УДК 004.056.53:004.032.26 **DOI:** 10.31673/2412-9070.2024.011419

В. О. СОСНОВИЙ, аспірант;
Н. О. ЛАЩЕВСЬКА, канд. техн. наук, доцент;
В. О. ВЛАСЕНКО, канд. техн. наук, доцент,
Державний університет інформаційно-комунікаційних технологій, Київ

РОЗРОБЛЕННЯ СТРУКТУРИ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ РОЗПІЗНАВАННЯ АТАК У КОМП'ЮТЕРНИХ СИСТЕМАХ

Інтеграція комунікаційних мереж та Інтернету промислового контролю в автоматизованих системах керування (АСК) підвищує їхню вразливість до кібератак, що призводить до руйнівних наслідків. Традиційні системи виявлення вторгнень (СВВ) здебільшого покладаються на попередньо визначені моделі та навчаються переважно на вже відомих кібератаках, а це означає, що традиційні СВВ не можуть впоратися з невідомими атаками. Крім того, більшість СВВ не зважають на незбалансованість наборів даних АСК, тому страждають від низької точності та високого рівня псевдопозитивних результатів під час використання. У статті запропоновано метод виявлення вторгнень NCO-двошаровий DIFF_RF-OPFYTHON для АСК, до складу якого входять модулі NCO, двошарові модулі DIFF_RF і модулі OPFYTHON. Виявлений трафік було розділено на три категорії двошаровим модулем DIFF_RF: відомі атаки, невідомі атаки та звичайний трафік. Далі відомі атаки класифіковано модулем OPFYTHON на конкретні атаки відповідно до особливостей трафіку атаки. Було використано модуль NCO, щоб поліпшити вхідні дані моделі та підвищити її точність. Результати показали, що запропонована модель краще пристосована для виявлення вторгнень, зокрема XGboost і SVM. Викриття невідомих атак також є значним. Точність набору даних, використаного в цій статті, досягає 98,13%. Рівень виявлення невідомих і відомих атак досягає відповідно 98,21% і 95,1% .

Ключові слова: машинне навчання; виявлення вторгнень; безпека мережі; модель протидії; автоматизована система керування.

Вступ

Постановка проблеми. Автоматизована система керування (АСК) була відносно незалежною та рідко приєднувалася до інтернету, а отже, зосереджувалася на доступності та швидкості системи, ігноруючи безпеку і стаючи вразливою до атак [1]. Хоча було проведено багато досліджень щодо методів виявлення відомого трафіку атак в історичному мережному трафіку, з появою невідомих атак в інтернеті важко забезпечити безпеку АСК лише виявленням відомих атак [2]. Крім того, трафік атаки зазвичай становить тільки невелику частину всього трафіку в АСК, а нерівномірний розподіл або дисбаланс даних також ускладнює створення моделі виявлення вторгнень [3]. Тому вкрай важливо ефективно виявляти невідомі атаки та розробляти модель для незбалансованих даних у АСК.

Аналіз останніх досліджень і публікацій. Методи виявлення вторгнень загалом можна поділити на такі, що ґрунтуються на неправильному використанні, та сформовані на основі аномалій [4]. Методи на основі зловживання виявляють трафік атаки в ICS, порівнюючи функцію виявленого трафіку з відомим трафіком атаки історичного трафіку. Модель, побудована на зловживанні, наприклад Snort [5], може ефективно виявляти відомі атаки з низьким рівнем помилкових позитивних результатів (FPR), але вона не може ідентифікувати невідомі атаки. Методи на основі аномалій виявляють трафік атаки в АСК через порівняння виявленого мережного трафіку зі звичайним мережним трафіком. Модель на основі аномалій [6] може виявляти весь аномальний трафік, включно з відомими та невідомими атаками. Однак ця модель має високий FPR і не може класифікувати атаки за ознакою трафіку атаки. Алгоритм ізольованого лісу (Isolation Forests) [7] став одним із найчастіше використовуваних алгоритмів для раннього виявлення вторгнень завдяки своїй простоті та швидкості. Проте такий алгоритм здатен лише відфільтрувати викиди, які не можуть відповідати вимогам дедалі складнішого середовища ICS.

