ЗМІСТ

СПИСОК УМОВНИХ СКОРОЧЕНЬ

АРТ – нейронна мережа адаптивної резонансної теорії

БШП – багатошаровий перспетрон

БД – база даних

ВШ – вхідний шар

ДАП – двонаправлена асоціативна пам'ять

ДШП – двохшаровий перспетрон

ЗІ – захист інформації

ЗЗІ – засоби захисту інформації

ІБ – інформаційна безпека

ІС – інформаційна система

ЕС – експертна система

ЕП – експертні правила

КС – комп'ютерна система

НМ – нейрона мережа

НММ – нейромережева модель

НМС – нейромережеве середовище

НК – несподівана кібератака

НСД – несанкціонований доступ

ОС – операційна система

ПБ – параметри безпеки

ПК – пружна карта (пружинна карта)

ПЗ – програмне забезпечення

ПС – програмне середовище

РБФ – нейронна мережа з радіальними базисними функціями

СВА – система виявлення атак

СВВ – система виявлення вразливостей

СЗІ – система захисту інформації

СЛД – синхронізоване лінійне дерево

СМ – семантична мережа

СНМ – семантична нейронна мережа

СШН – схований шар нейронів

РІС – ресурс інформаційної системи

ША – шаблон атаки

ШВ – шар виходу

ШД – шар додавання

ШНМ – швидка нейронна мережа

ШО – шар образів

ВСТУП

За останні декілька років в різних галузях науки, техніки, економіки та медицини збільшився інтерес до використання штучних НМ. Багато в чому популярність НМ пояснюється можливістю їх ефективного застосування в випадках, коли класичні "аналітичні" методи не спрацьовують. В теоретичних роботах [2, 4, 5, 7, 9, 10, 12, 13] присвячених НМ наголошується, що їх використання доцільне в задачах класифікації та кластеризації образів, апроксимації функцій, прогнозування, оптимізації, управління, створення інформаційно-обчислювальних систем з асоціативною пам'яттю. Відзначимо, що частково або в комплексі, вирішувати перераховані задачі доводитися при розробці методів і засобів ЗІ. Вперше відповідні пропозиції були доведені до широкого загалу в роботі [1], де показана методика та описані експерименти по діагностиці аномальної мережевої активності за допомогою БШП. На сьогодні відомі [3, 14, 15, 16] спроби використання НМ в компонентах СВА та СВВ. В більшості випадків йдеться про використання в таких системах управляючого елементу на базі того ж БШП. З його допомогою розв'язується задача розпізнавання реалізації атаки та/або потенційної вразливості КС, тобто задача розпізнавання образів. Вказані засоби захисту набули певного поширення, проте всі вони володіють рядом істотних недоліків, які обмежують їх практичну цінність [3, 4]. До вказаних недоліків відносяться високий рівень помилкових тривог, складність підбору оптимальних граничних параметрів, складність введення в систему нового суб'єкта/об'єкту контролю, недостатня адаптація до багатьох особливостей сучасного стану галузі інформаційних технологій. Крім того, в [16, 17] представлені методики використання інших типів НМ в задачах ЗІ. Однак в цих роботах практично відсутнє обґрунтування вибору типу мережі, а також розрахунок параметрів її архітектури, включаючи один із основних моментів - визначення та первинну обробку вхідних параметрів мережі.

Якщо ж навіть таке обґрунтування частково і є, то в ньому недостатньо висвітлені питання обмежень обчислювальних можливостей того чи іншого типу

мереж. В той же час ефективність розв’язання практичної задачі в значній мірі залежить і від типу і від параметрів мережі, а сучасна теорія дозволяє знайти достатньо точні відповіді на подібні питання [2, 4, 5, 6, 7, 8, 9, 10]. Це свідчить про актуальність та важливість проблеми визначення ефективності розробки управляючих елементів на базі різних типів НМ при створенні методів та засобів ЗІ. Власне вирішенню цієї проблеми і присвячена дана монографія. В ній на базі широкого аналізу теорії нейромережевих технологій проведена оцінка ефективності використання більшості класичних та деяких перспективних типів НМ при розв'язанні актуальних задач ЗІ. Для кожного типа оцінка підкріплена методикою адаптації мережі до умов прикладної задачі. Крім того, розроблена методика та наведені приклади застосування НМ для класифікації комп’ютерної мережі, розпізнавання макровірусів в системах активного захисту.

