УДК 658.511:004.056
DOI: 10.31673/2409-7292.2018.043543

В.А. Савченко, В.В. Савченко, С.В. Довбешко,
http://orcid.org/0000-0002-3014-131X
М.М. Алексєєв, А.М. Зідан

НЕЙРОМЕРЕЖЕВА ТЕХНОЛОГІЯ ВИЯВЛЕННЯ ІНСАЙДЕРСЬКИХ ЗАГРОЗ НА ОСНОВІ АНАЛІЗУ ЖУРНАЛІВ АКТИВНОСТІ КОРИСТУВАЧІВ

У статті досліджується один з методів виявлення інсайдерських загроз на основі аналізу журналів активності користувачів з використанням штучних нейронних мереж глибокого переконання. Показано, що для ефективного використання мереж глибокого переконання існує необхідність оптимізації структури мережі з пошуком оптимального числа прихованих шарів та кількості вузлів у кожному шарі. Запропоновано алгоритм адаптивної оптимізації мережі з використанням процедури відбору на основі методу дихотомії та правила золотого перетину. Здійснено моделювання, під час якого досягнуто достовірності виявлення загрози на рівні 91 – 92 %.

Ключові слова: інсайдер, інсайдерська загроза, журнал активності користувача, мережа глибокого переконання, адаптивна оптимізація

Вступ. На сьогоднішній день інсайдерські загрози перетворились на серйозну проблему для більшості організацій. Так, за даними опитування керівників корпорацій різного профілю, проведеним дослідницькою компанією Crowd Research Partners у 2018 році, до 90 % організацій у провідних країнах світу відчули свою беззахисність до інсайдерських атак. При цьому основними факторами, які обумовлюють ризик, було відзначено швидке зростання кількості користувачів з особливими правами доступу (37 %), кількості засобів доступу до критично важливих даних (35 %) та складності інформаційних технологій, що застосовуються (35 %). У більшості випадків (53 %) протягом року компанії страждали від атак інсайдерів до 5 разів, при цьому 27 % респондентів зазначили значно більшу кількість вторгнень [1].

І, хоча число шкідливих інсайдерів вважається незначним, істотні ризики для безпеки підприємств залишаються через можливість ненавмисних вторгнень, кількість яких дедалі збільшується. Дослідження кіберзлочинів показують, що нинішні або колишні співробітники організації є другою найбільшою загрозою безпеці, яку випереджують лише хакери, і, що кількість випадків порушення безпеки за останні роки збільшується у геометричній прогресії [2].

Інсайдерські загрози виходять від внутрішніх користувачів, які мають достатні права доступу, можуть ефективно маскуватися та приховувати свою діяльність, ускладнюючи можливість їх виявлення та запобігання загрозам. Ризик збільшується через швидкий розвиток хмарних обчислень, необхідність формування та підтримування великих обсягів даних, застосування технологій централізованої їх обробки. Тому, з метою зменшення втрат, що можуть спричинити інсайдерські загрози, двома найважливішими питаннями в галузі інформаційної безпеки стали виявлення та упередження інсайдерських загроз.

Постановка проблеми. Основним способом виявлення інсайдерської загрози є поведінковий, заснований на тому, що потенційний інсайдер видає свою майбутню атаку через попередні завчасні “індикатори”, які можуть спостерігатися на протязі деякого часу [3]. Базуючись на такому підході розроблено низку програмних комплексів (IBM System G, StaffCop Enterprise та ін.), які використовують різні методи виявлення та аналізу аномалій поведінки користувачів з метою прогнозування ворожих дій. Модулі виявлення інсайдерських загроз є необхідним елементом багатьох UEBA (User and Entity Behavior Analytics), DLP (Data Loss Prevention) та SIEM (Security Information and Event Management) систем. У той же час, виявлення інсайдера, який знає чи здогадується про існування таких засобів, залишається достатньо складною проблемою.

Напрямком вирішення даної проблеми є застосування більш тонких методів розпізнавання/прогнозування, які базуються, зокрема, на сучасних обчислювально-складних підходах в т.ч. на основі штучного інтелекту.