Метод, запропонований у [8], останнім часом стає все більш популярним як метод виявлення вторгнень на основі аномалій. Цей метод виявляє трафік атаки завдяки моніторингу та вивченню звичайних дій і подій на АСК [9-11]. Чим більша кількість шаблонів вивчається, тим точніше розглядуваний метод може виявляти аномальні дії та події. У статті [12] подано алгоритм для виявлення вторгнень у мережі SDN. Цей метод зорієнтовано на пошук аномалій, використовуючи витягнуті характеристики потоків, наприклад, тривалість, тип протоколу, сервіс. У [13] запропоновано полегшену систему виявлення та запобігання вторгненням, механізм ухвалення рішень базується на даних кадру та адресах джерела/приймача.

Крім того, існує багато поширених алгоритмів виявлення вторгнень, зокрема метод на основі механізму аналізу великих даних Apache Spark, який реалізує виявлення вторгнень у Sparks MLlib за допомогою алгоритму k-means [14]. Алгоритм k-means ґрунтується на вилученні змінних ознак. Коли в наборі даних багато шуму, алгоритму k-means важко досягти хороших результатів. Розроблений метод виявлення вторгнень для класифікації набору даних CBB поєднанням генетичного алгоритму для оптимального вибору ознак і LSTM з RNN описано в [15]. Спрощена залишкова мережа (S-ResNet) становить кілька каскадних і спрощених залишкових блоків. S-ResNet оптимізує проблему, пов'язану з тим, що залишкові мережі ResNets мають тенденцію переналаштовуватися на низькорозмірні та маломасштабні набори даних [16]. Однак, порівняно з машинним навчанням, кількість параметрів і розмір набору даних, потрібних для глибинного навчання, значні, що збільшує вартість навчання. Метод лінійного дискримінантного аналізу LDA покращує аналіз, а потім використовує його для зменшення розмірів ознак [17]. Проте такі методи не можуть виявити невідомі атаки. SVM і OCSVM також є поширеними алгоритмами виявлення вторгнень, вони схожі тим, що не можуть визначити категорію атаки.

Мета статті — розробити продуктивну, ефективну і надійну модель виявлення вторгнень на основі машинного навчання, яка дасть високі результати щодо точності класифікації, функції виявлення невідомих атак і показників оцінювання порівняно з наявними методами та моделями виявлення вторгнень. Показати перевагу запропонованої моделі та підтвердити, що саме ця модель забезпечує високу точність виявлення атак.

Основна частина

Щоб усунути обмеження, передбачені для методів та моделей виявлення вторгнень на основі неправильного використання та аномалій, а також незбалансованих навчальних зразків, пропонується двошарова модель виявлення вторгнень NCO DIFF_RF-OPFYTHON для ICS. На першому кроці normal_ DIFF_RF--OPFYTHON використовується модуль DIFF_RF, швидко фільтруючи аномальний трафік через порівняння виявленого мережного трафіку зі звичайним трафіком. Цей крок робить наступний модуль OPFYTHON більше непридатним для великої кількості звичайного трафіку, оскільки звичайний трафік відфільтровано модулем normal_DIFF_RF, який усуває дисбаланс навчальних зразків. Далі використовується модуль OPFYTHON, щоб порівняти відфільтрований аномальний трафік з трафіком відомої атаки. На цьому етапі відфільтрований аномальний трафік класифікується відповідно до ознаки трафіку атаки, що розв'язує проблему першого етапу з виявлення трафіку атаки, і неможливості класифікувати її ознаками.

Існує два типи проблем у наведеному normal_DIFF_RF-OPFYTHON:

1) точність моделі значною мірою залежить від точності normal_DIFF_RF;

2) ідентифікація невідомих атак, коли невідомий трафік атаки з'являється у виявленому мережному трафіку, OPFYTHON зі свого боку класифікує невідомий трафік атаки як найбільш подібний до відомої атаки.

Для поліпшення методу виявлення вторгнень normal_DIFF_ RF-OPFYTHON пропонуємо двошарову модель виявлення вторгнень NCO DIFF_RF-OPFYTHON. Аналізуючи експериментальні результати, метод вирішує дисбаланс навчальних вибірок із досить високою точністю та прийнятним рівнем виявлення невідомих атак.

