ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ СИСТЕМИ

УДК 004.645.34
DOI 10.17721/ISTS.2023.1.61-69

Тетяна Бабенко, orcid.org/0000-0003-1184-9483,
babenko.tetiana.v@gm
Андрій Бігдан, orcid.org/0000-0002-2940-6085,
abigdan@gmail.com
Лариса Мирутенко, orcid.org/0000-0003-1686-261X,
myrutenko.lara@gmail.com
Київський національний університет імені Тараса Шевченка, Київ, Україна

ІНТЕЛЕКТУАЛЬНА МОДЕЛЬ КЛАСИФІКАЦІЇ МЕРЕЖНИХ ПОДІЙ ІЗ КІБЕРБЕЗПЕКИ

Через збільшену складність сучасних комп'ютерних атак, виникає потреба у фахівцях із безпеки не тільки для виявлення шкідливої активності, але і для визначення відповідних кроків, які проходитиме зловмисник у ході виконання атаки. Незважаючи на те, що виявлення експлойтів і вразливостей зростає з кожним днем, розроблення методів захисту просувається помітно повільніше за розроблення методів нападу. Саме тому це все ще залишається відкритою дослідницькою проблемою. У цій статті представляємо дослідження у галузі ідентифікації мережних атак із використанням нейронних мереж, зокрема багатошарового персептрона Румельхарта, для виявлення та прогнозування майбутніх подій мережної безпеки на основі попередніх спостережень. Для забезпечення якості процесу навчання й отримання бажаного узагальнення моделі використано 4 млн записів, накопичених протягом 7 днів Канадським інститутом кібербезпеки. Наш результат демонструє, що моделі нейронних мереж, що базуються на багатошаровому персептроні, можуть використовуватися після уточнення для виявлення та прогнозування подій мережної безпеки.

Ключові слова: *безпека інформаційних систем; нейронна мережа; мережна безпека; прогнозування.*

1. ВСТУП

Щороку до загальної кількості програмного коду додається 111 трильйонів рядків, причому кожен рядок потенційно може бути новою вразливістю і, як наслідок, може бути реалізована атака нульового дня. Зазначимо, що технології, які використовуються зловмисниками для атак на комп'ютерні системи та мережі, стають усе складнішими та досконалішими [1]. Вивчають багато шляхів розв'язання цієї проблеми, зокрема і технології, що дозволяють створювати захищене програмне забезпечення [2]. Із цією метою для автоматизації процесів контролю подій безпеки в інформаційних системах традиційно використовують системи виявлення вторгнень (IDS/IPS – Intrusion Detection/Protection System), основним завданням яких є автоматизація процесу виявлення атак або неправомірного застосування [3–6]. Ці системи залежно від технології, що використовується для виявлення атак, прийнято розділяти на дві основні групи: системи виявлення зловмисної поведінки користувачів; системи виявлення аномальної поведінки комп'ютерної системи. У першому випадку порівнюється шаблон атаки з потоком подій, у другому – порівнюється шаблон нормальної поведінки системи з потоком подій. Прийнято вважати, що завдання виявлення вторгнень у мережі TCP/IP зводиться до розпізнавання задач [7–9]:

- структурні ознаки (сигнатури) відомих видів атак;
- інваріантні ознаки структури правильних обчислювальних процесів;
- кореляційні ознаки нормального функціонування розподілених обчислювальних систем.

У разі проблеми з розпізнаванням мережних аномалій виникають деякі труднощі, які в основ-

ному пов'язані зі зростаючим попитом на виявлення раніше невідомих атак і деструктивних впливів, що у свою чергу потребує:

• побудови еталонних множин нормального (семантично правильного) профілю поведінки системи в умовах невизначеності впливів зовнішнього середовища;

• визначення необхідних і достатніх інформативних ознак;

• побудови правил визначення аномалій.

Як правило, виявлення мережних аномалій здійснюється за схемою, в якій виокремлюють такі функціональні блоки [10]:

• аналіз інформації, що міститься в заголовках IP-дейтаграм;

• побудова прогнозування;

• пошук та оцінювання аномалій;

• реакція на аномалію;

• наповнення та/або редагування базових правил IDS/IPS.

Зібрану статистичну інформацію використовують для побудови математичної моделі прогнозування трафіка на основі циклічного аналізу часових рядів. Ця модель дозволяє прогнозувати завантаження мережі на основі пошуку частот у мережному трафіку.

У всіх випадках IDS не зможе виявити вторгнення, тому що не зможе відрізнити його від фонового білого шуму, який існує в будь-якій системі через слабкість інструменту аналізу пакетів або відсутність сигнатури відповідної атаки тощо.

Отже, більшість типових способів виявлення атак і протидії їм мають низьку точність і швидкість і не дозволяють ефективно протидіяти як відомим атакам, так і атакам нульового дня. Тому розробляється велика кількість різних технологій захисту комп'ютерних систем та мереж, заснованих на технологіях перевірки даних із використанням штучного інтелекту із застосуванням нейронних мереж [11–15]. Це пов'язано зі здатністю нейромережної структури розв'язувати завдання, що важко формалізуються, а також з її здатністю до навчання, самоорганізації та узагальнення. Такий підхід дозволяє отримувати моделі, здатні швидко адаптуватися до навколишнього середовища та прогнозувати розвиток процесу на основі якості узагальнення [16, 17].