Аналіз джерел. Основою для аналізу поведінки користувача є записи (logs) журналів активності, які дають змогу відслідковувати сценарії діяльності різних користувачів, виявляючи при цьому аномалії. Основними методами аналізу – методи математичної статистики, Баєсівського оцінювання та інтелектуальний аналіз даних (Data Mining) шляхом побудови асоціативних моделей поведінки, пошук аномалій як в окремій обчислювальній системі, так і у мережі в цілому.

Аналіз технологій виявлення аномалій статистичними методами, проведений у [4] дає змогу зробити висновок, що зазначений напрям має ще багато аспектів для удосконалення через значну кількість обмежень, що застосовуються. Так, неврахованими залишаються питання суперечливості записів в журналах поведінки, аналізу багатовимірних даних, адаптації методу до змінного розподілу активності, інтерпретації висновків та ін.

У [5] продемонстровано ймовірнісне розширення методу експоненційно-зваженого ковзного середнього для виявлення аномалій у потоковому середовищі. Цей метод базується на параметричній статистичній моделі, яка може адаптуватися до змінного розподілу потокових даних. У роботі [6] порівнюється метод центрального спостерігача, заснований на експертному оцінюванні з методом відокремленого дерева для задач виявлення аномалій по мережевим записам. У той же час, застосування статистичних методів не дає достатньої достовірності виявлення аномалій поведінки, оскільки базується на апріорному знанні процесу, спостереження за яким ведеться протягом тривалого часу. Крім того, такий підхід є малозастосовним для нових процесів (виконання користувачем нового завдання, яке не пов'язане з попередньою діяльністю).

Більш детальний аналіз поведінки інсайдера можливий на основі застосування нейромережевих технологій, зокрема, на основі моделей глибокого вивчення. Існують різноманітні види моделей глибокого вивчення [7], такі як згорткові нейронні мережі (CNN – Convolutional Neural Networks) [8], автокодувальник (SAE – Stacked Autoencoder) [9] та мережі глибокого переконання (DBN – Deep Belief Networks) [10, 11]. CNN відноситься до класу моделей зі спостерігачем, у той час, як SAE і DBN є моделями без центрального спостерігача.

Враховуючи переваги глибокого вивчення та відносно високу достовірність виявлення модель DBN може бути взята за основу побудови нейромережевої технології виявлення інсайдерських загроз. Ключовою особливістю застовування DBN є її здатність адаптивно змінювати структуру для забезпечення максимально можливого рівня виявлення загрози. Разом з тим, при застосуванні моделі DBN значні труднощі полягають у виборі оптимальної структури мережі.

Метою даної статті є розробка методу оптимізації структури моделі DBN при її застосуванні для вирішення завдання щодо виявлення інсайдерської загрози.

Виклад основного матеріалу. Вихідними даними для виявлення інсайдерів є журнали активності користувачів, у яких відображаються окремі параметри їх поведінки. Модель мережі DBN складається з декількох шарів обмежених машин Больцмана (RBM – Restricted Boltzmann Machine) – породжувальних стохастичних штучних мереж, здатних навчатися розподілу ймовірностей над набором їх входів [12]. Також, у мережі є шар зворотного розповсюдження (рис. 1). Основною концепцією моделі мережі DBN є те, що кожен рівень мережі здійснює навчання особливостям поведінки за відсутності центрального спостерігача, лише на основі попередніх спостережень. Таким способом проводиться налаштування кожного шару, при цьому результати роботи у окремому шарі сприймаються як входи наступного шару. Нарешті, вся мережа може бути налаштована тонко, використовуючи контрольоване випробовування, що робить вхід і вихід моделі максимально адекватними.