Метод виявлення вторгнень сформовано з трьох модулів: модуля NCO, модуля дворівневого DIFF_RF і модуля OPFYTHON. Загалом він охоплює такі кроки:

- попереднє оброблення вихідних даних мережного трафіку ICS, включно з перетворенням категоріальних змінних у числові змінні, нормалізацію та розділення даних;
- використання алгоритму NCO для оптимізації вхідної структури двошарового модуля DIFF_RF;
- навчання двошарового модуля DIFF_RF і модуля OPFYTHON за допомогою навчального набору;
- інтеграцію модуля NCO, дворівневого модуля DIFF_RF і модуля OPFYTHON для отримання повної моделі виявлення вторгнень (NCO-double-layer DIFF_RF-OPFYTHON) та оптимізації параметрів;
- використання набору для тестування та підтвердження здійсненності, надійності та переваги запропонованого методу.

У мережному трафіку АСК часто виникає дисбаланс даних. Дисбаланс навчальних вибірок чутливо впливає на точність моделі. Запропонована модель виявлення normal_DIFF_RF-OPFYTHON використовує модуль двійкової класифікації normal_DIFF_RF, щоб спочатку фільтрувати трафік атаки у виявленому мережному трафіку, а потім застосовує модуль мультикласифікації OPFYTHON для класифікації трафіку атаки. Отже, модулю OPFYTHON не потрібно підганяти великий звичайний трафік під час фази навчання, він усуває дисбаланс наборів даних, спричинений великим звичайним мережним трафіком.

Алгоритм DIFF_RF будує модель, підбираючи трафік тієї самої позначки, щоб визначити, чи належить виявлений мережний трафік цій позначці. Наприклад, під час фази навчання DIFF_RF відповідає лише звичайному мережному трафіку серед усього мережного трафіку. На етапі тестування визначається, чи є виявлений мережний трафік нормальним.

На стадії навчання модуля DIFF_RF потрібно визначити два метапараметри: ψ — кількість підмножин S, випадково взятих із навчальних вибірок, і t — так звана кількість дерев у лісі. DIFF_RF $F = \{T(S_1), T(S_2), ..., T(S_t)\}$ сформовано з $\{S_1, S_2, \ldots, S_t\}$, узятих випадковим чином. Розміри з високою ентропією можна порівняти з шумом, тому алгоритм DIFF_RF віддає перевагу використанню вхідних змінних середньої та низької ентропії для навчання моделі. Щоб здобути ентропію кожної вхідної змінної, потрібно обчислити ентропію всіх вхідних змінних за допомогою гістограм. Рівняння (1) унаочнює процес обчислення ентропії кожної вхідної змінної, рівняння (2) — процес нормалізації ентропії.

Розрахунок ентропії:

$$EE_i = \frac{-1}{\log_2(bins)} \sum_{k=1}^{bins} b_k / |S| \log_2 \left(b_k / |S| \right) \quad \forall i \in \{1,2,...,d\}, \tag{1}$$

де *bins* — кількість бінів на гістограмі; S — підмножина, випадково взята з навчальних вибірок.

Нормалізація ентропії:

$$H_{q1} = 1 - \frac{H_q}{\log_2 bins} \quad \forall i \in \{1,2,...,d\}, \tag{2}$$

де H_q — ентропія кожної вхідної змінної.

Під час тестування алгоритм DIFF_RF класифікує виявлений мережний трафік через обчислення балів і встановлення відповідного порогового параметра. Якщо оцінка менша за порогове значення, виявлений трафік вважається таким самим типом, що й навчальна вибірка. Інакше — це вважається різним мережним трафіком. Процес обчислення оцінки $\delta_T(x)$ кожного дерева DIFF можна записати у вигляді

$$\delta_T(x) = 2^{-\alpha \frac{1}{d} \sum_{i=1}^{d} \left(\frac{(x(i) - M_S(i))}{\sigma_S} \right)^2}, \tag{3}$$

де $M_S(i)$ — центроїд i-ї вхідної змінної в навчальних вибірках; $x(i)$ — значення i-ї вхідної змінної в тестових вибірках; σ_S — стандартне відхилення навчальних вибірок.

Оцінка, розрахована алгоритмом DIFF, є середнім математичним сподіванням кожного дерева в лісі. Визначення оцінки *pwas*(x) DIFF_RF, де E — середнє математичне сподівання, можна подати так:

$$pwas(x) = -E(\delta_T(x)). \tag{4}$$

Алгоритм OPFYTHON перетворює навчальний набір у повний граф, а повний граф утворено з кількох вузлів і дуг, що з'єднують вузли. Кожен зразок у навчальному наборі відповідає вузлу в повному графі, а дуга між двома вузлами відповідає відстані між двома сусідніми вузлами. Чим більша вага дуги між сусідніми вузлами, тим менша схожість між ними.