Штучний інтелект – термін у широкому сенсі – спирається за допомогою комп'ютерів на імітації можливостей людини: почуття, розуміння, реагування.

Машинне навчання – область штучного інтелекту у розділі комп'ютерних наук, де часто використовують статистичні методи, щоб дати комп'ютерам можливість "навчатись" (наприклад, поступово підвищувати продуктивність у конкретній задачі) [18].

Розглядаючи науку про дані, зазначимо, що для виконання (використання) алгоритмів машинного навчання необхідне визначення наборів даних, вибір відповідних змінних та метрик, а також виконання різних інформаційно-інженерних завдань: пошук прихованих залежностей, збір даних, навчання, інтеграція, візуалізація, визначення продуктивності алгоритмів тощо.

Кожна модель навчання має ґрунтуватися на певному алгоритмі. Зокрема, до них належать класифікація, кластеризація, асоціативні правила, поглиблене навчання, регресія, зіставлення із зразком. Вибір алгоритму залежить від кінцевої мети, яку потрібно досягти. Модель ідентифікації подій кібербезпеки, що розглядається в цьому дослідженні, заснована на навчанні з учителем та на алгоритмах класифікації нейронних мереж.

2. МАТЕРІАЛ І МЕТОДИ

У ході розв'язання завдання класифікації подій, що відбуваються у процесі мережної взаємодії, виникають проблеми, в основному пов'язані з необхідністю обліку невідомих атак і деструктивних впливів, що у свою чергу вимагає: системи в умовах невизначеності впливів зовнішнього середовища, визначення необхідних і достатніх інформативних ознак та побудови правил виявлення аномалій [19, 20].

Аналогічне і менш складне завдання виконано для атак із застосуванням (ін'єкції) SQL (Structured Query Language) [21]. SQL-ін'єкція – це атака, що реалізується шляхом модифікації запитів до бази даних за рахунок експлуатації вразливостей, що містяться у вебдодатках. Успішна реалізація атаки дає зловмиснику можливість отримати конфіденційну інформацію, змінити або знищити її.

Для синтезу й аналізу моделі ідентифікації атак SQL-ін'єкцій були підготовлені набори даних попереднього навчання, контролю та тестування. Навчальний набір даних містив параметри навчання; вибір параметрів був евристично заснований на аналізі основних ознак атаки, які можуть містити URL-адресу.

Штучні нейронні мережі – це математичні моделі та їхні програмні чи апаратні реалізації. Цей термін з'являється під час вивчення процесів, що відбуваються в мозку, та за спроби змоделювати ці процеси. Основними принципами є інтерпретація сенсорних даних за допомогою свого роду машинного сприйняття, маркування або угруповання

даних, що надходять. Образи, що розпізнаються, є числовими і містяться у векторах, в які транслюються будь-які інші дані [22].

Кожен вузол має один або кілька входів і один вихід. Нейрон має два режими роботи: режим навчання та режим використання або тестування. У режимі навчання нейрон навчається реагувати на певний вхідний шаблон. У робочому режимі нейрон реагує на вхідний шаблон і пов'язує вихід. Якщо нейрон отримує на вхід нетиповий набір параметрів, він сам визначає, активувати себе чи ні.

Кожен вхідний сигнал має відповідну вагу, яка розраховується на основі вхідних даних. Якщо це число перевищить поріг, нейрон спрацює.

Активація будь-якого нейрона регулюється його активаційною функцією.

Нейронні моделі працюють виключно з числовими даними, представленими в деякому числовому діапазоні, тому на першому етапі дослідження розроблено класифікатор URL. Це програмний модуль, який перетворює URL-адресу на двійковий формат і встановлює логічний ідентифікатор "true (1)", якщо адреса відноситься до атаки, і "false (0)" – в іншому випадку. Таким чином, для кожної з URL-адрес було згенеровано вхідний вектор, тобто вихідний вектор може бути представлений у форматі

$$X^{\mathrm{T}} = [x_1 x_2 \cdots x_n],$$

де n – кількість параметрів запиту, що використовується у шаблонах SQL.

На основі цього підходу нейромережна модель матиме на вході n нейронів. Кожен вектор характеризується параметром: доброякісний (0) – не відноситься до нападу, та ін'єкційний (1) – відноситься до нападу. Достатньо мати на виході лише один нейрон, що розділяє вхідні вектори на два класи 0 та 1.

У результаті відносна похибка синтезованої моделі на контрольному та дослідному зразках не перевищує 5 %. Таким чином, можемо застосувати такий підхід для ширшого спектра подій безпеки й атак.

Метою цього дослідження є вивчення можливостей використання штучних нейронних мереж, зокрема персептрона Румельхарта (окремий випадок персептрона Розенблатта) для виявлення мережних атак і прогнозування подій, пов'язаних із мережною безпекою [23, 24]. Для забезпечення якості процесу навчання й отримання бажаного узагальнення властивостей моделі необхідно мати значну кількість прикладів реалізації відповідних атак. В експериментальних цілях зібрано 4 млн записів, які протягом 7 днів накопичив Канадський інститут кібербезпеки. Мережна інфраструктура зловмисника складалася з 50 машин. Організація-жертва містила 5 відділів, і кожен з них використовував 420 користувацьких вузлів і 30 серверів. Набір даних містить інформацію про перехоплений мережний трафік і системні журнали кожної машини організації-жертви. Схему оброблення даних показано на рис. 1. Для аналізу набору даних використовували профільні поняття: B-профіль і M-профіль.