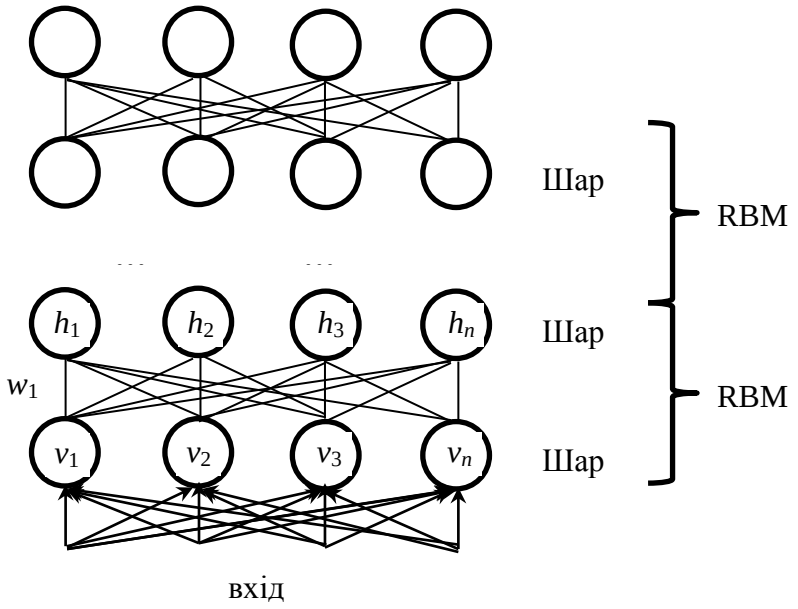

Рис.1. DBN модель.

Суть глибокого навчання за допомогою моделі DBN полягає у повному вивченні особливостей шляхом перебору параметрів структури мережі, при цьому ядром моделі є структури RBM.

Структура RBM складається з двох частин: видимого шару (v) та прихованого шару (h). Вузли одного шару між собою не з'єднані, тоді як вузли різних шарів з'єднані один з одним. При цьому розподіл ймовірностей $P(v,h)$ відповідає розподілу Больцмана. Тому, видимий шар і прихований шар структури RBM можуть бути представлені один одним. Встановивши початкові значення прихованого шару, ймовірність видимого шару $P(v|h)$ може бути визначена за рівняннями

$$P(v|h)=\prod_i P(v_i|h), \tag{1}$$

$$P(v_i=1|h)=\frac{1}{1+\exp\left(-\sum_j w_{ij}h_j-c_i\right)}. \tag{2}$$

У свою чергу, задавши видимий шар, ймовірність прихованого шару $P(h|v)$ визначається за рівняннями

$$P(h|v)=\prod_j P(h_j|v), \tag{3}$$

$$P(h_j=1|v)=\frac{1}{1+\exp\left(-\sum_i w_{ij}v_i-b_j\right)}, \tag{4}$$

де b – зсув видимого шару, c – зсув прихованого шару, а функція активації – сигма-функція.

Процес навчання моделі DBN полягає в постійному оновленні вагового параметра $\theta = \{w_1, w_2, ..., b_1, b_2, ..., c_1, c_2, ...\}$ з метою максимізації розподілу ймовірностей $P(v,h)$ вектора видимого шару v і прихованого шару h [13]. Це дасть можливість обчислення похідної $\frac{\partial \log P(v,h)}{\partial \theta}$, яка визначатиме кут нахилу розподілу ймовірностей $P(v,h)$. Формула оновлення вагових параметрів

$$\theta_{\tau+1} = \theta_\tau + \eta \frac{\partial \log P(v,h)}{\partial \theta}\bigg|_{\theta_\tau}, \quad (5)$$

де τ та η, відповідно, визначають число ітерацій та швидкість навчання під час RBM навчання.

Згідно з правилами контрастної дивергенції [14] кожен з компонентів може бути визначено за формулами

$$w_{\tau+1} = w_\tau + \eta\left(\langle v_i h_j \rangle_{ориг} - \langle v_i h_j \rangle_{обч}\right), \quad (6)$$

$$b_{\tau+1} = b_\tau + \eta\left(\langle v_i \rangle_{ориг} - \langle v_i \rangle_{обч}\right), \quad (7)$$

$$c_{\tau+1} = c_\tau + \eta\left(\langle h_j \rangle_{ориг} - \langle h_j \rangle_{обч}\right). \quad (8)$$

У виразах (6) – (8) індекс **"***ориг***"** позначає вихідні дані спостережень, а індекс **"***обч***"** – дані, які розраховуються за допомогою мережевої моделі.

Процес виявлення інсайдерської загрози на основі адаптивної оптимізації DBN.

