

ЗАСТОСУВАННЯ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ ДЛЯ ВИЯВЛЕННЯ АНОМАЛІЙ МЕРЕЖНОГО ТРАФІКУ

О. П. Подвисоцька¹, С. О. Носок¹

¹ Навчально-науковий Фізико-технічний інститут

Анотація

У даній роботі розглядається застосування алгоритмів машинного навчання для виявлення аномалій в мережному трафіку, що є однією з ключових задач в області кібербезпеки. Розглядаються різні методи кластеризації, зокрема K-means, DBSCAN, Isolation Forest і COPOD, для ідентифікації незвичайної поведінки, що може вказувати на зловмисні дії.

Ключові слова: машинне навчання, мережний трафік, виявлення аномалій, детектування викидів

Вступ

Виявлення аномалій у мережному трафіку відіграє критичну роль у захисті інформаційних систем від зловмисників та інших неавторизованих втручань. Використання методів машинного навчання в цій області дозволяє автоматизувати та оптимізувати процес детекції, адаптувавши системи до постійно змінюваних візрів трафіку та загроз. У цій роботі розглядаються особливості застосування різних алгоритмів машинного навчання для ідентифікації аномалій, від простих кластеризацій до складних алгоритмів, що засновані на ізоляції та копулах.

1. Способи виявлення аномалій за допомогою машинного навчання

Загалом задачі машинного навчання поділяються на задачі з учителем (регресія та класифікація) та навчання без учителя (кластеризація). Аномаліями в контексті кібербезпеки можна вважати ту активність, яка вказує на можливі дії зловмисників. А позаяк аномалії не можна віднести до передбачуваної поведінки системи, задача їх виявлення зводиться до пошуку викидів (outliers) в задачі кластеризації. Розглянемо деякі відомі алгоритми:

K-means. Найвідомішим алгоритмом кластеризації є метод k-середніх. Спочатку визначається кількість k кластерів, на які будуть поділятися дані. Центри цих кластерів на першому етапі визначаються випадковим чином. Далі кожен об'єкт з набору даних відноситься до того кластера, центроїд якого знаходиться найближче до нього. Після цього відбувається оновлення значень центроїдів - їм призначається середнє значення всіх об'єктів, віднесених до відповідних кластерів. Наведені пункти повторюються до стабілізації центроїдів (мінімізації сумарної квадратичної відстані між точками кластера та центрами цих кластерів) або до досягнення максимальної заданої кількості ітерацій. Проте для пошуку ви-

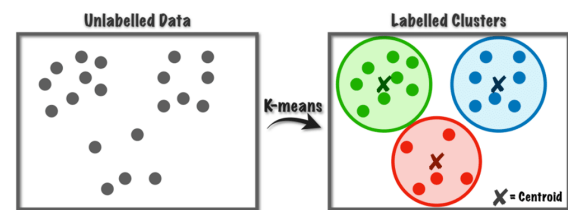


Рис. 1. Візуалізація кластеризації даних методом K-means

кидів представлений алгоритм є найкращим рішенням, тому що такі дані можуть спотворити кластери. Тому для заданої задачі слід розглянути інші методи, наведені нижче.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise). Базується на щільності [1] та ідентифікує кластери на наборах даних, які містять шум і викиди. Для цього методу необхідними є два наступні параметри: epsilon (ϵ) – максимальна відстань між двома точками, за якої вони будуть вважатися «сусідами», та minimum points (minPts) – мінімальна кількість точок у радіусі ϵ , необхідна для того, щоб точка вважалася частиною скупчення. Самі ж точки поділяються:

- кореневі (core points) – містять щонайменше minPts сусідніх точок в околі ϵ ;
- граничні (border point) – мають менше ніж minPts сусідніх точок, але знаходяться в ϵ -околі основної точки;
- шумові (noise points).

Спершу вибирається довільна точка та визначається її тип. Якщо вона є центральною, починається процес розширення кластера шляхом додавання всіх точок з ϵ -околу. Граничні точки також додаються в кластер, проте їхні околи далі не досліджуються, тому формування кластерів на них завершується. Процес триває доти, доки кожній точці

не буде призначено її тип. Відповідно, ті точки, що не потрапили в жодний кластер, визначаються як шумові.

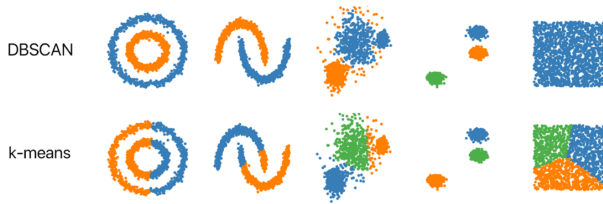


Рис. 2. Візуалізація кластеризації даних методом DBSCAN

Isolation Forest. Ліс ізоляції відрізняється від інших методів тим, що ґрунтується не на формуванні кластерів, а саме на пошуку аномалій серед даних, позаяк викиди простіше відокремити від нормальних спостережень. Алгоритм ізолює дані, вибираючи випадковий атрибут і довільно розбиваючи діапазон значення. Процес повторюється рекурсивно, створюючи дерево. Багато таких дерев формують ліс для ізоляції кожної точки в наборі даних. Таким чином, чим швидше те чи інше спостереження відокремлюється від інших, тим більша ймовірність того, що воно є аномалією.

