міжнародна науково-технічна конференція

БЕЗПЕКА СУЧАСНИХ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ СИСТЕМ

SMICS – 2025

Адаптивна модель для виявлення шкідливих URL з використанням машинного навчання

Петро Венгерський
Володимир Лесик
- Ivan Franko National University of Lviv

Track 2

Застосування технологій штучного інтелекту у кібербезпеці

Проблема

Зростання кількості кіберзагроз вимагає впровадження автоматизованих підходів для аналізу URL-адрес, оскільки традиційні методи не завжди ефективні проти нових атак. Шкідливі URL є поширеним інструментом для фішингових та інших зловмисних атак.

Мета дослідження

Розробка високоточного та швидкого методу аналізу URL-адрес з використанням алгоритмів Data Mining (DM) для виявлення шкідливих ресурсів, що підвищить рівень інформаційної безпеки.

Підхід до вирішення проблеми

Створення адаптивної моделі для виявлення шкідливих URL-адрес на основі ансамблевих методів машинного навчання, які поєднують переваги кількох класифікаторів для підвищення точності та стійкості.

Методологія

У дослідженні був обраний підхід керованої класифікації, де модель навчається на розмічених даних. Набори даних у цьому дослідженні - списки з сотнями тисяч URL. Для класифікації було отримано ознаки безпосередньо з URL. Загалом, у дослідженні було використано дев'ять алгоритмів класифікації.

Після класифікації було обрано 3 найбільш ефективних алгоритми для певного набору даних, які були згодом об’єднані у один ансамблевий класифікатор методом голосування.

Використані алгоритми класифікації

Деревоподібні методи: Decision Tree, Random Forest, Extra Trees Classifier.

Ансамблеві методи: Bagging Classifier, AdaBoost Classifier.

Градієнтний бустинг: XGBClassifier, LGBMClassifier, CatBoostClassifier.

Метод найближчих сусідів: KNeighborsClassifier.

Огляд та підготовка даних

Було використано три незалежні набори даних, кожен зі своєю структурою та призначенням. Набори необхідно було стандартизувати у єдиний формат для подальшого використання в моделях машинного навчання. Набори були очещені від повторів та іншого шуму і підведені під один стандарт з колонками “URL” та “тип”. Колонка “тип” має містити лише 2 типи - шкідливий та безпечний.

Кількість входжень у наборах

Набір даних №1: 453088 URL

Набір даних №2: 632508 URL

Набір даних №3: 326422 URL

Візуалізація прикладів входжень у наборах даних перед стандартизацією

5 rows × 2 columns

	url	type
0	br-icloud.com.br	phishing
1	mp3raid.com/music/krizz_kaliko.html	benign
2	bopsecrets.org/rexroth/cr/1.htm	benign
3	http://www.garage-pirenne.be/index.php?option=com_con…	defacement
4	http://adventure-nicaragua.net/index.php?option=com_m…	defacement

Приклади входжень для набору даних №1.

5 rows × 3 columns

	url	label	result
0	https://www.google.com	benign	0
1	https://www.youtube.com	benign	0
2	https://www.facebook.com	benign	0
3	https://www.baidu.com	benign	0
4	https://www.wikipedia.org	benign	0

Приклади входжень для набору даних №2.

5 rows × 2 columns

	URL	Label
0	nobell.it/70ffb52d079109dca5664cce6f317373782/login.S…	bad
1	www.dghjdgf.com/paypal.co.uk/cycgi-bin/webscrcmd=_hom…	bad
2	serviciosbys.com/paypal.cgi.bin.get-into.herf.secure.…	bad
3	mail.printakid.com/www.online.americanexpress.com/ind…	bad
4	thewhiskeydregs.com/wp-content/themes/widescreen/incl…	bad

Приклади входжень для набору даних №3.

Генерація ознак

Далі було сформовано ознаки, які можна витягти безпосередньо з URL для подальшого навчання класифікаторів. На основі огляду попередніх досліджень та експериментального підходу був складений перелік ознак, які демонстрували високу загальну кореляцію з цільовою змінною, тобто з показником, що характеризує клас URL (безпечний чи шкідливий).