Існує 4 основних етапи процесу виявлення інсайдерської загрози на основі адаптивної оптимізації DBN, який включає: формування журналів інсайдерської поведінки, попередню обробку записів внутрішньої поведінки, глибоке вивчення особливостей інсайдерської поведінки та класифікацію моделей поведінки [15].

Перший етап – збір початкової інформації про поведінку користувачів. Оскільки журнали активності існують у вигляді окремих записів, то вони повинні бути попередньо оброблені і перетворені в стандартну чисельну форму. Оброблені дані використовуються як вхідні дані моделі для глибокого вивчення DBN. Модель глибокого вивчення повністю аналізує функції вхідних даних, змінюючи значення ваги θ мережі, в результаті чого виводяться значення функції, які описують поведінку користувача на четвертому етапі. На цьому етапі завдання полягає у перетворенні початкових записів до стандартної чисельної форми, що є необхідним для подальшого глибокого вивчення на наступному етапі. Ця процедура полягає у обробці записів та їх нормалізації.

1) *Обробка записів:* записи поведінки користувачів здійснюються у різній формі відповідно до їх специфіки. Однак, загроза інсайдерської атаки часто буває спричинена комбінацією різних типів поведінки. Тому журнали поведінки інсайдерів повинні бути формалізовані відповідно до конкретного типу поведінки. Записи поведінки користувачів фіксуються, як правило, у форматі 4-елементних кортежів, які включають: час появи активності; суб'єкта поведінки; джерело активності та особливості поведінки. Перші 3 елементи є загальними елементами для всіх типів записів, які легко обробляти, тоді як особливості поведінки (четвертий елемент) будуть завжди різні, відповідно до функціональної специфіки конкретного користувача. Тому вхідні дані журналу більш доцільно обробляти шляхом застосування методу типізації поведінки, який можна подати

наступним чином: припустимо, що в організації зафіксовавно n типів записів b_1, b_2, ... b_n користувачів. Для будь-якого типу поведінки b_i, елемент буде дорівнювати 0, якщо поведінка не відповідає певному встановленому типу. Іншими словами, для кожного типу поведінки користувача існує визначене значення коду запису. Таким чином, якщо в організації зафіксовано n типів записів користувачів, то записи поведінки інсайдерів, після їх обробки, будуть представлені допоміжними n+3 записами, і кожен новий запис може бути віднесений до вже існуючого n+3 кортежу.

2) *Нормалізація записів:* з метою вивчення особливостей поведінки користувачів на основі моделі DBN дані мають бути нормалізовані. Для цього найбільш доцільним є застосування процедури нормалізації на основі батч-нормалізації [16] – методу прискорення глибокого навчання, який вирішує проблему спотворення сигналу під час його проходження через внутрішні шари мережі. Батч-нормалізація передбачає приведення початкових даних до форми, яка має нульове математичне очікування та одиничну дисперсію. Нормалізація виконується перед входом до кожного шару шляхом обчислення математичного очікування та дисперсії для визначеного батча (пакету) $B = x_1,...,x_m$ наступним чином $m_B = \frac{1}{m}\sum_{i=1}^{m} x_i$, $\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - m_B)^2$. За допомогою цих статистичних характеристик функція активації перетворюється таким чином, щоб вона мала нульове математичне очікування та одиничну дисперсію на усьому батчі: $\hat{x}_i = \frac{x_i - m_B}{\sqrt{\sigma_B^2 + \varepsilon}}$ де ε – параметр, що захищає від ділення на 0 у випадку, коли середньоквадратичне відхилення наближається до 0.

3) *Адаптивна оптимізація моделі* DBN *під час вивчення особливостей інсайдерської поведінки*: нормалізовані дані формуються після попередньої обробки записів поведінки користувачів, відповідно до яких структура глибокої мережі адаптуєтья для вивчення їх поведінки. У результаті цього етапу мережа має бути здатною розрізняти нормальну поведінку користувачів та ненормальну поведінку користувачів-інсайдерів. Навчальна модель DBN пристосовується для випробовувань та здійснює попереднє навчання на структурі мережі з декількома прихованими шарами RBM, використовуючи функцію активації w та правила контрастної дивергенції, а потім здійснює зворотне поширення, таким чином налаштовуючи параметри структури мережі для формування оптимізованої моделі DBN.