B-профіль (Benign – доброякісний) – інкапсуляція поведінки користувача з використанням різних методів машинного навчання та статистичного аналізу (таких як, K-Means, Random Forest, SVM і J48).

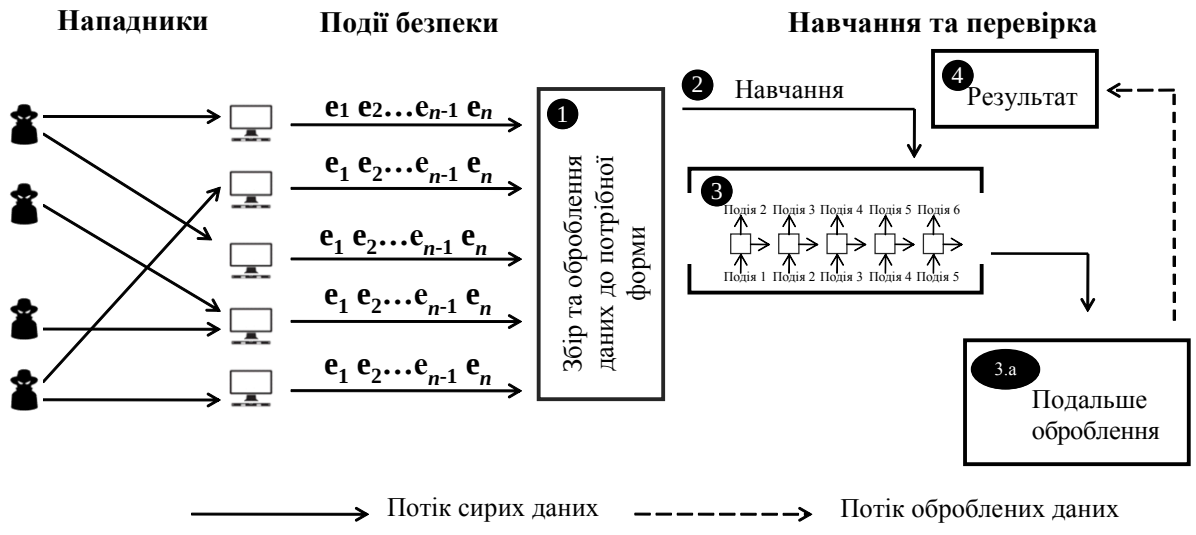

Рис. 1. Схема оброблення даних

Інкапсульовані функції – це розподіл пакетів протоколу за розміром, кількістю пакетів у потоці, певною структурою корисного навантаження, розміром корисного навантаження та запитом тимчасового поділу для протоколу.

Модель складається з таких кроків:

1. Збір даних про мережну активність із системних журналів і журналів подій.

2. Оброблення попередніх даних і зведення їх до потрібного вигляду.

3. Навчання та тестування нейронної мережі.

4. Аналіз результатів.

У моделюванні використовували такі протоколи: HTTPS, HTTP, SMTP, POP3, IMAP, SSH та FTP. За результатами частотного аналізу даних зроблено висновок, що основна маса трафіка представлена HTTP- та HTTPS-пакетами. M-профіль (Malignant – зловмисний) – спроба однозначно описати сценарій атаки. У найпростішому випадку люди можуть інтерпретувати ці профілі, а потім їх виконувати. В ідеалі для інтерпретації та виконання цих сценаріїв використовуватимуться автономні агенти разом із компіляторами. Було розглянуто різні сценарії реалізації атак:

• мережне проникнення: у цьому сценарії зловмисний файл надіслано жертві електронною поштою. Після успішної експлуатації вразливості на комп'ютері жертви запускався бекдор, після чого комп'ютер жертви використовувався для сканування внутрішньої мережі та пошуку нових поштових скриньок;

• відмова в обслуговуванні HTTP: у цьому сценарії як основні інструменти використовувалися Slowloris і LOIC (Low Orbit Ion Cannon – додаток для стрес-тестування й атаки типу "відмова в обслуговуванні" (DoS або DDoS) із відкритим вихідним кодом, написаний на C#), які, як відомо, роблять вебсервери повністю недоступними через одну атакуючу систему. Атака за допомогою Slowloris починається з повного з'єднання TCP із віддаленим сервером. Інструмент підтримує з'єднання відкритим, відправляючи дійсні неповні HTTP-запити на сервер через певні проміжки часу, щоб уникнути закриття сокетів. Оскільки будь-який вебсервер може обслуговувати обмежену кількість можливих підключень, вичерпання всіх сокетів буде лише питанням часу, і в результаті жодних інших підключень не буде прийнято;

• атака на вебпрограми: у цьому сценарії використовувався вразливий вебдодаток (DVWA – Damn Vulnerable Web App), який спеціально розроблено, щоб допомогти фахівцям із безпеки перевірити свої навички в аналізі безпеки. Першим кроком є сканування вебсайту за допомогою сканера вразливостей вебзастосунків, а потім виконання різних типів вебатак на вебсайт, включаючи SQL-ін'єкції, ін'єкції системних команд і можливість завантаження незахищених файлів.