Фінальні ознаки

- ***letters_count***: кількість літер в URL.
- ***digits_count***: кількість цифр в URL.
- ***special_chars_count***: кількість спеціальних символів.
- ***shortened***: індикатор скороченої URL.
- ***abnormal_url***: ознака аномальності URL.
- ***secure_http***: наявність протоколу HTTPS.
- ***root_domain***: аналіз кореневого домену.
- ***rv***: коефіцієнт репутації.
- ***isGlobal***: індикатор належності до міжнародного домену верхнього рівня.

Навчання моделей та огляд результатів

Першим етапом процесу навчання моделей є розподіл даних на навчальний та тестовий набори. Використовується стандартний підхід, при якому дані діляться у співвідношенні 70:30, де 70% даних виділяється для навчання моделі, а 30% — для оцінки її ефективності на нових, раніше невідомих прикладах.

Після розподілу дані для навчання передаються раніше згаданим моделям: DecisionTreeClassifier, RandomForestClassifier, ExtraTreesClassifier, AdaBoostClassifier, BaggingClassifier, XGBClassifier, LGBMClassifier, CatBoostClassifier та KNeighborsClassifier.

Результати класифікації, отримані за допомогою цих алгоритмів, оцінювалися за метриками точності, повноти, позитивної точності та F1-міри. Кожен алгоритм показав різні рівні ефективності залежно від використаного набору даних, а результати могли змінюватися в залежності від типу шкідливого URL, характеристик набору даних та розподілу навчальних прикладів.

Результати навчання моделей

Classifier performance:

	Classifier	Accuracy	Recall	Precision	F1-Score
6	cat_boost	0.911910	0.911910	0.913109	0.911641
7	extra_trees	0.911659	0.911659	0.912529	0.911433
1	random_forest	0.910636	0.910636	0.911509	0.910406
4	gradient_boost_xgb	0.909102	0.909102	0.910734	0.908768
8	bagging_classifier	0.904677	0.904677	0.905770	0.904393
5	gradient_boost_lgb	0.898502	0.898502	0.901637	0.897924
0	decision_tree	0.880574	0.880574	0.880602	0.880586
2	ada_boost	0.854003	0.854003	0.859233	0.852659
3	k_neighbors	0.833606	0.833606	0.833652	0.833626

Результати моделювання для набору даних №1.

Classifier performance:

	Classifier	Accuracy	Recall	Precision	F1-Score
6	cat_boost	0.979477	0.979477	0.979632	0.979468
5	gradient_boost_lgb	0.978022	0.978022	0.978296	0.978008
4	gradient_boost_xgb	0.977658	0.977658	0.977811	0.977648
2	ada_boost	0.973846	0.973846	0.974866	0.973807
7	extra_trees	0.973368	0.973368	0.974060	0.973337
1	random_forest	0.973222	0.973222	0.973604	0.973201
8	bagging_classifier	0.972555	0.972555	0.972884	0.972535
0	decision_tree	0.959369	0.959369	0.959390	0.959373
3	k_neighbors	0.590053	0.590053	0.598517	0.586598

Результати моделювання для набору даних №2.

Classifier performance:

	Classifier	Accuracy	Recall	Precision	F1-Score
6	cat_boost	0.841746	0.841746	0.841631	0.839531
7	extra_trees	0.841309	0.841309	0.840672	0.839563
1	random_forest	0.840166	0.840166	0.839396	0.838548
4	gradient_boost_xgb	0.837687	0.837687	0.837844	0.835107
5	gradient_boost_lgb	0.828718	0.828718	0.828480	0.826027
8	bagging_classifier	0.828317	0.828317	0.827796	0.825861
0	decision_tree	0.789519	0.789519	0.789509	0.789514
3	k_neighbors	0.775688	0.775688	0.774472	0.774926
2	ada_boost	0.772672	0.772672	0.776488	0.762897

Результати моделювання для набору даних №3.