Для оптимізації структури можуть використовуватись методи дихотомії (поділ інтервалу на два рівних піддіапазони і поступове зменшення кількості прихованих вузлів), або правило “золотого перетину” (поділ інтервалу на два нерівні підінтервали, де співвідношення більшої частини до всієї частини дорівнює співвідношенню меншої частини до більшої частини, тобто 0.618) [17].

4) *Класифікація інсайдерської загрози на основі адаптивної оптимізації* DBN.

Оптимізована мережева модель формується після глибокого вивчення поведінки користувачів, що дозволяє виявити та класифікувати поведінку інсайдерів. Існує два види класифікації для глибокого вивчення – бікласифікація та мультикласифікація. Як правило, у бікласифікації використовується сігма-функція, в той час як в мультикласифікації використовується функція *softmax*, яка для k класів обчислюється за формулою [15]

$$f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}}, \tag{9}$$

де, f_j – ймовірність того, що результатом класифікації є *j*.

Сума ймовірностей, яка задовольняє всім результатам класифікації, дорівнює 1, тобто:

$$\sum_k f_k(z) = \frac{\sum_k e^{z_k}}{\sum_k e^{z_k}} = 1. \quad (10)$$

У цьому дослідженні поведінка користувачів класифікується за 6 категоріями. Одна – це нормальна поведінка користувача, інші – це ненормальна поведінка за 5 сценаріями, що свідчить про наявність інсайдерської атаки.

Алгоритм адаптивної оптимізації DBN при виявленні інсайдерської загрози

Виявлення інсайдерської загрози на основі адаптивної оптимізації DBN за журналами поведінки дозволяє адаптивно обирати глобальну оптимальну структуру мережі.

У якості змінних параметрів задамо:

input – кількість вузлів у вхідному шарі;

output – кількість вузлів вихідного шару;

results – набір оптимальних результатів;

layers – структура мережі DBN;

c – кількість прихованих шарів.

Процес оптимізації структури мережі може бути визначено за наступним алгоритмом:

1. Початок: Введення початкових даних: ***input*** = кількість початкових змінних; ***output*** = кількість класифікаторів; ***results*** = []; ***layers*** = []; ***c*** = 1.

2. Присвоюємо: максимальне значення кількості вузлів на ***c***-му прихованому шарі: ***i*** = ***input.***

3. Перевірка умови: якщо ***i*** > ***output***, то: обчислюється кількість вихідних вузлів на ***c***-му прихованому шарі $output_j$, а саме – мінімальна кількість вузлів на ***c***-му прихованому шарі та перехід до кроку 4; **у протилежному випадку** – досягнуто глобальної оптимальної структури ***layers***, алгоритм завершується.

4. Присвоюємо: кількість вибіркових вузлів на ***c***-му прихованому шарі є ***j***, а початкове значення ***j = i.***

5. Перевірка умови: якщо $j < output_j$, то: додаємо ***j*** до структури мережі для створення нової тимчасової структури мережі $layer_j$, до якої застосовується глибоке вивчення за допомогою моделі DBN; додаємо результат $result_j$ до набору результатів на ***c***-му прихованому шарі ***result***, $j = j - 1$, та переходимо до кроку 5 і продовжуємо цикл; **у протилежному випадку –** вибираємо оптимальний результат $resOptimal_j$ на ***c***-му прихованому шарі ***result***; кількість вузлів ***resOptimal*** може розглядатися як оптимальна кількість вузлів; перехід до кроку 6.

6. Додаємо $resOptimal_j$ до оптимального набору результату ***results*** на ***c***-му прихованому шарі.

7. Додаємо оптимальну кількість вузлів ***resOptimal*** до структури мережі на ***c***-му прихованому шарі для формування нової структури мережі ***layers*.**

8. Обчислюємо максимальну кількість вузлів наступного прихованого шару ***i*** = ***resOptimal.***

9. Додаємо ще один прихований шар ***c*** = ***c*** + 1, і переходимо до кроку 3.