До способів виявлення вразливостей програмного забезпечення для SQL-ін'єкцій належать: функціональне тестування (чорна/біла скринька); фазинг (техніка автоматизованого тестування програмного забезпечення, яка полягає в тому, що на вхід програми подають недійсні, невідповідні або випадково згенеровані дані); статичний, динамічний, ручний аналізи вихідного коду. Також слід зазначити, що паралельно з пошуком вразливостей у програмованих програмах зазвичай використовується широкий спектр WAF (Web Application Firewall – брандмауер вебдодатків). Як правило, вони засновані на двох моделях безпеки: на основі підпису та на основі правил. Кожна із цих моделей має свої переваги й недоліки, але їхнім загальним недоліком є неможливість виявлення загроз "нульового дня", і зауважимо, що використання WAF може лише частково перекрити вектор атак [25–28].

Таким чином, більшість типових підходів до забезпечення безпеки від атак SQL-ін'єкцій не дозволяють отримати достатній рівень безпеки через низьку точність ідентифікації та швидкість роботи. Тому нині з'явилася велика кількість різних технологій захисту комп'ютерних систем і мереж, які ґрунтуються на технологіях інтелектуального аналізу даних, зокрема і на використанні нейронних мереж, що дозволяє ефективно протидіяти вже відомим атакам та атакам "нульового дня". Розроблено такі атаки:

• атака грубою силою: в основному спрямована на підбір комбінації імені користувача та пароля для отримання доступу до облікового запису користувача. Для цієї атаки існує безліч інструментів, таких як модулі Hydra, Medusa, Ncrack, Metasploit та Nmap NSE. Крім того, є деякі інструменти, такі як hashcat і hashpump для злому хеш-паролів. Але одним з найповніших багатопотокових інструментів є Patator, який написаний на Python і є більш гнучким, ніж інші. Він також може зберігати кожну відповідь в окремому файлі журналу для подальшого перегляду та оброблення. Для переліку паролів ми використали словник із 90 млн слів;

• атаки на останнє оновлення: атаки, засновані на деяких відомих вразливостях, які можуть бути реалізовані протягом певного періоду часу (це екстраординарні вразливості, які іноді зачіпають мільйони серверів або жертв, і зазвичай

потрібно кілька місяців, щоб виправити весь вразливий програмний код), одна з найвідоміших в останні роки – Heartbleed.

Детальну інформацію про виявлені атаки та засоби, використані для їхньої реалізації, представлено в табл. 1.

***Таблиця* 1**

Атаки на цільові системи

Тип атаки	Застосовані інструменти
Атака грубої сили (Brute-force)	FTP – Patator SSH – Patator
Атака на відмову в обслуговуванні (DoS)	Hulk, GoldenEye, Slowloris, SlowHTTPtest, Heartleech
Вебатака	Damn Vulnerable Web App (DVWA), власний Selenium Framework (XSS та Brute force)
Інфільтраційна атака	Перший рівень: завантаження Dropbox на комп'ютері з Windows. Другий рівень: Nmap і PortScan
Атака ботнету	Ares (розробка Python): віддалена оболонка (remote shell), завантаження/вивантаження файлів, захоплення знімків екрана та запис клавіш, що натискаються
DDoS разом із PortScan	Low Orbit Ion Canon (LOIC) для UDP, TCP або HTTP запитів

3. РЕЗУЛЬТАТИ

Існує два підходи до аналізу мережних атак: один ґрунтується на аналізі мережної активності, інший – на аналізі вмісту пакетів. У цьому дослідженні ми зосередилися на підході, заснованому на аналізі мережної активності. Аналіз активності мережі проводився за допомогою спеціалізованого програмного забезпечення CICFlowMeter, яке генерує двонаправлені потоки, де відправлення першого пакета визначає шлях до джерела призначення та назад до вихідної системи, і дозволяє отримати більше 80 статистичних атрибутів мережного трафіка. Для цілей моделювання визначено 67 параметрів мережного трафіка у кожному потоці.

Під час підготовки даних було додано новий атрибут Label, який ідентифікує потік як конкретну атаку чи нормальну роботу інформаційних служб. Таким чином, усі дані були позначені відповідно до таких значень: Benign, FTP-BruteForce, SSH-Bruteforce, DoS-GoldenEye, DoS-Slowloris, DoS-SlowHTTPTest, DoS-Hulk, DDoS attacks-LOIC-HTTP, DDoS-LOIC-UDP, DDOS-HOIC, Brute Force–Web, Brute Force – XSS, Infiltration, Bot.

Синтез нейромережної моделі (рис. 2) виконано на основі багатошарового персептрона Румельхарта. Багатошаровий персептрон Румельхарта є окремим випадком персептрона Розенблатта, в якому вагові коефіцієнти нейрона коригують за допомогою алгоритму зворотного розповсюдження похибки. Особливістю нейронної мережі є більше одного шару (зазвичай двох чи трьох шарів) [22]. Отже нейронна мережа у вигляді персептрона Розенблатта ділить вхідні вектори на два класи 0 і 1. Навчальна послідовність формується з двох масивів: вхідного масиву X і масиву цілей Y, який привласнює кожному зі вхідних векторів одного з двох класів.