У процесі моделювання нехай Z буде набором даних, який утворено з наборів для навчання та тестування, позначених відповідно як Z_1 та Z_2. Можна визначити граф $G = (V, A)^3$, який належить до Z, так що $\sigma(s) \in V$, де S означає вибірку в наборі даних Z, а $\sigma(\cdot)$ — функцію вилучення ознак. Крім того, нехай A буде відношенням суміжності, яке з'єднує семпли у V, а chord-distance — функцією відстані, яка зважує ребра в A. Звичайно, є багато варіантів функції відстані для алгоритму OPFYTHON, і відстань хорди є тільки одним із них. Відстань хорди можна обчислити за формулою

$$\text{відстань хорди} = \sqrt{2 - 2\left(\frac{\sum_{i=1}^{n} x_1 y_1}{\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i^2} \right)}. \tag{5}$$

Під час навчання нехай π_S — шлях у G, який закінчується у вузлах $s \in V$, і нехай $\langle \pi_S \cdot (s,t) \rangle$ — зв'язок між шляхом π_S і дугою $(s, t) \in A$. Класифікатор OPF спрямовано на встановлення набору прототипних вузлів $S \subseteq V$ за допомогою функції вартості f, визначеної таким рівнянням:

$$f_{\max}(\pi_S \cdot \langle s,t \rangle) = \max\{f_{\max}(\pi_S), d(st)\}, \tag{6}$$

$$f_{\max}(\langle S \rangle) = \begin{cases} 0, \text{ якщо } s \in S \\ +\infty, \text{ в іншому разі}, \end{cases}$$

де $f_{\max}(\pi_S \cdot \langle s,t \rangle)$ — максимальна відстань між сусідніми вибірками по шляху $\pi_S \cdot \langle s,t \rangle$). Отже, його алгоритм навчання мінімізує $f_{\max}$ для кожної вибірки $t \in z_1$, призначаючи оптимальний шлях $P(t)$ з мінімальною вартістю, визначеною рівністю

$$C(t)=\min_{\forall \pi_t \in (Z_{1,A})}\{f_{\max}(\pi_t)\}. \quad (7)$$

На етапі тестування кожен зразок t буде з'єднано зі зразком $s \in V_1$, ставши частиною вихідного графа. Метою алгоритму є знаходження оптимального шляху $P(t)$, який з'єднує прототип із вузлом t, що досягається оцінюванням шляху за допомогою функції оптимальної вартості:

$$C(t)=\min_{\forall s \in Z_1}\{\max\{C(s),d(s,t)\}\}. \quad (8)$$

Як зазначалося раніше, існує два типи проблем у моделі normal_DIFF_RF-OPFYTHON:

1) на точність моделі значною мірою впливає точність normalJDIFF_RF;

2) модель не в змозі ідентифікувати невідомі атаки.

Для першої проблеми потрібно покращити модуль DIFF_RF за допомогою вкладеної кластерної оптимізації (NCO). NCO містить нестабільність у кожному кластері, а нестабільність, спричинена внутрішньокластерним шумом, не поширюється між кластерами. Порівняно з попереднім удосконаленням модель NCO-normal_DIFF_RF-OPFYTHON має кращий розподіл ваги вхідних змінних і зменшує помилку прогнозування моделі.

Певні коваріаційні структури у вхідних змінних можуть збільшити помилку передбачення моделі. Припустивши, що кореляційна матриця між двома змінними дорівнює C, матрицю C можна діагоналізувати як CW = WA, де:

$$\mathrm{C}=\begin{bmatrix}1 & \rho\\ \rho & 1\end{bmatrix},\quad \Lambda=\begin{bmatrix}1+\rho\\ 1-\rho\end{bmatrix},\quad W=\begin{bmatrix}\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}}\end{bmatrix}. \quad (9)$$

Далі інвертуємо C, щоб дістати C^{-1}

$$C^{-1}=W\Lambda^{-1}W'=\frac{1}{|C|}\begin{bmatrix}1 & -\rho\\ -\rho & 1\end{bmatrix}, \quad (10)$$

де $|C| = \Lambda_{1,1}\Lambda_{2,2} = (1 + \rho)(1 - \rho) = 1 - \rho^2$. З наведеного випливає, що коефіцієнти кореляції, які відхиляються від 0, призводять до того, що $|C|$ наближається до 0, а це зі свого боку спричинює різке зростання значень C^{-1}. Під час навчання така структура сигналу збільшить похибку прогнозування моделі.