10. Обираємо оптимальний результат з набору результатів ***results***, що включає всі приховані шари; відповідний індекс є ***max***, а шари ***layers*** **[1, ..., *max*]** є глобальною оптимізованою структурою мережі.

11. Кінець алгоритму.

Моделювання та аналіз результатів. Для перевірки адекватності моделі та оцінки одержаних результатів було змодельовано низку спостережень за поведінкою користувачів. Загальне число спостережень склало 6000 записів роботи 30 користувачів, з яких 15 розглядалися як навчальна вибірка, решта 15 – у якості контрольної вибірки. Серед загального числа з 6000 записів 200 записів були навмисно сформовані так, що визначали підозрілу поведінку, яка свідчила про наявність інсайдерської загрози. Після інтеграції та нормалізації даних оптимізаційна модель DBN була навчена на навчальній вибірці та перевірена на тестовому наборі.

Моделювання здійснено з використанням середовища розробки Keras, яка має достатньо потужні можливості щодо обробки результатів та дозволяє, за необхідності, вносити правки у модель [18].

Вибірка записів поведінки користувачів імітує 1 варіант підозрілої поведінки та 5 сценаріїв нормальної поведінки: робота з файлами (створення, видалення, зміна імені чи типу файла та ін.), пересилання електронної пошти, застосування периферійних пристроїв (накопичувачів, принтерів), доступ до мережевих ресурсів та інші варіанти поведінки. Окрім варіанту поведінки записи реєструють також дані про користувача – час, робоче місце та підрозділ працівника.

До записів, які відображають підозрілу поведінку, було включено операції, які відбувалися у неробочий час, активність працівника перед залишенням робочого місця, надсилання великої кількості листів, спроби входу до чужих облікових записів з надсиланням файлів на власні електронні скриньки, наднормова кількість копіювання файлів. Такі записи супроводжувалися інформацією про час, статус користувача, ім'я хосту та варіанту поведінки користувача.

У якості ключової гіпотези обрано правильне визначення підозрілої поведінки, як поведінки інсайдера. Результати класифікації, як правило, оцінюються за наступними п'ятьма показниками:

похибка другого роду $FNR = FN / N_{загр}$;

похибка першого роду $FPR = FN / N_{норм}$;

вірно прийнята гіпотеза $TNR = TN / N_{загр}$;

вірно-відхилена гіпотеза $TPR = TP / N_{норм}$;

та рівень точності $AR = \dfrac{TP + TN}{TP + FN + TN + FP}$.

При цьому: *TP* – кількість правильно визначених випадків регламентованої поведінки; *TN* – кількість правильно визначених потенційних загроз; *FN* – кількість пропущених потенційних загроз, які було визначено як легальна поведінка; *FP* – кількість випадків легальної поведінки, які було помилково класифіковано як загрозу; $N_{норм}$ – загальна кількість випадків нормальної поведінки користувачів, $N_{загр}$ – загальна кількість випадків загрозливої поведінки користувачів.

Через те, що *FNR* та *TNR* а також *FPR* та *TPR* зв'язані між собою, є можливість досліджувати лише один з показників, у якості якого доцільно обрати кількість вірно прийнятих гіпотез *TNR* та вірно відхилених гіпотез *TPR*.

Під час оцінювання кількості навчальних зразків, які обробляються одночасно за одну ітерацію оптимізації (рис. 2) було визначено, що коли кількість зразків становить 60, *TNR* досягає максимального значення 91.3 %, що свідчить про достатньо ефективну адаптивну оптимізацію моделі DBN.

