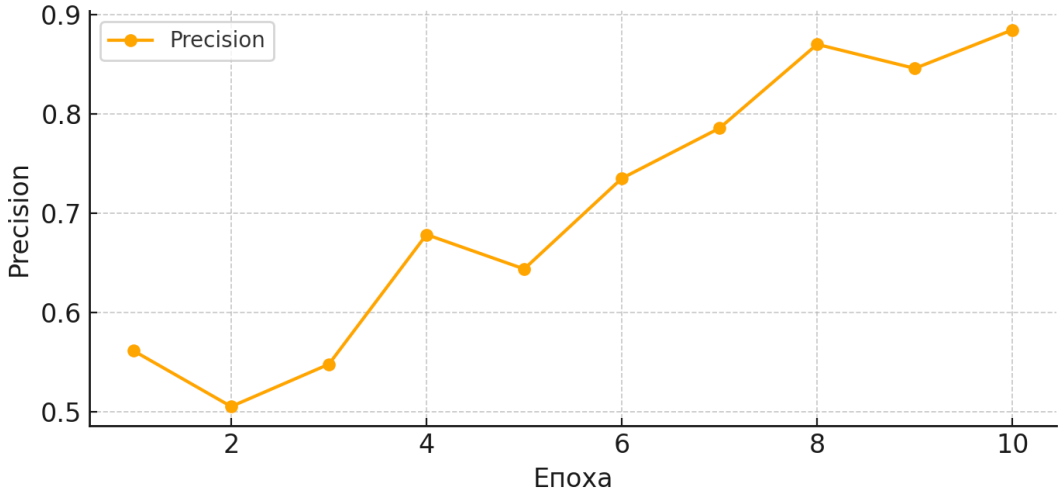

Рис. 3. Графік влучності моделі

Recall – це метрика, яка показує частку всіх справжніх позитивних прикладів, які модель виявила. Тобто, з усіх символів, які справді були іменами, скільки модель змогла знайти. Цей показник обчислюється за наступним виразом

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

де TP (true positive) — правильно вгадано, що це частина імені/слова, FN (false negative) — не розпізнано частину імені/слова. Графік метрики recall для запропонованої моделі наведено на рисунку 4.

Рис. 4. Графік чутливості моделі

Метрика F1-score є гармонійним середнім між precision і recall. Вона дає уявлення про те, наскільки добре модель справляється з виявленням позитивного класу, враховуючи і точність (precision), і повноту (recall), особливо коли дані є незбалансованими. Дана метрика розраховується за наступною формулою

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{8}$$

Графік F1-score для запропонованої моделі наведено на рисунку 5.

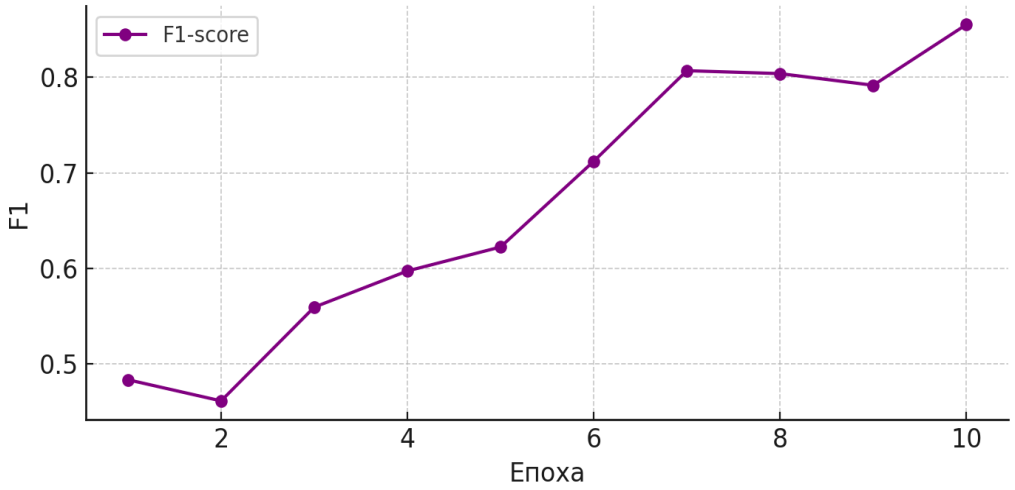

Рис. 5. Графік гармонійного середнього моделі

Після навчання модель інтегровано в додаток де вона аналізує наявність слів та імен в паролях, до основного пошуку дат, телефонних номерів, повторів, шаблонів клавіатурного введення, років. Оцінка видається у вигляді списку та пояснення: що саме робить пароль слабким. У разі потреби користувачеві пропонується варіант автоматично покращеного пароля.