Подамо етапи оброблення алгоритму NCO.

Крок 1. Кластеризація всіх вхідних змінних у підмножини висококорельованих змінних ієрархічним методом.

Крок 2. Розрахунок оптимальних розподілів для кожної з цих підмножин висококорельованих змінних окремо.

Крок 3. Розрахунок оптимальних розподілів для кожної зі змінних у всіх підмножинах сильно корельованих змінних.

Крок 4. Розрахунок скалярного добутку внутрішньокластерних розподілів (крок 2) і міжкластерних розподілів (крок 3) для здобуття остаточного оптимального розподілу.

Блок-схему NCO зображено на рис. 1.

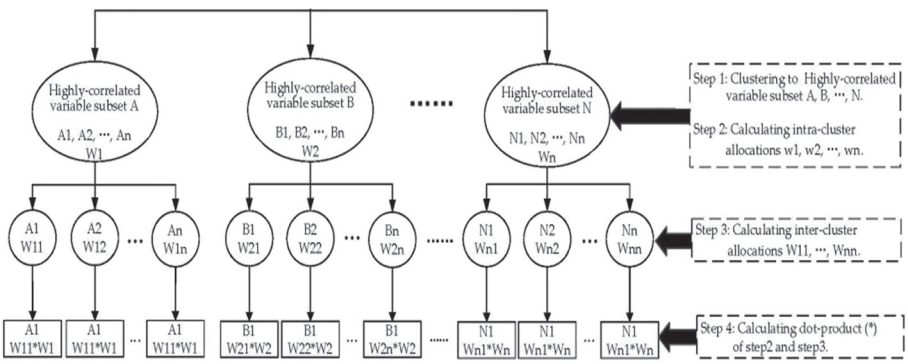

Рис. 1. Блок-схема NCO

Друга проблема в NCO-normal_DIFF_RF-OPFYTHON — вона не може ідентифікувати невідомі атаки. Коли невідомий трафік атаки з'являється у виявленому мережному трафіку, модуль OPFYTHON класифікує невідомий трафік атаки як найбільш схожу атаку.

Отже, далі потрібно вдосконалити структуру моделі на основі наведених раніше дій. Пропонуємо метод NCO-двошаровий DIFF_RF-OPFYTHON. Цей метод додає рівень модуля anomaly_DIFF_RF до моделі NCO-normal_DIFF_RF-OPFYTHON. Під час навчання anomaly_DIFF_RF відповідає лише відомому

трафіку атаки серед усього мережного трафіку. На етапі тестування оцінюється, чи є виявлений мережний трафік відомим трафіком атаки. Модуль anomaly_DIFF_RF може лише визначити, чи є виявлений мережний трафік відомою атакою, тобто він може класифікувати виявлений мережний трафік лише за двома категоріями: належить чи не належить до відомих атак. Отже, модуль anomaly_DIFF_RF класифікує як звичайний мережний трафік, так і відомий трафік атаки як невідомий трафік атаки. Однак перед модулем anomaly_DIFF_RF модуль normal_DIFF_RF класифікує виявлений мережний трафік як такий, що належить або не належить до нормального мережного трафіку. Завдяки встановленню дворівневого модуля DIFF_RF, сформованого з модулів normal_DIFF_RF і модуля anomaly_DIFF_RF, виявлений мережний трафік поділяється на три категорії: нормальний трафік, відомі атаки та невідомі атаки (рис. 2).

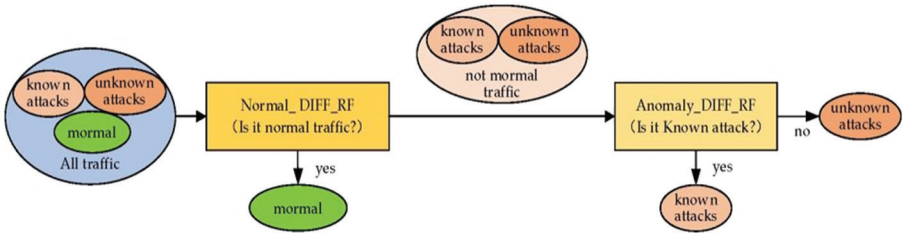

Рис. 2. Потік двошарової моделі DIFF_RF

Після наведеного покрокового вдосконалення остаточна двошарова модель NCO DIFF_RF-OPFYTHON вирішує такі три проблеми:

1) незбалансовані зразки навчання, спричинені надмірним нормальним мережним трафіком;
2) низька точність через модуль DIFF_RF;
3) нездатність ідентифікувати невідомі атаки.