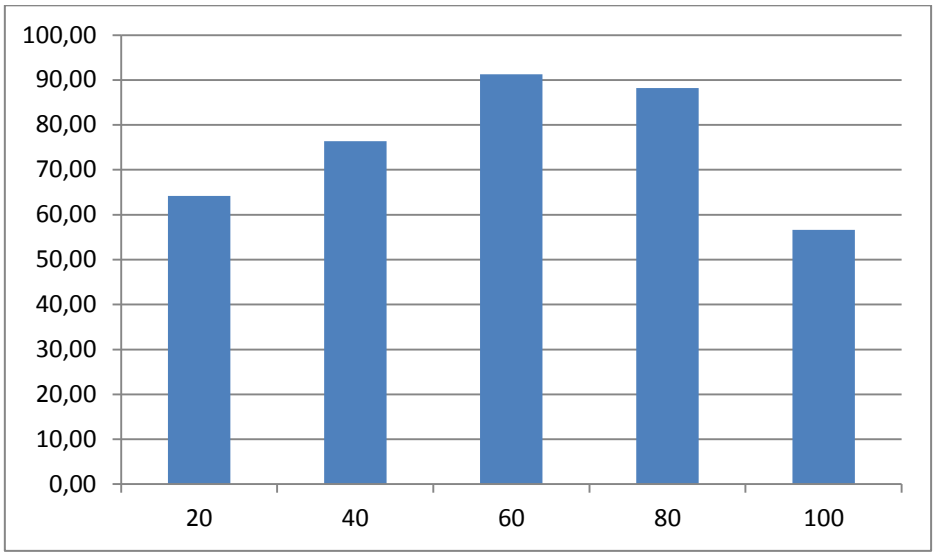

Рис.2. Оцінка кількості навчальних зразків, які обробляються одночасно за одну ітерацію оптимізації за TNR.

При цьому ж значенні *TNR* оцінка доцільної кількості ітерацій RBM показує, що модель досягає максимуму продуктивності при мінімумі ітерацій RBM. Тобто, навіть при лише одній ітерації можна досягти максимального значення *TNR* (рис. 3).

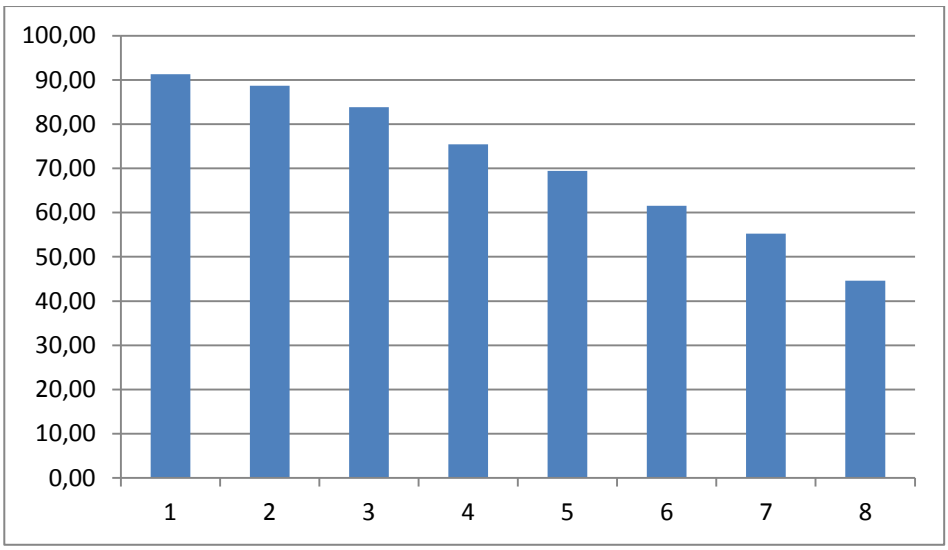

Рис.3. Визначення кількості ітерацій RBM

При визначенні кількості прихованих шарів їх оптимальне число становить від 2 до 6. При цьому забезпечуються максимальні значення *TNR*. У подальшому, збільшення кількості шарів не призводить до суттєвого покращення цього показника, у той же час робота моделі суттєво ускладнюється.

Порівняльна оцінка ефективності оптимізації за методом дихотомії та правилом "золотого перетину" дає змогу зробити висновок про те, що обидва цих методи володіють достатньою продуктивністю і дають майже однакові результати (91 – 92 % правильно визначених випадків інсайдерської загрози). Разом з тим, застосування адаптивного методу

оптимізації золотого перетину краще, ніж використання методу дихотомії, за швидкістю роботи моделі (6,4 с проти 9,6 с відповідно).