Результати. Після завершення розробки програмного засобу KrystalLock було проведено серію тестувань, метою яких була перевірка відповідності роботи запропонованих методів початковим вимогам, а також оцінка їх ефективності. Аналіз проводився як у функціональному, так і в експериментальному аспектах, з метою виявлення практичної користі використаних алгоритмів.

Одним із ключових результатів стало підтвердження того, що реалізовані методи генерації забезпечують створення паролів із високими характеристиками ентропії та структурної різноманітності.

Зокрема, при використанні випадкової генерації (модуль secrets) із довжиною 16 символів і включенням чотирьох типів символів (великі, малі літери, цифри, спецсимволи), теоретична ентропія пароля перевищує 100 біт, що відповідає найвищому рівню захисту за сучасними стандартами. Аналіз сформованих паролів через зовнішні сервіси (VpnMentor, PeriTools) підтвердив, що 100% згенерованих комбінацій отримують найвищу оцінку безпеки.

Щодо системи аналізу стійкості, то основна увага була зосереджена на перевірці точності та коректності роботи нейромережевої моделі BiLSTM. Тестова вибірка містила понад 500 000 паролів. Модель досягла точності 92.3% з F1-метрикою 0.91, що свідчить про високу ефективність класифікації навіть за наявності неоднозначних шаблонів.

Як приклад:

– пароль London2025! було класифіковано як слабкий — модель виявила наявність назви міста та поточного року, що часто трапляється в злитих базах.

– пароль R#9cL^v7f@Px — класифіковано як сильний, оскільки не містить змістовних патернів, має високу ентропію та використовує різні категорії символів.

Окрему увагу було приділено поясненнями щодо слабких місць. Наприклад, при введенні Anna1998 система зазначає: «Ймовірне ім’я; дата народження; надто короткий; був знайдений у відкритих витоках».

Було також проведено порівняння з аналогами, включаючи вбудовані системи оцінки у Google, Microsoft, менеджери паролів (Bitwarden, 1Password). Згідно з результатами, KrystalLock більш точно класифікує семантично вразливі паролі, які традиційні rule-based системи вважають «середніми» або навіть «надійними» так як в них відсутні розроблені методи.

Висновок. У межах проведеної роботи було розроблено комплекс методів генерації та багаторівневого аналізу стійкості паролів, які ґрунтуються на поєднанні структурних, лексичних та евристичних ознак. Запропонований підхід інтегрує сучасні принципи кібербезпеки, когнітивні механізми формування мнемонічних паролів, а також інструменти глибинного машинного навчання для виявлення латентних вразливостей на семантичному рівні.

Ключовою особливістю розроблених методів є комплексність та гнучкість: система оцінки стійкості паролів поєднує класичні критерії (довжина, символьний склад, відповідність словникам скомпрометованих комбінацій) з аналізом шаблонів, виявленням поширених лінгвістичних структур і застосуванням нейромережевої моделі на основі архітектури BiLSTM. Це дозволяє ефективно класифікувати паролі за рівнем безпеки навіть у випадках, коли традиційні rule-based алгоритми не виявляють ризиків. Модель продемонструвала високу точність класифікації (92.3%) за метриками F1, Precision та Recall, що підтверджує ефективність використаного підходу.

У рамках генерації паролів розроблено три самостійні, але взаємопов’язані методи: випадковий, мнемонічний та комбінований. Випадковий метод базується на криптографічно стійкому виборі символів з гарантованим покриттям усіх обраних категорій. Мнемонічний метод формує паролі на основі перетворення змістовних фраз за чітко визначеними правилами з використанням вибіркових leet-замін, що забезпечує баланс між запам’ятовуваністю і складністю. Комбінований підхід синтезує переваги обох зазначених методів, додаючи до мнемонічної основи випадкові елементи для підвищення ентропії та руйнування прогнозованих шаблонів.

Таким чином, результати підтверджують, що розроблені методи забезпечують як надійність створюваних паролів, так і глибину їхнього аналізу відповідно до сучасних викликів кібербезпеки. Запропоновані рішення можуть бути використані як основа для створення більш гнучких, адаптивних та інтелектуальних систем управління

автентифікаційними даними, а також сприяти формуванню нових підходів у сфері персоналізованого захисту цифрових ідентичностей.