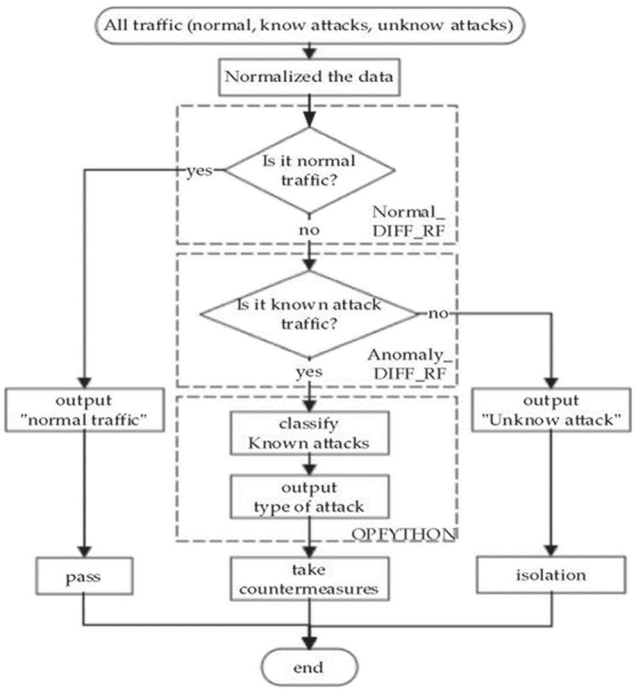

Рис. 3. Процес побудови кінцевої моделі

Повний процес побудови остаточної двошарової моделі NCO DIFF_RF-OPFYTHON унаочнює рис. 3.

Висновки

У статті запропоновано ефективну і надійну модель виявлення вторгнень на основі машинного навчання, а саме NCO-double-layer DIFF_RF-OPFYTHON. Модель було навчено на власному наборі даних і двох загальнодоступних наборах даних, які моделюють сценарії невідомих атак завдяки створенню окремих розділів для наборів даних. Точність і швидкість виявлення невідомих атак методу досягають відповідно 98,75% і 98,4%, що є задовільним експериментальним результатом.

Базовими модулями класифікатора було взято неглибинні алгоритми машинного навчання, що можна покращити, скориставшись ускладненою архітектурою нейронної мережі. У статті всі виявлені невідомі атаки позначені як одна категорія. У наступних дослідженнях планується невідомі атаки кластеризувати та аналізувати для подальшої їх класифікації виявлених невідомих атак.

Список використаної літератури

*1. **Лунь Ю., Сан Л.** Огляд виявлення вторгнень у промислових системах керування // Міжн. J. Distrib. Sens. Netw. 2018.*

*2. **Лю Х., Ленг Б.** Методи машинного та глибокого навчання для систем виявлення вторгнень: огляд. апл. Sci. 2019.*

*3. **Дисбаланс** даних у класифікації: Експериментальна оцінка / Е. Тхабта, С. Хаммуд, Е. Камалов, А. Гонсалвес // Інф. Sci. 2020. 513. С. 429–441.*

*4. **Ян З., Лю ХД, Лі Т.** Систематичний огляд літератури методів і наборів даних для виявлення мережевих вторгнень на основі аномалій // обчис. Secur. 2022. 116.102675.*

*5. **Шах ПАР, Іссак Б.** Порівняння продуктивності систем виявлення вторгнень і застосування машинного навчання до системи Snort // Генератор майбутнього комп. 2018. 80. С. 157–170.*

*6. **Гуріна А., Єлісєєв В.** Аномальний метод виявлення кількох класів мережевих атак // Інформація. 2019. 10.84.*

*7. **Харірі С., Добрий М. Ц., Бруннер Р. Д.** Extended Isolation Forest // IEEE Trans. знати дані інж. 2019. 33. С. 1479–1489.*