РОЗДІЛ 1
АНАЛІЗ ПРОБЛЕМИ РОЗПІЗНАВАННЯ КІБЕРАТАК

1.1. Проблематика розпізнавання мережевих кібератак

Аналіз теоретичних робіт [21, 25] вказує на те, що в загальному випадку для ІС принципова доцільність застосування нейромережевих методів розпізнавання визначається:

1. Відсутністю або низькою ефективністю явних алгоритмічних методів розпізнавання.

2. Можливістю навчання НМ.

3. Можливістю забезпечення технічних аспектів використання НМ.

Як правило, труднощі пов’язані з навчанням НМ [21, 25, 26]. Для ефективного навчання необхідно виконати нас тупні умови:

1. В піддослідному процесі визначити кількісні достатньо інформативні показники, котрі будуть використані в якості вхідних та вихідних параметрів НМ.

2. Сформувати навчальну вибірку НМ.

3. При використанні визначеного обсягу обчислювальних ресурсів та допустимій помилці навчання термін навчання НМ не повинен перевищувати заданий інтервал часу.

Деталізуємо перераховані умови з точки зору застосування НМ в основних методах виявлення атак – зловживань та аномалій [18, 23, 26]. Зазначимо, що при використанні в СВА методу зловживань, НМ застосовується для розпізнавання сигнатур атак (СА), тобто величин контрольованих параметрів, котрі вказують на реалізацію атаки. При використанні методу аномалій НМ застосовується для виявлення відхилень контрольованих параметрів від шаблонів нормальної поведінки (ШНП) ресурсів ІС.

Потенційно в якості вхідних параметрів НМ можуть бути використані всі зареєстровані на експлуатації функціональні параметри ІС, котрі можна застосувати для виявлення певного типу атаки. Перед поданням в НМ вказані функціональні параметри потрібно відповідним чином закодувати та нормалізувати [21, 25]. В найпростішому випадку вихід НМ повинен вказувати на

наявність або відсутність атаки. В складніших випадках вихід НМ повинен ще й вказувати на тип атаки. Відповідно, для формування навчальних прикладів необхідні статистичні дані про величини функціональних параметрів у випадку реалізації атаки та під час нормального функціонування ІС. Також в навчальній вибірці слід якомога повніше відобразити всі можливі образи атак та профілі нормальної поведінки об'єк тів ІС. Тому множину навчальних прикладів можна визначити за допомогою наступного виразу:

$$V_A \bigcup V_N \in O, \tag{1.1}$$

де О – множина навчальних прикладів, V_A – множина образ ів можливих атак, V_N – множина профілів нормальної поведінки.

Крім того, при формуванні навчальної вибірки слід враховувати наступні вимоги:

1. Мережа навчена на прикладах нормального/аномального функціонування однієї ІС може видавати неправильний результат для інших ІС.

2. Як правило, додавання та вилучення із складу ІС навіть окремих об'єктів призводить до зміни характеристик параметрів захищеності, а відповідно, і до необхідності внесення змін до навчальної вибірки НМ.

3. Мінімальна кількість навчальних прикладів повинна мінімум в 10-20 разів перевищувати кількість вхідних параметрів [21, 25]:

$$P_{min} \geq (10..20)N_X, \tag{1.2}$$

де $P_{\min}$ – мінімальна кількість навчальних прикладів, N_x – кількість вхідних параметрів НМ.

4. Кількість навчальних прикладів не може бути нескінченною.

5. Приклади навчальної вибірки повинні пропорційно представляти всі класи, які повинна розпізнати НМ.

