DOI 10.36074/logos-15.11.2024.038

ОЦІНЮВАННЯ РЕЛЕВАНТНОСТІ МЕТОДУ МУРАШИНИХ КОЛОНІЙ (ACO) ДЛЯ ВИРІШЕННЯ ВИКОРИСТАННЯ В СИСТЕМАХ ВИЯВЛЕННЯ ВТОРГНЕННЯ

Іщенко Артем Андрійович[1], Гальчинський Леонід Юрійович[2]

1. здобувач вищої освіти Фізико-технічного інституту
Національний технічний університет України «Київський політехнічний інститут» ім. Ігоря Сікорського, УКРАЇНА

2. канд. техн. наук, доцент, доцент кафедри інформаційної безпеки
Національний технічний університет України «Київський політехнічний інститут» ім. Ігоря Сікорського, УКРАЇНА
ORCID ID: 0000-0002-3805-1474

Анотація. *В епоху цифрових технологій та стрімкого наукового зростання нові відкриття відбуваються кожен день, а з цим, об'єми даних, що курсують по мережам світу збільшуються настільки, що становиться технічно неможливо швидко обробляти весь потік вхідної інформації. У зв'язку з цим, на перший план виходить питання оптимізації вже існуючих технологій, що в тому числі це є актуальним для задач інформаційної безпеки, адже вплив та важливість інформаційного простору продовжують лише зростати, як і ризики, пов'язані з можливим незаконним заволодінням чи модифікацією будь яких чутливих даних.*

Вступ. У сучасному світі зростаюча залежність від інформаційних технологій і мережевих систем призводить до збільшення загроз кібербезпеки. Системи виявлення вторгнень є ключовими компонентами в забезпеченні інформаційної безпеки, оскільки вони дозволяють виявляти і запобігати несанкціонованому доступу до інформаційних ресурсів. Однак, ефективність систем виявлення вторгнень значною мірою залежить від якості алгоритму, який використовується для навчання моделей виявлення атак. Невдалий вибір алгоритму може призвести до низької точності та великої кількості помилкових спрацьовувань, що, в свою чергу, впливає на загальну надійність системи, тому життєво важливим є системний та впорядкований

підхід до їх відбору. У свою чергу релевантність обраного алгоритму можна оцінити на множині ознак, які використовуються для навчання моделей виявлення атак. Незважаючи на наявність певного набору алгоритмів, які використовуються для навчання при виявленні атак, питання який з найкращий є відкритим. Тому пошуки нових підходів, зокрема евристичних і відповідне їх оцінювання залишається актуальним.

1. Метод мурашиних колоній

Метою дослідження є визначення релевантності використання алгоритму мурашиних колоній (Ant colony optimization algorithm), у порівнянні з іншими існуючими методами, для вирішення проблеми оптимізації ознак в системах виявлення вторгнення, використання якого, як показують існуючі дослідження, цілком можливо як для цієї задачі [4], та і для інших складних завдань, таких як криптоаналіз [7]. Евристичні методи грунтуються на підсвідомому мисленні і характеризуються неусвідомленим (інтуїтивним) способом дій для досягнення усвідомлених цілей. Евристичні методи ще називають методами інженерного (винахідливого) творчості. Простіше кажучи, евристика - це не повністю математично обґрунтований (або навіть «не зовсім коректний»), але при цьому практично корисний алгоритм. Зважаючи на запит про пошук все більш ефективних методів для IDS, доцільно провести дослідження застосування евристичних методів, зокрема методу мурашиних колоній.