Висновки та напрями подальших досліджень. Аналіз записів поведінки користувачів будь-якої системи свідчить, що переважна більшість випадків поведінки є нормальним варіантом діяльності працівників організації. У той же час, збитки, спричинені організації у результаті інсайдерської атаки, можуть бути достатньо суттєвими. Тому виявлення інсайдерів є актуальним та важливим завданням.

Використання моделей глибокого вивчення дає можливість виявляти особливості записів активності користувачів від початку до кінця, відшукуючи приховані зв'язки та виявляючи паралелі. Застосування алгоритму адаптивної оптимізації DBN дає змогу досягти ефективності розпізнавання ситуації на рівні 91 – 92 %, що свідчить про значні перспективні можливості методу.

У рамках проведеного експерименту було визначено, що оптимальним розміром пакету навчальних зразків, які обробляються одночасно за одну ітерацію, є 60 і модель може досягти задовільної продуктивності коли RBM повторюється лише один раз. При цьому, за показником TNR, можна досягти максимального значення точності визначення на рівні 91.3 %.

Напрямом подальших досліджень моделей DBN може бути широке коло питань ефективного навчання моделі в умовах сильної зашумленості початкових даних, наявності значного числа пропущених даних, роботи методу в умовах навмисного спотворення варіантів поведінки інсайдерів.

Список використаних джерел

1. Insider Threat. 2018 Report. Crowd Research Partners. Режим доступу https://www.ca.com/content/dam/ca/us/files/ebook/insider-threat-report.pdf
2. Greitzer FL, Moore AP, Cappelli DM, Andrews DH, Carroll LA, Hull TD. Combating the insider cyber threat. IEEE Secur Priv 2008; 6: 61-64.
3. Мартьянов Е.А. Возможность выявления инсайдера статистическими методами // Системы и средства автоматики. –2017, т. 27, № 2. – С. 41– 47.
4. Chandola, V.; Banerjee, A.; and Kumar, V. 2012. Anomaly detection for discrete sequences: A survey. IEEE TKDE 24(5):823–839.
5. Carter, K. M., and Streilein, W. W. 2012. Probabilistic reasoning for streaming anomaly detection. InProc. SSP, 377–380.
6. Gavai, G.; Sricharan, K.; Gunning, D.; Hanley, J.; Singhal, M.; and Rolleston, R. 2015. Supervised and unsupervised methods to detect insider threat from enterprise social and online activity data.Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications6(4):47–63.
7. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science 2006; 313: 504.
8. Zeiler MD, Fergus R. Visualizing and understanding convolutional networks. In: European Conference on Computer Vision; 6–12 September 2014; Zurich, Switzerland. pp. 818-833.
9. Cao LL, Huang WB, Sun FC. Building feature space of extreme learning machine with sparse denoising stackedautoencoder. Neurocomputing 2016; 174: 60-71.
10. Hinton GE, Osindero S, Teh YW. A fast learning algorithm for deep belief nets. Neural Comput 2006; 18: 1527-1554.
11. Bengio Y. Learning Deep Architectures for AI. Foundations and Trends in Machine Learning. Delft, the Netherlands: Now Publishers, 2009.
12. Hinton GE. A practical guide to training restricted Boltzmann machines. In: Montavon G, editor. Neural Networks: Tricks of the Trade 2012. 2nd ed. Berlin, Germany: Springer. pp. 599-619.
13. Salakhutdinov R, Hiton G. An efficient learning procedure for deep boltzman machines. Neural Comput 2012; 24: 1967-2006.
14. Hiton GE. Training products of experts by minimizing contrastive divergence. Neural Comput 2002; 14: 1771-1800.
15. Zhang J., Chen Y., Ju A. Insider threat detection of adaptive optimization DBN for behavior logs. Turkish Journal of Electrical Engineering & Computer Sciences. (2018) 26: 792 – 802.
16. Ioffe S., Szegedy C. Batch Normalization: Accelerating Deep Network Training b y Reducing Internal Covariate Shift. Mode of Access: https://arxiv.org/pdf/1502.03167.pdf

17. Stakhov A.P. The generalized principle of the golden section and its applications in mathematics, science, and engineering. Chaos Soliton Fract 2005; 26: 1157-1182.
18. Keras: The Python Deep Learning library. Mode of Access https://keras.io/