Тому формування достатнього обсягу навчальних прикладів може викликати проблеми, пов'язані як з масштабуванням навчальних даних (вимоги 1, 2), так власне із збором достатньої кількості статистичної інформації (вимоги 3, 4, 5). Зазначимо, що проблема масштабування характерна для методу виявлення

аномалій та пов’язана з формуванням ШНП. В той же час для СВА, котрі базуються на пошуку сигнатур атак, характерною є проблема збору статистичної інформації.

Максимально допустиму тривалість розробки НМ можна оцінити так

$$T_f \leq T_a, \tag{1.3}$$

де T_f – максимально допустима тривалість розробки НМ, T_a – термін, протягом якого ризик від реалізації атаки не перевищує встановлену межу.

Для визначення величини T_a можливо застосувати методи експертного оцінювання, розроблені в [22].

Враховуючи, що

$$T_f = T_{max} + t, \tag{1.4}$$

Отримаємо

$$T_{max} = T_f - t, \tag{1.5}$$

де T_{max} – максимально допустима тривалість формування навчальної вибірки, t – термін навчання НМ.

Термін навчання НМ в значній мірі залежить від архітек тури НМ. Відповідно висновків [19, 20, 23, 26], найбільше поширення в СВА отримали НМ типу багатошарового персептрону (БШП), мережі Кохонена (SOM), ймовірнісної мережі (PNN) та мережі радіальної базисної функції (РБФ). БШП вважається найбільш "інтелектуальною" НМ, SOM придатна до самонавчання, а перевагами PNN та РБФ є простота і надійність. Можна вважати, що для SOM, PNN та РБФ термін навчання прямо пропорційний кількості навчальних прикладів [21]

$$t \approx \mu_1 \tau e^{-\varepsilon} P(N_X + N_Y), \tag{1.6}$$

де P – кількість навчальних прикладів, ε – допустима помилка навчання НМ, τ – тривалість однієї обчислювальної операції процесу навчання, N_x – кількість вхідних параметрів, N_Y – кількість вихідних параметрів, $\mu_1 \approx 0{,}1$ – коефіцієнт пропорційності.

Зазначимо, що відповідно [26] при моделюванні НМ на сучасних персональних комп'ютерах, $\mu_1 \approx$ 0,01с. Для БШП термін навчання знаходиться в квадратичній залежності від кількості навчальних прикладів

$$t \approx \mu_2 \tau\, e^{-\chi\varepsilon} P^2 (N_X + N_Y)^2, \tag{1.7}$$

де $\chi \approx 1$ – емпіричний коефіцієнт, $\mu_2 =$ 0,001 – коефіцієнт пропорційності.

Зазначимо, що величини коефіцієнтів μ_1 та μ_2 визначені шляхом експертного оцінювання з використанням результатів [21].

Оскільки $T_{max} = f(P)$ і $t = f(P)$, то з урахуванням (1.4) отримаємо

$$T_f = f(P), \tag{1.8}$$

де P– кількість навчальних прикладів (обсяг навчальної вибірки).

Для доведення (1.6) використано нас тупний підхід. В загальному випадку термін навчання НМ розраховується за допомогою сформованого в [21] рівняння

$$t = \xi\tau, \tag{1.9}$$

де ξ – кількість обчислювальних операцій, необхідних для запам'ятовування навчальних прикладів.

При цьому мінімальна та максимальна кількість обчислювальних операцій для запам'ятовування P статистично подібних навчальних прикладів розраховується за допомогою наведених в [14, 15] виразів:

$$\xi_1 \min \uparrow opt \approx 0{,}4\, \mu_{\downarrow 2}\, e\uparrow(-\chi\varepsilon)\, \mu P\uparrow 2 \quad [\![(N_1 X + N_1 Y)]\!]\uparrow 2, \tag{1.10}$$

$$\xi_1 \max \uparrow opt \approx 4\, \mu_{\downarrow 2}\, e\uparrow(-\chi\varepsilon)\, P\uparrow 2 \quad [\![(N_1 X + N_1 Y)]\!]\uparrow 2, \tag{1.11}$$

де $\xi_{\min}^{opt}$ $\xi_{\max}^{opt}$ кількість обчислювальних операцій для багатошарового персептрону з кількістю схованих нейронів, яка дорівнює нижній та верхній межі оптимального діапазону, P – кількість навчальних прикладів.