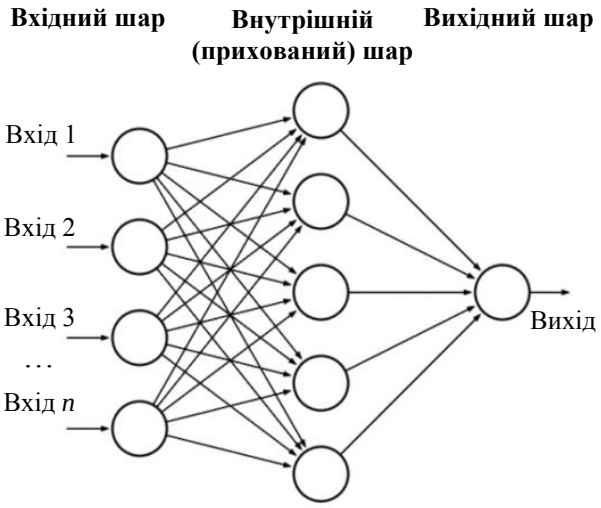

Рис. 2. Модель нейронної мережі

У дослідженні використовувалися три шари: вхідний, вихідний і один прихований. Операції нейронних мереж зворотного поширення похибки можна розділити на два етапи: пряме і зворотне поширення. На етапі прямого поширення вхідний шаблон застосовується до вхідного шару, і його ефект поширюється шар за шаром по мережі до отримання результату. Фактичне вихідне значення мережі порівнюється з очікуваним виходом, і для кожного з вихідних вузлів обчислюється сигнал помилки. Оскільки всі приховані вузли певною мірою сприяли виявленню помилок у вихідному шарі, помилки вихідних сигналів поширюються назад від вихідного шару до кожного вузла прихованого (внутрішнього) шару, що впливає на вихідний шар. Потім цей процес повторюється шар за

шаром, поки кожен вузол мережі не отримає повідомлення про помилку, що описує його відносний внесок у загальну помилку.

Після визначення сигналу помилки для кожного вузла дані про помилки будуть використовуватися для оновлення значень ваги кожного з'єднання, доки мережа не ввійде у стан, що дозволяє закодувати всі схеми навчання. Алгоритм зворотного розповсюдження помилки шукає мінімальне значення функції помилки у просторі вагових значень, використовуючи техніку, яка називається дельта-правилом або градієнтним спуском. Шкали, які мінімізують функцію помилок, вважаються розв'язанням проблеми навчання [11, 12].

У той час, як певний шаблон передається вхідному шару у процесі навчання, зважена сума входу j-го вузла у прихованому шарі обчислюється за формулою

$$Net_j = \sum w_{ij} x_j + \theta_j. \quad (1)$$

Рівняння (1) використовують для розрахунку загального входу до нейрона θ_j, який є зваженим вузлом зміщення, що завжди має вихідне значення 1. Вузол зміщення вважається "псевдовходом" для кожного нейрона у прихованому та вихідному шарі і використовується для розв'язання проблем у ситуаціях, коли значення вхідного шаблона дорівнює нулю. Якщо будь-який вхідний шаблон містить нульові значення, нейронна мережа може бути навчена без вузла зміщення.

Щоб вирішити, чи активувати нейрон, значення потенціалу Net_j дії передається у відповідну функцію активації. Результуюче значення функції активації визначає вихід нейрона та стає вхідним значенням для нейронів у наступних шарах, які з ним пов'язані.

Оскільки однією з вимог до алгоритму зворотного поширення помилки є те, що функція активації має бути диференційована, типовою функцією є сигмоїдальне рівняння:

$$O_j = x_k = \frac{1}{1+ e^{-Net_j}}. \quad (2)$$

Слід зазначити, що можуть використовуватися інші типи функцій, наприклад, гіперболічні. Рівняння (1) та (2) застосовують для визначення вихідного значення вузла k у вихідному шарі.

Синтез моделі ґрунтувався на створенні власного програмного забезпечення, що реалізує навчання та тестування моделі. Основні математичні алгоритми, використані для нормалізації даних і навчання моделі, виконувались із використанням Weka API (інтерфейс прикладного програмування). Weka – це програмне забезпечення з відкритим вихідним кодом, випущене під GNU General Public License, що містить набір алгоритмів машинного навчання для задач інтелектуального аналізу даних. Він містить інструменти для підготовки даних, класифікації, регресії, кластеризації, правил вилучення асоціацій і візуалізації. Weka містить API, який написаний на Java та реалізує існуючі алгоритми навчання з мінімальними налаштуваннями. Остаточний модуль навчання та перевірки написано мовою програмування Java.

Для ідентифікації події із заданою точністю потрібно, щоб відносна помилка не перевищувала 4 %. Через великий обсяг навчальних даних він був розбитий на кілька блоків за типом атаки і прийнято рішення синтезувати окрему нейромережну модель для ідентифікації кожного типу атаки.

Для класифікації атак із використанням синтезованої моделі створено спеціальну процедуру з таким алгоритмом.

1. Завантаження тренувальних даних.

2. Визначення номінальних значень із числових. Нейронна мережа представляє числові значення в певному відношенні, ваги коригуються від значення величини. Під час процесу нормалізації даних залежності між певними атрибутами можуть бути неправильно інтерпретовані і заплутати процес навчання. Для критичних числових атрибутів, таких як порт (1-65565), необхідно змінити представлення. Для розв'язання цієї проблеми використовують форматування числових значень до номінальних.

3. Нормалізація даних. Нормалізація даних у нейронних мережах – це процес оптимізації значень набору даних для числових типів з великого діапазону в діапазон значень від 0 до 1 із збереженням пропорційності. Нормовані значення значно збільшують швидкість навчання моделі та не порушують правильність її навчання.