*8. **Ніеміес М., Косцей Р., Гдовскі Б.** Багатоваріантний евристичний підхід до виявлення вторгнень у мережевих середовищах // Ентропія, 2021.*

*9. **Бангі Х., Бухнова Б.** Останні досягнення в системі машинного навчання виявлення вторгнень на транспорті: огляд // Procedia Comput. Sci. 2021. 184. С. 877–886.*

*10. **Кілінцер І.Ф., Ертам Е., Сенгур А.** Методи машинного навчання для виявлення вторгнень у кібербезпеку: набори даних і порівняльне дослідження // обчис. Netw. 2021. 188.*

*11. **Механізм** виявлення вторгнень на основі модульної нейронної мережі / Х. Луо, К. Ши, Е. Цяо, Ю. Лі // Матеріали 2-ї міжнар. конф. з машинного навчання, великих даних і бізнес-аналітики (MLBDBI) 2020, Тайюань, Китай, 23-25 жовтня 2020 р. С. 419–423.*

*12. **Прасат М. К., Перумал Б.** Метаевристична класифікація байєсівської мережі для виявлення вторгнень // Міжн. J. Netw. кер. 2019.*

*13. **Евристична** система виявлення та запобігання вторгненням / І. Мухопадхяй, К. С. Гупта, Д. Сен, П. Гупта // Матеріали Міжнар. конф. та семінару з обчислювальної техніки та зв'язку (IEMCON) 2015 р., Ванкувер, Британська Колумбія, Канада, 15-17 жовтня 2015 р. С. 1–7.*

*14. **Azeroual O., Нікіфорова А.** Apache Spark і система виявлення вторгнень на основі MLlib або як технології великих даних можуть захистити дані // Інформація 2022. 13. 58. [CrossRef]*

*15. **Використання** нейронної мережі з довгою короткочасною пам'яттю (LSTM-RNN) для класифікації мережевих атак / П. С. Мухурі, П. Чаттерджі, Х. Юань [та ін.] // Інформація 2020. 11. 243.*

*16. **Сяо Ю, Хіао Х.** Система виявлення вторгнень на основі спрощеної залишкової мережі // Інформація 2019. 10. 356.*

*17. **Покращена** класифікація ELM на основі LDA для алгоритму виявлення вторгнень у програмі IoT / Д. Чжен, З. Гонг, Н. Ван, П. Чен // Sensors. 2020.*

V. Sosnovyi, N. Lashchevska, V. Vlasenko

DEVELOPMENT OF A NEURAL NETWORK STRUCTURE FOR RECOGNITION OF ATTACKS IN COMPUTER SYSTEMS

The integration of industrial control communication networks and the Internet into the Industrial Control System (ICS) increases their vulnerability to cyber-attacks, leading to devastating consequences. Traditional intrusion detection systems (IDS) mostly rely on predefined models and are trained mostly on specific cyber attacks, which means that traditional IDS cannot deal with unknown attacks. In addition, most IDSs do not take into account the imbalance of ICS datasets, and therefore suffer from low accuracy and high false positives when used. In the article, we propose an NCO-double-layer DIFF_RF-OPFYTHON intrusion detection method for ICS, which consists of NCO modules, two-layer DIFF_RF modules, and OPFYTHON modules. Detected traffic will be divided into three categories by the two-layer DIFF_RF module: known attacks, unknown attacks, and normal traffic. Next, the known attacks will be classified by the OPFYTHON module into specific attacks according to the characteristics of the attack traffic. We use the NCO module to improve model inputs and improve model accuracy. The results show that the proposed method outperforms traditional intrusion detection methods such as XGboost and SVM. Detecting unknown attacks is also significant. The accuracy of the data set used in this article reaches 98.13%. The detection rate of unknown and known attacks reaches 98.21% and 95.1%, respectively. In this article, the basic modules of the classifier are shallow machine learning algorithms. This can be improved by using a more powerful neural network architecture. In the article, all detected unknown attacks are marked as one category. In further studies, the unknown attacks can be clustered and analyzed to further classify the detected unknown attacks. It is worth noting that the number of unknown attacks is only a small part, so it is difficult to classify unknown attacks using cluster analysis. Since training a more powerful neural network requires a lot of data, it is important to investigate how to train a new model when the number of samples belonging to that class is limited.

Keywords: machine learning; intrusion detection; network security; countermeasure mode; industrial Control System.