Після підстановки (1.10, 1.11) в (1.9) одержимо:

$$0{,}4 e^{-\chi\varepsilon} \mu_2 \tau P^2 (N_X + N_Y)^2 \le t \le 4 e^{-\chi\varepsilon} \mu_2 \tau P^2 (N_X + N_Y)^2, \tag{1.12}$$

Вираз (1.7) отримуємо після тривіальних спрощень та перетворень виразу (1.12).

В підсумку, аналіз умов, пов’язаних з навчанням НМ, дозволяє стверджувати, що перспективи використання нейромережевих засобів виявлення кібератак певного типу в основному зумовлюється:

- можливістю реєстрації експлуатаційних параметрів ІС, котрі будуть використані в якості вхідних параметрів НМ;

- можливістю в прийнятний термін сформувати мінімально допустиму навчальну вибірку та провести навчання НМ.

В першому наближенні оцінити вище зазначені можливості доцільно за допомогою методу, який складається із наступних етапів:

1. Оцінити можливість отримання статистичних даних, які можна використати в якості навчальної вибірки НМ, призначеної для виявлення атак певного типу. В процесі оцінки слід врахувати можливість реєстрації функціональних параметрів ІС, котрі можна використати в якості вхідних параметрів НМ, наявність достатнього обсягу зареєстрованих даних та можливість експериментального створення статистичних даних.

2. Визначити номенклатуру вхідних та вихідних параметрів НМ.

3. Базуючись на результатах [25, 26], визначити допус тиму помилку навчання НМ.

4. Базуючись на результатах [26], визначити архітек туру НМ.

5. Відповідно виразу (1.2), розрахувати мінімально допустиму кількість навчальних прикладів.

6. Відповідно виразів (1.6, 1.7), враховуючи тип НМ, розрахувати термін навчання НМ на мінімально допустимій кількості навчальних прикладів.

7. Визначити допустимий термін навчання НМ.

8. Оцінити можливість навчання НМ на мінімально допустимій кількості навчальних прикладів за допустимий термін навчання.

9. Використовуючи експертні дані та вираз (1.3), визначити максимально допустиму тривалість розробки НМ.

10. Використовуючи вираз (1.5), розрахувати максимально допустимий термін формування навчальної вибірки.

11. За допомогою експертних даних оцінити можливість формування мінімально допустимої кількості навчальних прикладів за максимально допустимий термін формування навчальної вибірки.

12. Застосовувати НМ доцільно тільки при позитивних оцінках восьмого та одинадцятого етапів.

Розглянемо роботу методу на конкретному прикладі виявлення кібератак типу IP-спуфінг.

Етап 1. Відповідно [24, 27], для виявлення атак даного типу необхідна статистика, яка стосується наступних функціональних параметрів: кількість одночасних підключень, швидкість обробки запитів, затримка між запитами, кількість пакетів з однаковими адресами відправника та отримувача, вік віртуального каналу та кількість віртуальних каналів. Реєстрацію таких параметрів можливо здійснити мережевими екранами та СВА. В якості доступної бази даних можливо використати KDD-99 [27]. Тому оцінка першого етапу позитивна.

Етап 2. Номенклатура вхідних параметрів НМ буде відповідати вказаним функціональним параметрам.

Кількість вхідних параметрів $N_X = 5$. Для виявлення атаки даного типу можливо обмежитись одним вихідним параметром, величина якого буде вказувати на впевненість СВА у наявності/відсутності атаки типу IP-спуфінг. Тобто $N_Y = 1$.

Етап 3. Базуючись на результатах [25, 26], визначено, що допустима помилка навчання НМ $\varepsilon = 0{,}05$.

Етап 4. Використавши результати [26], визначено, що слід використати БШП.

Етап 5. Підставивши $N_X = 5$ в вираз (1.2), отримаємо: $P_{min} \geq (10..20) \times N_X = 20 \times 5 = 100$. Таким чином, мінімальна кількість навчальних прикладів дорівнює $P_{min} = 100$.