Метод мурашиних колоній є одним з найбільш ефективних методів рішення пошукових задач комбінаторного напряму. Ідея метода складається в рішення задачі оптимізації шляхом запровадження непрямого зв'язку між автономними агентами. Незважаючи на примітивність дій мурах, діяльність всієї колонії досить розумна, тобто мурашина колонія, по суті, є природною багатоагентній системою. Непрямий обмін - стігмержі (stigmergy), являє собою рознесене в часі взаємодія, при якому одна особина змінює деяку область навколишнього середовища, а інші використовують цю інформацію пізніше, коли в неї потрапляють. «Мурашині» алгоритми є ефективним способом вирішення завдань пошуку і оптимізації, які дозволяють графову інтерпретацію, що підтверджується емпіричними дослідженнями. До переваг варто віднести можливість застосування до широкого спектру завдань і гарантовану збіжність. З недоліків можна відзначити сильну залежність від початкових параметрів налаштування алгоритму, які підбираються тільки виходячи з практичного досвіду. Однак для визначення можливостей цього підходу можливо тільки шляхом порівняльного аналізу з відомими методами.

2. Оптимізація ознак в системах виявлення вторгнень

Вибір ознак – це процес, який вибирає підмножину ознак з усіх початкових ознак, що є цілком важливим через велику кількість можливих

нерелевантних ознак у наборі даних. Також, вибір ознак можна розглядати як задачу оптимізації для оптимальної підмножини ознак, які краще задовольняють поточні потреби для виконання специфічних завдань. Якість оптимізації при виборі ознак за своєю сутністю є NP-складною задачею, яку можна виміряти за допомогою певних критеріїв оцінювання, але на сьогоднішній день не існує оптимального еталонного рішення для пошуку оптимальної підмножини ознак. Типовий процес відбору ознак включає:

1. Створення підмножини ознак.
2. Оцінка обраної підмножини ознак.
3. Визначення критеріїв завершення.
4. Оцінювання результатів роботи алгоритму[2].

Процедури вибору підмножин ознак можуть бути реалізовані за допомогою різних алгоритмів, кожен з який вибирає підмножини ознак для оцінки на основі певної стратегії пошуку. Найпопулярнішими методами, які можуть використовуватися для даної задачі, є методи: Ant colony optimization [3], Recursive Feature Elimination, Tree-Based Feature Selection, Gradient Boosting, SelectK Best [4], та методи випадкової або константного вибору ознак.

За результатами проведених емпіричних досліджень скороченої версії датасету KDD Cup 99 (10%) для задачі оптимізації ознак в системах виявлення вторгнення за допомогою алгоритму мурашиних колоній, було отримано наступні результати:

Таблиця 1

Результати оцінки за усіма ознаками без застосування CAO

Результати	RandomForest
Accuracy	0.9997705
Precision	0.9997706
Recall	0.9997705
F1 Score	0.9997706

[авторська розробка]

Таблиця 2

Результати оцінки із застосуванням CAO для оптимізації ознак

Кіл-кість ітера-цій	**Accuracy**		**Precision**		**Recall**		**F1 Score**		**Кількість вибраних ознак**	
	max	**average**	**max**	**avera-ge**	**max**	**average**	**max**	**average**	**min**	**avera-ge**
50	0.999694	0,984624	0,999694	0,984319	0,999694	0,984624	0,999694	0,981283	24	35
Найкраща ітерація:										
№ 26	0.999694		0.999694		0.999694		0.999694		27	

[авторська розробка]

При виконанні програми спостерігалась тенденція до поступового зменшення середньої кількості обраних ознак з кожною ітерацією при відносному збереженні точності оцінювання, що свідчить про поступову адаптацію алгоритму до виконання задачі через зміни коефіцієнту феромону на кожній з ознак.

У фінальному найкращому варіанті після проходження 50 ітерацій, з 41 ознаки було обрано наступні 27 ознак:

Таблиця 3

Ознаки, обрані у найкращій ітерації

Обрані ознаки	Важливість ознаки
dst_bytes	0.222379
logged_in	0.128552
dst_host_srv_diff_host_rate	0.090148
src_bytes	0.087216
srv_count	0.081350
protocol_type	0.057033
diff_srv_rate	0.055224
service	0.052592
flag	0.048621
same_srv_rate	0.037743
dst_host_same_srv_rate	0.033775
dst_host_srv_count	0.026399
dst_host_diff_srv_rate	0.023303
dst_host_srv_rerror_rate	0.011033
duration	0.010026
hot	0.008849
dst_host_serror_rate	0.007583
srv_serror_rate	0.007121
dst_host_rerror_rate	0.004568
serror_rate	0.003014
srv_rerror_rate	0.002440
is_guest_login	0.000685
num_root	0.000223
num_file_creations	0.000087
num_access_files	0.000021
urgent	0.000014
is_host_login	0.000000