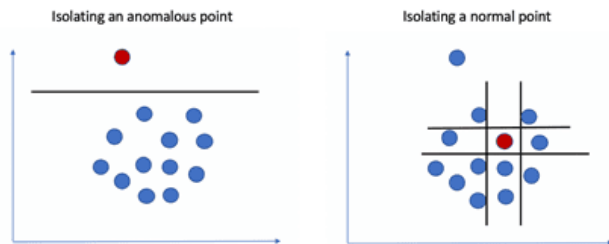


Рис. 3. Візуалізація кластеризації даних методом Isolation Forest

Isolation Forest є особливо корисним тоді, коли на великих наборах даних потрібно швидко ідентифікувати аномалії, які є рідкісними та не мають чітко визначеної структури чи взаємозв'язків з нормальними об'єктами.

COPOD (Copula-based Outlier Detection). Відносно новий алгоритм навчання, особливості якого полягає у відсутності гіперпараметрів [2]. Адже в задачі пошуку аномалій ми далеко не завжди можемо коректно встановити значення цих параметрів для моделі – зазвичай їх призначають на основі попереднього досвіду, інформації про певний набір даних, статистично або просто випадковим чином. Перш за все, COPOD вивчає маргінальні розподіли кожної змінної в наборі даних. Маргінальний розподіл – це розподіл ймовірностей для окремої змінної, ігноруючи вплив або наявність інших змінних. Для кожної змінної визначається, як часто вона приймає певні значення, і формується розподіл ймовірностей цих значень. Далі COPOD вико-

ристовує копули – це функцію, яка дозволяє моделювати взаємозв'язки між різними змінними, зберігаючи при цьому їх індивідуальні маргінальні розподіли. Копули використовуються для побудови сумісного розподілу ймовірностей для двох або більше змінних, використовуючи тільки їх маргінальні розподіли. На наступному етапі обчислюється оцінка аномальності для кожного спостереження. Отже, ще одна перевага даного методу в тому, що він вважає об'єкт аномалією лише тоді, коли він є таким за більшістю параметрів, а не лише за одним виміром.

2. Використання на реальних даних

Було взято дані мережного трафіку за певний проміжок часу. Загальна кількість записів у вибірці складає 1 млн записів. Попередньо було проведено очищення цих даних і їх кодування задля зручної обробки, а також для приховування реальних IP-адрес. Отримані результати можна побачити нижче:

```
kmeans = KMeans(n_clusters=2, random_state=42)
df['Cluster'] = kmeans.fit_predict(df)

centroids = kmeans.cluster_centers_

distances = np.linalg.norm(df - centroids[df['Cluster']], axis=1)
df['Distance'] = distances

threshold = np.percentile(df['Distance'], 99)
df['Outliers'] = df['Distance'] > threshold

df[df['Outliers']]
```

	Time	Source	Destination	Protocol	Length	Cluster	Distance	Outliers
973136	11392.281587	25	3550	34	1434	1	4174.923162	True
973137	11392.281587	25	3550	34	1434	1	4174.923162	True
973138	11392.281587	25	3550	34	1434	1	4174.923162	True
973139	11392.281587	25	3550	34	1434	1	4174.923162	True
973140	11392.281587	25	3550	34	1434	1	4174.923162	True
...	...	...	...	...	...	...	...	...
999990	11793.904997	25	3550	34	66	1	4376.490141	True
999996	11793.920779	19	3551	34	60	1	4380.334611	True
999997	11793.920779	19	3551	34	60	1	4380.334611	True
999998	11793.920946	3550	18	34	60	1	4268.560561	True
999999	11793.920973	3550	18	34	60	1	4268.560583	True

10000 rows × 8 columns

Рис. 4. Результати аналізу алгоритмом K-means

З наведених результатів можна побачити, що велику роль відіграють гіперпараметри всіх алгоритмів. Від їхніх значень зокрема залежить кількість об'єктів, які ми готові розглядати як потенційні аномалії.