Етап 6. Підставивши в вираз (1.7) $P_{min} = 100$, $N_X = 5$, $N_Y = 1$, $\mu_2 = 0{,}01$, $\chi = 1$, $\tau = 10^{-2}$, отримано

$$t \approx \mu_2 \tau e^{-\chi\varepsilon} P^2 (N_X + N_Y)^2 = 0{,}001 \times 10^{-2} \times e^{-1\times 0{,}05} \times 100^2 \times (5+1)^2 = 34{,}24\ c .$$

Етап 7. Відповідно [25, 26], допустимий термін навчання НМ становить с (24 години).

Етап 8. Оскільки $t < t_d$, то оцінка можливості навчання НМ на мінімально допустимій кількості навчальних прикладів за допустимий термін позитивна.

Етап 9. На основі експертного оцінювання визначено, що термін, протягом якого ризик від реалізації кібератаки не перевищує встановлену межу, становить $T_a = 30$ діб. Підставивши отриману величину в вираз (1.3) отримано $T_f \leq 30$. Таким чином, максимально допустима тривалість розробки НМ становить 30 діб.

Етап 10. Підставивши $T_f = 30$ діб = 2592000 с та $t =$ 34,24 в вираз (1.5), отримано $T_{max} = T_f - t = 2592000 - 34{,}24 = 2591965{,}76 c$

Етап 11. За допомогою експертних даних визначено можливість формування мінімально допустимої кількості навчальних прикладів оцінювання $P_{min} = 100$ протягом $T_{max} =$ 2591965,76 с. Таким чином, оцінка даного етапу позитивна.

Етап 12. Оскільки оцінки восьмого та одинадцятого етапу позитивні, то зроблено висновок про можливість застосування НМ для виявлення кібератак типу IP-спуфінг.

Зазначимо, що використання запропонованого методу багато в чому ускладнюється необхідністю залучення висококваліфікованих експертів, знання яких необхідні для оцінки можливості формування в прийнятний термін мінімально допустимої навчальної вибірки (дев'ятий та одинадцятий етапи).

Разом з тим, достатньо відом і БД, в яких представлені образи кібератак, які можуть бути використані в якості навчальних прикладів нейромережевих засобів розпізнавання. Очевидно, якщо кількість таких образів більша, ніж мінімально допустима кількість навчальних прикладів, то дев' ятий та десятий етапи виконувати не потрібно, а оцінка одинадцятого пункту позитивна. Тому в спрощеному варіанті методу замість 9-11 етапів слід оцінити можливість

формування мінімальної навчальної вибірки на основі доступних баз даних образів кібератак.

1.2 Аналіз сучасних методів розпізнавання кібератак

Інформаційна безпека — це стан захищеності систем обробки і зберігання даних, при якому
забезпечено конфіденційність, доступність і цілісність інформації, або комплекс заходів, спрямованих на забезпечення захищеності інформації від несанкціонованого доступу, використання, оприлюднення, руйнування, внесення змін, ознайомлення, перевірки, запису чи знищення (у цьому значенні частіше використовують термін «захист інформації»).

Інформаційна безпека держави характеризується ступенем захищеності і, отже, стійкістю основних сфер життєдіяльності (економіки, науки, техносфери, сфери управління, військової справи, суспільної свідомості і т. д.) по відношенню до небезпечних (дестабілізуючих, деструктивних, суперечних інтересам країни тощо), інформаційним впливам, причому як до впровадження, так і до вилучення інформації

Поняття інформаційної безпеки не обмежується безпекою технічних інформаційних систем чи безпекою інформації у чисельному чи електронному вигляді, а стосується усіх аспектів захисту даних чи інформації незалежно від форми, у якій вони перебувають.

Аналіз науково-практичних робіт, присвячених вдосконаленню систем виявлення атак (СВА) дозволяє стверджувати, що в таких системах НМ застосовуються для виявлення атак на основі узагальнення статистичних даних, відображених в навчальних прикладах [18, 19, 24, 26, 28, 29, 30]. Крім того, можна зробити висновок, що більшість відповідних науково-практичних робіт присвячені адаптації архітектури НМ до умов поставленої задачі. Так в роботі [26] розроблено метод визначення оптимального типу архітектури НМ. Робота [29] присвячена задачам вдосконалення структури та алгоритму навчання

багатошарового персептрону, призначеного для використання в СВА.