Ансамблювання найефективніших моделей

Для подальшого підвищення ефективності був застосований VotingClassifier, який об'єднав прогнози трьох найкращих класифікаторів. Цей ансамбль працює за принципом жорсткого голосування, де кожен базовий класифікатор «голосує» за належність URL-адреси до одного з класів, а остаточне рішення приймається на основі більшості голосів. Такий підхід дозволив досягти невеликого, але стабільного покращення продуктивності ~ 0.4%

```
Voting Classifier Performance:
Accuracy: 0.9153192725582476
Recall: 0.9153192725582476
Precision: 0.9163995346409288
F1-Score: 0.9150787772206201
```

Результати моделювання VotingClassifier для набору даних №1

```
Voting Classifier Performance:
Accuracy: 0.9787006212934664
Recall: 0.9787006212934664
Precision: 0.9789092319497146
F1-Score: 0.9786891452385303
```

Результати моделювання VotingClassifier для набору даних №2

```
Voting Classifier Performance:
Accuracy: 0.848297601776727
Recall: 0.848297601776727
Precision: 0.8479760919225958
F1-Score: 0.84650278481803
```

Результати моделювання VotingClassifier для набору даних №3

Висновки

Проведене дослідження демонструє ефективність та перспективність застосування методів керованого машинного навчання для високоточного виявлення шкідливих URL-адрес. Ключовим результатом дослідження є підтвердження, що використання ансамблевого підходу перевершує за ефективністю окремі класифікатори. Такий метод забезпечує високу точність без складної підготовки даних. Експерименти показали, що окремі алгоритми втрачають продуктивність на різних наборах даних, тоді як ансамбль, зокрема метод голосування, забезпечує стабільне зростання точності та надійності.

Це дозволяє створити інструмент, здатний навчатися на різнорідних наборах даних для точної класифікації без тривалої адаптації. Це має безпосереднє практичне значення, оскільки надає надійний механізм для посилення інформаційної безпеки та захисту користувачів від кіберзагроз.

Література

[1] A. Basit, M. Zafar, X. Liu, A. R. Javed, Z. Jalil, K. Kifayat, A comprehensive survey of AI-enabled phishing attack detection techniques, Telecommunication Systems 76, 2021, 1–16.
[2] S. C. Jeeva, E. B. Rajsingh, Intelligent phishing URL detection using association rule mining, Human-centric Computing and Information Sciences 6(1), 2016, 1–19.
[3] N. Koppula, V. P. Ganti, P. Gandhe, URL-Based Phishing Detection, International Journal of Recent Technology and Engineering (IJRTE) 9(1), 2020, 1872–1875.
[4] В. А. Лахно, В. П. Малюков, І. В. Малюкова, О. Аткелди, О. В. Криворучко, А. М. Десятко, К. В. Степашкіна, Модель аналізу стратегій при динамічній взаємодії учасників фішингових атак, Кібербезпека: освіта, наука, техніка 4(20), 2023, 1–15.
[5] T. G. Dietterich, Ensemble methods in machine learning, in: Multiple Classifier Systems, Springer, Berlin, 2000, pp. 1–15.
[6] V. N. Vapnik, Statistical Learning Theory, Wiley, New York, NY, 1998, 736 p.
[7] J. Friedman, T. Hastie, R. Tibshirani, The Elements of Statistical Learning, Springer, New York, NY, 2001, 745 p.
[8] IBM, Decision trees. URL: https://www.ibm.com/topics/decision-trees.
[9] L. Nautiyal, M. Ram, P. Malik, Machine Learning for Cyber Security, Springer, Singapore, 2022, 250 p.
[10] E. Bertino, W. Susilo, X. Chen, Cyber Security Meets Machine Learning, Springer Nature, Singapore, 2022, 163 p.
[11] J. H. Friedman, Greedy function approximation: A gradient boosting machine, Annals of Statistics 29(5), 2001, 1189–1232.