3. Проблеми пошуку аномалій

Як і в будь-якій задачі в процесі моніторингу мережного трафіку є ризик виникнення хибнопозитивного чи хибнонегативного детектування. Це пов'язано з тим, що дані, які передаються, вочевидь не є постійними – їхні обсяг, адреси відправників та отримувачів, типи пакетів та інші параметри можуть змінюватися залежно від того, що вимагається від си-

```
from sklearn.cluster import DBSCAN

dbscan = DBSCAN(eps=5, min_samples=8)
clusters = dbscan.fit_predict(features)

df['Cluster'] = clusters

df['Outliers'] = df['Cluster']
df = df.drop(['Cluster'], axis = 1)

df[df['Outliers'] == -1]
```

	Time	Source	Destination	Protocol	Length	Distance	Outliers
0	0.000000	3549	1235	38	357	2997.129400	-1
1	0.000115	3549	1235	38	357	2997.129302	-1
2	0.000140	1237	3550	38	698	3075.761893	-1
3	0.000172	3148	3550	34	60	3003.864563	-1
5	0.000468	3549	3149	34	60	2992.141114	-1
...	...	...	...	...	...	...	...
999947	11793.890972	1859	3550	23	400	3788.510512	-1
999952	11793.893117	3549	24	34	426	4250.177134	-1
999975	11793.901251	3549	1856	34	186	3733.895663	-1
999982	11793.903309	25	3550	34	530	4363.611360	-1
999989	11793.904260	3549	1856	23	457	3726.807396	-1

283381 rows x 7 columns

Рис. 5. Результати аналізу алгоритмом DBSCAN

```
from sklearn.ensemble import IsolationForest

iso_forest = IsolationForest(contamination=0.01, random_state=42)

features = df.values

iso_forest.fit(features)

outliers = iso_forest.predict(features)

df['Outliers'] = outliers

dec()
df[df['Outliers'] == -1]
```

	Time	Source	Destination	Protocol	Length	Outliers
55	0.742796	645	644	21	92	-1
56	0.743621	3802	3806	18	84	-1
57	0.743991	645	1911	18	64	-1
58	0.744769	3802	3806	18	84	-1
59	0.745039	645	1911	18	64	-1
...	...	...	...	...	...	...
999852	11792.360851	3598	3550	10	356	-1
999862	11792.367502	3549	3600	10	454	-1
999876	11792.734589	3549	28	10	228	-1
999885	11792.793686	2551	3550	10	356	-1
999895	11792.800655	3549	2554	10	454	-1

10000 rows x 6 columns

Рис. 6. Результати аналізу алгоритмом Isolation Forest

стеми. Саме тому процес детектування варто базувати на оцінці ймовірності та корекції відповідного порогу чутливості. Це дозволить відсіяти частину хибних спрацювань. Проте таке рішення майже то-

чно знизить відсоток знаходження аномалій фактичних.

Висновки

Алгоритми машинного навчання демонструють значний потенціал у виявленні аномалій мережно-

```
from pyod.models.copod import COPOD

model = COPOD(contamination=0.01)

model.fit(features)

outliers = model.predict(features)

df['Outliers'] = outliers

# dec()
df[df['Outliers'] == 1]
```

	Time	Source	Destination	Protocol	Length	Cluster	Distance	Outliers
39	0.239700	3796	3801	33	64	0	3352.530019	1
40	0.336619	26	3550	22	90	0	3683.289856	1
41	0.336868	3549	25	22	90	0	3599.706767	1
55	0.742796	645	644	21	92	0	3448.259623	1
56	0.743621	3802	3806	18	84	0	3355.153896	1
...	...	...	...	...	...	...	...	...
999995	11793.909442	1859	3550	34	60	1	3805.464300	1
999996	11793.920779	19	3551	34	60	1	4380.334611	1
999997	11793.920779	19	3551	34	60	1	4380.334611	1
999998	11793.920946	3550	18	34	60	1	4268.560561	1
999999	11793.920973	3550	18	34	60	1	4268.560583	1

10000 rows x 8 columns

Рис. 7. Результати аналізу алгоритмом COPOD

го трафіку, пропонуючи широкий спектр методів для різних типів даних та умов. Алгоритми, такі як Isolation Forest та COPOD, зокрема, можуть бути ефективним інструментом у швидкому виявленні аномалій без потреби в ручному налаштуванні параметрів. Однак, існують проблеми з хибнопозитивними та хибнонегативними результатами, що вимагають подальших досліджень для оптимізації порогів чутливості та адаптації моделей під конкретні мережні середовища.

Перелік використаних джерел

1. Schindler T., Schlicht S., Thoben K.-D. Towards Benchmarking for Evaluating Machine Learning Methods in Detecting Outliers in Process Datasets. — 2023. — Dec. — URL: <https://www.mdpi.com/2073-431X/12/12/253>.
2. COPOD: Copula-Based Outlier Detection / Z. Li, Y. Zhao, N. Botta, C. Ionescu, X. Hu. — 2020. — Sept. — URL: <https://arxiv.org/abs/2009.09463>.