[авторська розробка]

Таблиця 4

Таблиця результуючого рівня феромону

№	Ознака	Коефіцієнт результуючого феромону	№	Ознака	Коефіцієнт результуючого феромону
1	duration	0.88	22	is_guest_login	0.87
2	protocol_type	0.88	23	count	0.40
3	service	0.54	24	srv_count	0.52
4	flag	0.47	25	serror_rate	0.87
5	src_bytes	0.88	26	srv_serror_rate	0.33
6	dst_bytes	0.88	27	rerror_rate	0.41
7	land	0.79	28	srv_rerror_rate	0.88
8	wrong_fragment	0.63	29	same_srv_rate,	0.47
9	urgent	0.76	30	diff_srv_rate,	0.63
10	hot	0.88	31	srv_diff_host_rate	0.76
11	num_failed_logins	0.76	32	dst_host_count	0.76
12	logged_in	0.87	33	dst_host_srv_count	0.14
13	num_compromised	0.59	34	dst_host_same_srv_rate	0.53
14	root_shell	0.55	35	dst_host_diff_srv_rate	0.88
15	su_attempted	0.79	36	dst_host_same_src_port_rate	0.42
16	num_root	0.66	37	dst_host_srv_diff_host_rate	0.88
17	num_file_creations	0.84	38	dst_host_serror_rate	0.88
18	num_shells	0.41	39	dst_host_srv_serror_rate	0.81
19	num_access_files	0.87	40	dst_host_rerror_rate	0.87
20	num_outbound_cmds	0.78	41	dst_host_srv_rerror_rate	0.15
21	is_host_login	0.88			
22	is_guest_login	0.87			
23	count	0.40			
24	srv_count	0.52			

[авторська розробка]

Аналізуючи результати, можна зробити висновки, що значення важливості ознак, корелюють із значеннями результуючих значень коефіцієнту феромону, отже, можна сказати, що алгоритм дійсно підходить для реалізації задачі оптимізації ознак в системах виявлення вторгнень.

3. Багатокритеріальний аналіз рішень та його методи

Станом на сьогодні, складні завдання ухвалення рішень розв'язуються за допомогою математичних моделей, статистичних методів, економічних теорій та комп'ютерних технологій, що дозволяють автоматизувати розрахунки та оцінювання ймовірних розв'язків.

Одним із варіантів вирішення є використання багатокритеріального аналізу (далі – MCDA) що одним із найефективніших методів у цій сфері та містить різні методи, кожен з яких має свої особливості. Цей підхід враховує як якісні, так і кількісні критерії, необхідні для визначення оптимального розв'язку. Але, з іншого боку, його використання викликає додаткові труднощі саме на етапі аналізу, в частині необхідності правильного підбору критеріїв, від яких залежить правильна та ефективна оцінка або порівняння [1].

4. Використання методу ELECTRE III у задачі визначення релевантності методів оптимізації ознак у системах виявлення вторгнень

Враховуючи існуючи методи MCDA та з урахуванням особливостей задачі оцінювання релевантності алгоритмів для оптимізація ознак в системах виявлення вторгнень, було обрано метод Elimination Et Choice Traduisant la Realite (ELECTRE), а саме, його модифікацію ELECTRE III у зв'язку з тим, що цей метод використовує нескладну та зрозумілу логіку, та має можливість порівняння альтернатив навіть при сильній розбіжності критеріїв, але при цьому також має інструментарій для обмеження порівнювальної між різними варіантами, тобто має нечіткий характер прийняття рішень [5]-[6].