4. Класифікатор узгоджує вихідне значення нейронної мережі з типом виявленої атаки.

5. Валідатори описують вихідні класи мережі, які використовуються для відображення значення класифікації користувача.

6. Відбувається класифікація шаблонів на основі отриманої відповіді від нейромережі.

7. Визначення помилки мережі. Якщо дані, що передаються у класифікатор, містять шаблони збігів, можна обчислити відносну помилку мережі.

Ми суворо вимагаємо, щоб дані навчання, перевірки та тестування були отримані з різних машин, щоб вони могли ідентифікувати можливості в кінцевих точках, які не є частиною даних навчання. Зокрема, ми навчаємо модель для 100 епох,

перевіряємо продуктивність моделі після кожної епохи та вибираємо модель, що забезпечує найкращу продуктивність для даних перевірки. Детальна інформація про оцінку адекватності отриманої моделі для представлення тестової підмножини даних наведена в табл. 2. Ця вибірка відображає різні типи атак та їхню відносну похибку ідентифікації у разі застосування синтезованої моделі.

***Таблиця* 2**

Результати випробувань моделі

Назва атаки	Відносна помилка, %
FTP- Bruteforce	0,15
SSH- Bruteforce	0,10
Dos attack GoldenEye	97,50
Dos attack Slowloris	98,96
Dos attack LOIC-UDP	100,00
Dos attack HOIC	0,00
Dos attack HULK	0,00
HTTP Benign	80,40
Infiltration	100,00
Botnet	0,004
Inf-Bot Benign	0,00

4. ВИСНОВКИ

Аналіз отриманих результатів показує, що відносна помилка ідентифікації у процесі представлення тестових зразків до синтезованої моделі суттєво відрізняється для різних типів мережних атак. Як видно з табл. 2, модель не може ідентифікувати такі типи DoS-атак, як GoldenEye, Slowloris, LOIC-UDP та HTTP Benign. Проте загалом подальші дослідження можливості використання цього типу нейронних мереж на розв'язання завдань ідентифікації мережних атак дуже перспективні. Під час досягнення прийнятних результатів точність моделі ідентифікації дозволить не тільки виявляти мережні атаки, а й виконувати прогноз подій мережної безпеки на основі ретроспективних даних, накопичених в інформаційній системі за період і переданих нейромережною моделлю для навчання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Chen, P., Desmet, L, & Huygens, C. (2014). A study on advanced persistent threats in *IFIP International Conference on Communications and Multimedia Security*. Aveiro, Portugal, pp. 63–72.

[2] Stringhini, G., & Thonnard, O. (2015). That ain't you: Blocking spearphishing through behavioural modelling in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*.

[3] SPECTRUM.IEEE.ORG INTERNATIONAL. (2019, Feb.). *01 The WhiteHat Hacking Machine*, pp. 30–35.

[4] Denning, Dorothy E. (1986, May), An Intrusion Detection Model in *Proceedings of the Seventh IEEE Symposium on Security and Privacy*, pp. 119–131.

[5] Scarfone, K, & Mell, P. (2007, Feb.). Guide to Intrusion Detection and Prevention Systems (IDPS). *NIST Special Publication on Computer security*, pp. 58–69.

[6] Daş, R., Karabade, A. & Tuna, G. (2015, 16–19 May). Common Network Attack Types and Defense Mechanisms in *Signal Processing and Communications Applications Conference (SIU)*, pp. 2658–2666.

[7] Bellovin, S. M., AT & T. Lab Res., USA. (2004, 6–10 Dec.). A look back at security problems in the *TCP/IP protocol suite 20th Annual Computer Security Applications Conference*. USA, pp. 268–286.

[8] Borkar, A., Donode, A., & Kumari, A. (2017, 23–24 Nov.). A survey on Intrusion Detection System (IDS) and Internal Intrusion Detection and protection system (IIDPS in *International Conference on Inventive Computing and Informatics (ICICI)*. Coimbatore, India, pp. 878–880.

[9] Azhagiri, M., Rajesh, A., & Karthik, S. (2015). Intrusion detection and prevention system: technologies and challenges. *International Journal of Applied Engineering Research*. India, ISSN 0973-4562, vol. 10, no. 87, pp. 1–11.

[10] Daş, R., & Baykara, M. (2015, October). A Survey on Potential Applications of Honeypot Technology in Intrusion Detection Systems, *in International Journal of Computer Networks and Applications (IJCNA)*, vol. 2, no. 5, pp. 203–208.

[11] Linh Van Ma Van Quan Nguyen; Jin-young Kim; Kwangki Kim; & Jinsul Kim (2018). Applications of Anomaly Detection Using Deep Learning on Time Series Data in *16th Intl Conf on Dependable, Autonomic and Secure Computing*, pp. 393–396.

[12] Usage of Machine Learning for Intrusion Detection in a Network. *International Journal of Computer Networks and Applications (IJCNA)*, vol. 3, Issue 6, (2016, November–December) Prachi Department of CSE & IT, The NorthCap University. India, pp. 139–145.

[13] Shen, Y., Mariconti, E., Vervier, P. A., & Stringhini, G. (2018). "Tiresias", in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security – CCS '18*, pp. 592–605.

[14] Lianbing, Z. (2016). Study on Applying the Neural Network in Computer Network Security Assessment in *Eighth International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, pp. 639–642.