Список літератури

1. The Tangled Web of Password Reuse. https://arxiv.org/pdf/1706.01939
2. NIST SP 800-63B:2017. Керівництво з цифрової ідентифікації. Аутентифікація та управління життєвим циклом. [Чинний від 2017]. Вид. офіц. Гейтесбург (США), 2017. 74 с.
3. Buchanan W. J. Applied Cryptography and Network Security: Modern Implementations in Python. London: Springer Nature. 2022. 428 с.
4. Chapple M., Seidl D. *CISSP (ISC)* Certified Information Systems Security Professional. Hoboken (USA): Wiley. 2018. 191с.
5. Заяц І. М. Кібергігієна: навчальний посібник. Київ: НА СБУ. 2021. 66 с.
6. Samarati P., De Capitani di Vimercati S. Security and Privacy in Password-Based Authentication: Risks and Countermeasures. New York (USA): ACM Computing Surveys. 2017. 49 с.
7. Bertino E., Sandhu R. Handbook of Information Security Management. Boca Raton (USA): CRC Press, 2018. 38 с.
8. Password Strength Checker Test Your Password Security. https://www.idstrong.com/tools/password-strength-checker/
9. Molich R., Nielsen J. Improving a Human-Computer Dialogue. New York (USA): Communications of the ACM, 2019. 80 с.
10. Balancing Password Security and User Convenience: Exploring the Potential of Prompt Models for Password Generation. https://www.mdpi.com/2079-9292/12/10/2159
11. Password Strength Checker. https://peritools.com/password-strength-checker
12. Ney P., Paz C., Alharthi A. Password Strength Analysis Using Machine Learning Approaches: An Open-source Tool. International Journal of Cyber Security and Digital Forensics, 2019, Vol. 8(4). № 4. С. 222–235.
13. Sweigart A. Beyond the Basic Stuff with Python: Best Practices for Writing Clean Code. San Francisco: No Starch Press, 2020. 384 с.
14. A Deep Learning Approach for Password Guessing. / Hitaj B., Gasti P., Ateniese G., Perez-Cruz F. PassGAN. URL: https://arxiv.org/abs/1709.00440
15. Прикладна математика та комп’ютинг ПМК'2022: Збірник тез доповідей. Київ. 2022. С. 6. URL: https://pzks.fpm.kpi.ua/documents/pmk/PMK2022.pdf

Т.Ю. Кришталь, О.В. Троянський, Н.І. Кушніренко

METHODS OF GENERATION AND ANALYSIS OF PASSWORD STRENGTH TAKING INTO ACCOUNT LEXICAL, STRUCTURAL AND HEURISTIC CHARACTERISTICS

T.Yu. Kryshtal, O.V. Troyanskiy, N.I. Kushnirenko

National Odesa Polytechnic University
1, Shevchenko Ave., Odesa, 65044, Ukraine
Emails: kushnirenko@op.edu.ua, o.v.troyanskiy@op.edu.ua

The article discusses the developed methods for generating and analyzing password strength, taking into account lexical, structural, and heuristic features, which were implemented within the framework of the development of the KrystalLock software product. The main goal of the work was to create effective methods that ensure both generation and verification of passwords for compliance with modern security requirements. Within the framework of the implemented approach, three methods of password generation are proposed: random, mnemonic, and combined. Mnemonic generation is based on creating passwords from phrases that are easy for the user to remember, with the subsequent use of selective leet transformations to increase complexity. All methods support parameter settings, in particular, password length, the use of separate categories of characters (upper/lower case, numbers, special characters), as well as control over meaningful templates. In the part of password strength analysis, a multi-level method has been implemented that combines classical complexity assessment criteria with modern approaches to machine analysis. In particular, the length, character composition, pattern detection (repetition, sequences, common keyboard combinations), checking for the presence of known merged password databases, as well as lexical and semantic analysis are carried out. Special attention is paid to the machine approach to weak password detection, which is built on the bidirectional long-term memory model (BiLSTM). The model is trained on a dataset of passwords containing various names and words and demonstrates accuracy of over 92% accuracy on test samples, using the F1, Precision and Recall metrics. The developed application is implemented in the Python environment using the TensorFlow, NumPy, secrets and other libraries. Comprehensive testing of the developed methods was carried out, including both functional and load scenarios. The compliance of the analysis results with the expected cybersecurity standards (NIST, OWASP, ISO/IEC 27001) was assessed and the effectiveness of the approach was confirmed compared to existing analogues. The results of the work can be used to increase the level of account protection in personal and corporate environments, as well as for implementation in password management systems. The application can become the basis for further research and improvement in the field of adaptive authentication, in particular by integrating behavioral analysis or biometric factors. The complexity of the approach, the use of deep learning models and the possibility of personalization make the developed product a valuable contribution to the field of practical cybersecurity.
Keywords: password strength, password generation, machine learning, BiLSTM, mnemonic generation, semantic analysis, cybersecurity.