Таблиця 5

Підготовлені за результатами дослідження дані для оцінювання

	ACO	Recursive Feature Elimina-tion	Tree-Based Feature Selection	Gradient Boosting	SelectK Best	Random Feature Selection	Constant Feature Selection
Time	0,65	0,77	0,97	0,96	0,98	0,99	0,98
Accuracy	0,99	0,99	0,99	0,99	0,99	0,90	0,80
Precision	0,99	0,99	0,99	0,99	0,99	0,92	0,64
Recall	0,99	0,99	0,99	0,99	0,99	0,90	0,80
F1Score	0,99	0,99	0,99	0,99	0,99	0,92	0,72
Effective-ness	0,63	0,65	0,85	0,99	0,23	0,23	0,10
Number of signs	0,90	0,90	0,95	0,99	0,90	0,90	0,85

[власна розробка]

Для обчислення ефективності виконання алгоритму спочатку було нормалізовано дані за допомогою Min-Max нормалізації, перетворивши кожне значення критеріїв «Час», «Кількість обраних ознак» та «Точність» в діапазон від 0 до 1.

Формула для нормалізації:

$$x_n = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

Після цього для обчислення ефективності моделі використано формулу:

$$\text{Ефективність} = a * Accuracy_n + b * (1 - Time_n) + c * K_n \quad (2)$$

де a, b та c - вагові коефіцієнти, що відображають важливість кожного нормалізованого значення критерію (0.6, 0.1 та 0.3 відповідно)

Час та кількість обраних ознак було перетворено, корелюючи значення від найкращого до найгіршого, використовуючи проміжок від 0 до 1, де значення кращого результату відповідно знаходиться ближче до 1.

Зазначені дані для оцінок алгоритмів було оброблено за допомогою програмного забезпечення XLSSTAT на базі MS Excel.

За результатами опитування експертів було визначено наступна вага критеріїв для оцінювання:

Таблиця 6

Вага критеріїв для оцінювання, визначена експертами

Time	0,2
Accuracy	0,9
Precision	0,45
Recall	0,45
F1Score	0,5
Effectiveness	0,62
Number of signs	0,41

[власна розробка]

Таблиця 7

Значення порогів Indifference, Preference та Veto

Indifference	Preference	Veto
0,20	0,25	0,35
0,01	0,02	0,10
0,02	0,04	0,10
0,02	0,04	0,10
0,01	0,02	0,10
0,09	0,11	0,20
0,04	0,06	0,20

[власна розробка]

Результати:

Таблиця 8

Матриця відповідності

a/b	ACO	Recursive Feature Elimina-tion	Tree-Based Feature Selection	Gradient Boosting	SelectK Best	Random Feature Selection	Constant Feature Selection
ACO	1,000	1,000	0,710	0,652	0,943	0,943	0,943
Recursive Feature Elimina-tion	1,000	1,000	0,766	0,708	0,989	0,977	0,989
Tree-Based Feature Selection	1,000	1,000	1,000	0,766	1,000	1,000	1,000
Gradient Boosting	1,000	1,000	1,000	1,000	1,000	1,000	1,000
SelectK Best	0,824	0,824	0,766	0,708	1,000	1,000	1,000
Random Feature Selection	0,173	0,173	0,115	0,057	0,348	1,000	1,000
Constant Feature Selection	0,115	0,115	0,057	0,057	0,115	0,115	1,000

[власна розробка]

Таблиця 9

Матриця достовірності

a/b	ACO	Recursive Feature Elimina-tion	Tree-Based Feature Selection	Gradient Boosting	SelectK Best	Random Feature Selection	Constant Feature Selection
ACO	1,000	1,000	1,000	1,000	0,000	0,000	0,000
Recursive Feature Elimina-tion	1,000	1,000	1,000	1,000	0,000	0,000	0,000
Tree-Based Feature Selection	0,000	0,000	1,000	1,000	0,000	0,000	0,000
Gradient Boosting	0,000	0,000	0,766	1,000	0,000	0,000	0,000
SelectK Best	0,943	0,989	1,000	1,000	1,000	0,004	0,000
Random Feature Selection	0,943	0,977	1,000	1,000	1,000	1,000	0,000
Constant Feature Selection	0,943	0,989	1,000	1,000	1,000	1,000	1,000

[власна розробка]