[15] Li J., & Dong, C. (2010). Research on Network Security Situation Prediction-Oriented Adaptive Learning Neuron in *Second International Conference on Networks Security, Wireless Communications and Trusted Computing*, vol. 2, pp. 483–485.

[16] Shin, E. C. R., Song, D., & Moazzezi, R. (2015, 12–14 August). Recognizing Functions in Binaries with Neural Networks in *USENIX Security Symposium Washington*, pp. 611–626.

[17] Kuznetsov, A. A., Smirnov, A. A., Danilenko, D. A., & Berezovsky, A. (2015).The statistical analysis of network traffic for the intrusion detection and prevention systems. *Telecommunications and Radio Engineering*, vol. 74, Issue 1, pp. 61–78.

[18] Menshawy, A. (2018). *Deep Learning By Example: A Hands-on Guide to Implementing Advanced Machine Learning Algorithms and Neural Networks*. Pact Publishing Ltd. 442 p.

[19] Naumenko, N. I., Stasev, Yu. V., & Kuznetsov, A. A. (2007, May), Methods of synthesis of signals with prescribed properties. *Cybernetics and Systems Analysis*, vol. 43, Issue 3, pp. 321–326.

[20] Duman, S., Kalkan-Cakmakci, K, Egele, M., William K. Robertson, K., & Kirda, E. (2016, 10–14 June). EmailProfiler: Spearphishing Filtering with Header and Stylometric Features of Emails in *IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, USA, pp. 121–126.

[21] Hubskyi, O., Babenko, T., Myrutenko, L., & Oksiiuk, O. (2021). Detection of SQL injection attack using neural networks. *Advances in Intelligent Systems and Computing*, 1265 AISC, pp. 277–286.

[22] Haykin, S. (2010). *Neural Networks and Learning Machines: International Edition. 3rd edn.* Pearson Education. 936 p.

[23] Stringhini, G., Holz, T., Stone-Gross, B., Kruegel, C., & Vigna, G. (2011, 8–12 August). *BotMagnifier: Locating Spambots on the Internet" in Proceedings of the 2011 USENIX Security Symposium San Francisco*. CA, pp. 427–443.

[24] Toliupa, S., Babenko, T., & Trush, A. (2017). The building of a security strategy based on the model of game management in *4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T*. Kharkiv, Ukraine, pp. 103–108.

[25] Multiple Buffer Format String Vulnerabilities in SQL Server. http://www.microsoft.com/technet/security/bulletin/MS01-060.asp, last accessed 2020/03/11.

[26] Taking the monkey work out of pentesting, http://pentestmonkey.net/

[27] The Web Application Security Consortium / SQL Injection. http://projects.webappsec.org/SQL-Injection

[28] Microsoft SQL Server extended stored procedure vulnerability (technical explanation and exploit code) SecuriTeam, https://securiteam.com/windowsntfocus/6n0010u0kw/

REFERENCES

[1] Chen, P., Desmet, L, & Huygens, C. (2014). A study on advanced persistent threats in *IFIP International Conference on Communications and Multimedia Security*. Aveiro, Portugal, pp. 63–72.

[2] Stringhini, G., & Thonnard, O. (2015). That ain't you: Blocking spearphishing through behavioural modelling in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*.

[3] SPECTRUM.IEEE.ORG INTERNATIONAL. (2019, Feb.). *01 The WhiteHat Hacking Machine*, pp. 30–35.

[4] Denning, Dorothy E. (1986, May), An Intrusion Detection Model in *Proceedings of the Seventh IEEE Symposium on Security and Privacy*, pp. 119–131.

[5] Scarfone, K, & Mell, P. (2007, Feb.). Guide to Intrusion Detection and Prevention Systems (IDPS). *NIST Special Publication on Computer security*, pp. 58–69.

[6] Daş, R., Karabade, A. & Tuna, G. (2015, 16–19 May). Common Network Attack Types and Defense Mechanisms in *Signal Processing and Communications Applications Conference (SIU)*, pp. 2658–2666.

[7] Bellovin, S. M., AT & T. Lab Res., USA. (2004, 6–10 Dec.). A look back at security problems in the *TCP/IP protocol suite 20th Annual Computer Security Applications Conference.* USA, pp. 268–286.

[8] Borkar, A., Donode, A., & Kumari, A. (2017, 23–24 Nov.). A survey on Intrusion Detection System (IDS) and Internal Intrusion Detection and protection system (IIDPS in *International Conference on Inventive Computing and Informatics (ICICI)*. Coimbatore, India, pp. 878–880.

[9] Azhagiri, M., Rajesh, A., & Karthik, S. (2015). Intrusion detection and prevention system: technologies and challenges. *International Journal of Applied Engineering Research*. India, ISSN 0973-4562, vol. 10, no. 87, pp. 1–11.

[10] Daş, R., & Baykara, M. (2015, October). A Survey on Potential Applications of Honeypot Technology in Intrusion Detection Systems, *in International Journal of Computer Networks and Applications (IJCNA)*, vol. 2, no. 5, pp. 203–208.

[11] Linh Van Ma Van Quan Nguyen; Jin-young Kim; Kwangki Kim; & Jinsul Kim (2018). Applications of Anomaly Detection Using Deep Learning on Time Series Data in *16th Intl Conf on Dependable, Autonomic and Secure Computing*, pp. 393–396.