Ще одним напрямком досліджень є розробка підходів до використання нових малоапробованих нейромережевих моделей. Наприклад, робота [19] присвячена засобам розпізнавання на базі кібернейрону, а в роботі [30] пропонується використовувати карту Кохонена, що функціонує відповідно принципів штучних імунних систем.

Разом з тим критики використання НМ вказують на те, що в багатьох випадках мережева атака являє собою набір нестандартних операцій, характеристики яких не відображаються в зареєстрованих статистичних даних – навчальних прикладах. Відповідно і розпізнати новий тип мережевої атаки за допомогою НМ можливо тільки після її реалізації. Таким чином, використанню НМ заважає її суттєвий недолік – погана адаптація до нових типів мережевих атак. Наведене твердження дещо суперечливе. Наприклад, в роботі [18] запропоновано методи моделювання параметрів, які характеризують мережеву атаку. Однак традиційний підхід до використання НМ виключно в якості статистичного аналізатора, становить серйозну перепону їх подальшому впровадженню в СВА. На нашу думку виправити вказаний недолік можливо за рахунок використання в НМ експертних знань, що дозволить в першу чергу підвищити оперативність розпізнавання нових типів мережевих атак характеристики яких не представлені в навчальній вибірці, яка сформована на основі зареєстрованих статистичних даних. Крім того, розширюється множина видів мережевих атак яку може розпізнати СВА. При цьому в доступній літературі не знайдено методу подання експертних знань в НМ, призначених для використання в контурах розпізнавання систем захисту інформації, хоча в [27] запропоновані відповідні загальнотеоретичні підходи.

Слід зазначити, що на сьогодні відомі різноманітні підходи до подання та використання експертних знань. На початковому етапі досліджень доцільно орієнтуватись на базові методи подання. Один із таких методів базується на продукційних правилах типу:

Якщо умова істина\хибна →(Висновок)

Зазначимо, що продукційні правила дозволяють описати експертні знання у вигляді взаємозв'язків: «причина» → «наслідок», «явище» → «реакція», «ознака» → «факт». Крім того, застосування логічних операторів дозволяє проводити комбінування взаємозв'язків. Очевидно, що вказані взаємозв'язки та їх комбінації можуть бути використані для подання експертних знань щодо виявлення мережевих атак на основі аналізу параметрів, котрі характеризують функціонування КС. Таким чином метою даної статті є розробка методу подання експертних знань в нейромережевих засобах розпізнавання мережевих атак на КС за рахунок застосування продукційних правил.

1.3. Перспективні шляхи вдосконалення систем розпізнавання кібератак

Призначенням етапу є розроблення множини ефективних НМЗ та визначення шляхів їх можливого вдосконалення. Для цього використовується розроблений метод оцінювання ефективності. Вхідними даними етапу є $У, О, \mathbf{M}_{ve}, M_{vg}, D_{\min}$. Етап виконується за два кроки.

Крок 1 – розрахунок показників ефективності. На цьому кроці визначаються величини елементів ***Ф,D,A***, за допомогою яких для кожного $m_i \in M_{ve}$ розраховується інтегральний показник ефективності D_i^{Σ}.

Крок 2 – порівняння ефективності. Крок призначено для формування $\mathbf{M_e}$ – множини ефективних НМЗ $\mathbf{M_e}$ та визначення найбільш ефективного НМЗ. Для формування $\mathbf{M_e}$ використовується правило

$$якщо\ D_i^{\Sigma} > D_{\min} \rightarrow m_i \in \boldsymbol{M_e}.$$

Правило для визначення найбільш ефективного НМЗ виглядає так:

$$якщо\ D_i^{\Sigma} = \max(D) \wedge \left(D_i^{\Sigma} > D_{\min}\right) \rightarrow m_i = m^{\max}.$$