Таблиця 10

Матриця випередження

a/b	ACO	Recursive Feature Elimina-tion	Tree-Based Feature Selec-tion	Gradient Boosting	SelectK Best	Random Feature Selection	Constant Feature Selection
ACO	I	P	P	P	P	P	P
Recursive Feature Elimina-tion	NP	I	P	P	P	P	P

Продовження табл. 10

a/b	ACO	Recursive Feature Elimina-tion	Tree-Based Feature Selec-tion	Gradient Boosting	SelectK Best	Random Feature Selection	Constant Feature Selection
Tree-Based Feature Selection	NP	NP	I	I	P	P	P
Gradient Boosting	NP	NP	I	I	P	P	P
SelectK Best	NP	NP	NP	NP	I	P	P
Random Feature Selection	NP	NP	NP	NP	NP	I	P
Constant Feature Selection	NP	NP	NP	NP	NP	NP	I

[власна розробка]

Де:

1. a P b означає, що дія a є кращою перед дією b.
2. a NP b означає, що дія a не є кращою перед дією b.
3. a R b означає, що дія a не порівнянна з дією b.
4. a I b означає, що дія a байдужа до дії b.

За результатами дослідження можемо зробити наступні висновки:

✓ GradientBoosting є найкращим алгоритмом;

✓ Tree-BasedFeatureSelection зайняв друге місце;

✓ ACO та RecursiveFeatureElimination є однаковими по ефективності та разом займають 3 місце;

✓ SelectKBest – займає 4 місце;

✓ RandomFeatureSelection – передостанній за ефективністю;

✓ ConstantFeatureSelection є найгіршим.

Висновки. У роботі було визначено релевантність використання алгоритму мурашиних колоній шляхом детального дослідження роботи самого алгоритму при виконанні задачі оптимізації вибору ознак в системах виявлення вторгнень та порівнянні та оцінюванні отриманих результатів із сімома іншими існуючими методами оптимізації вибору ознак в системах виявлення вторгнень за допомогою методу ELECTRE III.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ:

[1] Taherdoost, H., & Madanchian, M. (2023). Multi-Criteria Decision Making (MCDM) Methods and Concepts. *Encyclopedia*, *3*(1), 77-87. URL: https://doi.org/10.3390/encyclopedia3010006.

[2] Huan Liu & Lei Yu. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, *17*(4), 491–502. URL: https://doi.org/10.1109/tkde.2005.66.

[3] Mehdi Hosseinzadeh Aghdam & Peyman Kabiri (2016). Feature Selection for Intrusion Detection System Using Ant Colony Optimization. *International Journal of Network Security*, 18(3), 420-432. URL: http://ijns.jalaxy.com.tw/contents/ijns-v18-n3/ijns-2016-v18-n3-p420-432.pdf.

[4] Kocyigit, E., Korkmaz, M., Sahingoz, O. K., & Diri, B. (2024). Enhanced feature selection using genetic algorithm for machine-learning-based phishing URL detection. *Applied sciences*, *14*(14), 6081. URL: https://doi.org/10.3390/app14146081.

[5] Galchynsky, L., Graivoronskyi, M., & Dmytrenko, O. (2021). Evaluation of Machine Learning Methods to Detect DoS / DDoS Attacks on IoT. *CEUR Workshop Proceedings*, 3241, 225–236, URL: https://ceur-ws.org/Vol-3241/paper21.pdf .

[6] Тостоган, Є. Г., & Гальчинський, Л. Ю. (2021). Вибір інструментів оцінювання кіберризиків для організацій на основі багатокритеріального аналізу. *XXII Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених теоретичні і прикладні проблеми фізики, математики та інформатики*, (22), 176–178, URL: https://ela.kpi.ua/handle/123456789/69948

[7] Іщенко А. А., & Кубайчук О. О. (2021). The relevance of the ant algorithm as a tool for cryptanalysis. *Науково-практична конференція студентів, курсантів, аспірантів, докторантів та молодих вчених «Актуальні питання застосування інформаційно-телекомунікаційних систем»*, (1), 50.