[12] Usage of Machine Learning for Intrusion Detection in a Network. *International Journal of Computer Networks and Applications (IJCNA)*, vol. 3, Issue 6, (2016, November–December) Prachi Department of CSE & IT, The NorthCap University. India, pp. 139–145.

[13] Shen, Y., Mariconti, E., Vervier, P. A., & Stringhini, G. (2018). "Tiresias", in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security – CCS '18*, pp. 592–605.

[14] Lianbing, Z. (2016). Study on Applying the Neural Network in Computer Network Security Assessment in *Eighth International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, pp. 639–642.

[15] Li J., & Dong, C. (2010). Research on Network Security Situation Prediction-Oriented Adaptive Learning Neuron in *Second International Conference on Networks Security, Wireless Communications and Trusted Computing*, vol. 2, pp. 483–485.

[16] Shin, E. C. R., Song, D., & Moazzezi, R. (2015, 12–14 August). Recognizing Functions in Binaries with Neural Networks in *USENIX Security Symposium Washington*, pp. 611–626.

[17] Kuznetsov, A. A., Smirnov, A. A., Danilenko, D. A., & Berezovsky, A. (2015).The statistical analysis of network traffic for the intrusion detection and prevention systems. *Telecommunications and Radio Engineering*, vol. 74, Issue 1, pp. 61–78.

[18] Menshawy, A. (2018). *Deep Learning By Example: A Hands-on Guide to Implementing Advanced Machine Learning Algorithms and Neural Networks*. Pact Publishing Ltd. 442 p.

[19] Naumenko, N. I., Stasev, Yu. V., & Kuznetsov, A. A. (2007, May), Methods of synthesis of signals with prescribed properties. *Cybernetics and Systems Analysis*, vol. 43, Issue 3, pp. 321–326.

[20] Duman, S., Kalkan-Cakmakci, K, Egele, M., William K. Robertson, K., & Kirda, E. (2016, 10–14 June). EmailProfiler: Spearphishing Filtering with Header and Stylometric Features of Emails in *IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, USA, pp. 121–126.

[21] Hubskyi, O., Babenko, T., Myrutenko, L., & Oksiiuk, O. (2021). Detection of SQL injection attack using neural networks. *Advances in Intelligent Systems and Computing*, 1265 AISC, pp. 277–286.

[22] Haykin, S. (2010). *Neural Networks and Learning Machines: International Edition. 3rd edn.* Pearson Education. 936 p.

[23] Stringhini, G., Holz, T., Stone-Gross, B., Kruegel, C., & Vigna, G. (2011, 8–12 August). *BotMagnifier: Locating Spambots on the Internet" in Proceedings of the 2011 USENIX Security Symposium San Francisco*. CA, pp. 427–443.

[24] Toliupa, S., Babenko, T., & Trush, A. (2017). The building of a security strategy based on the model of game management in *4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T*. Kharkiv, Ukraine, pp. 103–108.

[25] Multiple Buffer Format String Vulnerabilities in SQL Server. http://www.microsoft.com/technet/security/bulletin/MS01-060.asp, last accessed 2020/03/11.

[26] Taking the monkey work out of pentesting, http://pentestmonkey.net/

[27] The Web Application Security Consortium / SQL Injection. http://projects.webappsec.org/SQL-Injection

[28] Microsoft SQL Server extended stored procedure vulnerability (technical explanation and exploit code) SecuriTeam, https://securiteam.com/windowsntfocus/6n0010u0kw/

Стаття надійшла до редколегії 20.02.2023

Intelligent model of classification of network cyber security events

Due to the increased complexity of modern computer attacks, there is a need for security professionals not only to detect harmful activity but also to determine the appropriate steps that an attacker will go through when performing an attack. Even though the detection of exploits and vulnerabilities is growing every day, the development of protection methods is progressing much more slowly than attack methods. Therefore, this remains an open research problem. In this article, we present our research in network attack identification using neural networks, in particular Rumelhart's multilayer perceptron, to identify and predict future network security events based on previous observations. To ensure the quality of the training process and obtain the desired generalization of the model, 4 million records accumulated over 7 days by the Canadian Cybersecurity Institute were used. Our result shows that neural network models based on a multilayer perceptron can be used after refinement to detect and predict network security events.

Keywords: *security of information systems; neural network; network security; prognostication.*

Тетяна Бабенко,
д-р техн. наук, проф.
Професор кафедри кібербезпеки та захисту інформації Київського національного університету імені Тараса Шевченка. Київ, Україна.

Tetyana Babenko,
Dr. Sci. (Engin.), Prof.
Professor of the Department of Cyber Security and Information Protection, Taras Shevchenko Kyiv National University. Kyiv, Ukraine.

Андрій Бігдан,
Асистент кафедри кібербезпеки та захисту інформації Київського національного університету імені Тараса Шевченка. Київ, Україна.

Andrii Bigdan,
Assistant of the Department of Cyber Security and Information Protection, Taras Shevchenko Kyiv National University. Kyiv, Ukraine.

Лариса Мирутенко,
канд. техн. наук, доц.
Доцент кафедри кібербезпеки та захисту інформації, Київського національного університету імені Тараса Шевченка. Київ, Україна.

Larisa Myrutenko,
PhD (Engin.), Associate Prof.
Associate Professor of the Department of Cyber Security and Information Protection, Taras Shevchenko Kyiv National University. Kyiv, Ukraine.