Виходом цього етапу є $\mathbf{M_e}$ – множина ефективних НМЗ і m^{max}– найбільш

ефективний НМЗ. Проведено порівняння ефективності відомих нейромережевих методів розпізнавання кібератак та запропонованої методології нейромережевого оцінювання ПБ ІС – КМНО. Для цього застосовано розроблений метод оцінювання ефективності. Величини базових критеріїв оцінювання ефективності наступні: $\varphi_{по} = 0$, $\varphi_{ота} = 1$, $\varphi_{бва} = 1$, $\varphi_{опа} = 1$, $\varphi_{бпа} = 0$, $\varphi_{омн} = 0$, $\varphi_{веп} = 1$, $\varphi_{мна} = 1$.

Визначені вагові коефіцієнти базових критеріїв оптимізації: $\alpha_{1,ота} = 0{,}5$, $\alpha_{1,бва} = 1$, $\alpha_{1,опа} = 0{,}5$, $\alpha_{1,бпа} = 1$, $\alpha_{1,омн} = 1$, $\alpha_{1,мна} = 0{,}5$, $\alpha_{2,одв} = 1$, $\alpha_{3,веп} = 1$, $\alpha_{3,мна} = 0{,}5$, $\alpha_{4,ота} = 0{,}5$, $\alpha_{4,бва} = 1$, $\alpha_{4,мна} = 1$, $\alpha_{5,по} = 0{,}5$, $\alpha_{5,ота} = 0{,}5$, $\alpha_{5,бва} = 1$, $\alpha_{5,опа} = 0{,}5$, $\alpha_{5,бпа} = 1$, $\alpha_{5,омн} = 0{,}5$, $\alpha_{5,мна} = 0{,}5$. В першому наближенні припускається, що всі вагові коефіцієнти інтегральних критеріїв дорівнюють $\gamma = 1$

1.4. Висновки до першого розділу

Проведений в першому розділі аналіз проблеми захисту комп'ютерних систем від мережевих кібератак дозволяє стверджувати:

- мережеві кібератаки являються однією із найбільших загроз інформації для більшості комп'ютерних систем;
- одним з основних рубежів захисту є профілактика мережі та своєчасне оновлення програмного забезпечення;
- основною проблемою реалізації профілактичних заходів є проблема розпізнавання нових видів кібератак;
- недоліки методик розпізнавання мережевих кібератак пов'язані з недостатньою ефективністю математичного забезпечення вказаних методики;
- важливим напрямком вдосконалення захисту від кібератак є підвищення ефективності методики розпізнавання;
- підвищити ефективність методик розпізнавання можливо за рахунок використання такого напрямку розвитку теорії штучного інтелекту, як НМ.

РОЗДІЛ 2

ВИКОРИСТАННЯ ЕКСПЕРТНИХ СИСТЕМ ДЛЯ РОЗПІЗНАВАННЯ КІБЕРАТАК

2.1. Аналіз сучасного стану найромережевих технологій

Проведений аналіз сучасного стану найромережевих технологій дозволяє сформулювати висновок про те, що доцільність застосування конкретного типу НМ слід визначати на основі співставлення характеристик мережі з умовами прикладної задачі. До вказаних характеристик та умов відносяться:

1 – параметри навчальних даних,

2 – загальні обмеження процесу навчання,

3 – вимоги до обчислювальних потужностей,

4 – вимоги до вихідної інформації,

5 – обмеження технічної реалізації НМ,

6 – сфера застосування.

Розглянемо вказані характеристики в ракурсі захисту ПЗ комп'ютерних систем.

1. До основних параметрів навчальних даних відносяться:

– Кількість параметрів, що характеризують навчальний приклад.

– Вид параметрів, дискретний (символьний) чи безперервний (числовий).

– Загальна кількість навчальних прикладів.

– Наявність помилок (шуму) в навчальних прикладах.

– Наявність кореляції навчальних прикладів.

– Можливість та необхідність попередньої обробки вхідних даних з метою їх нормалізації та видалення шуму.

– Повнота вибірки, тобто можливість відображення в ній всіх аспектів процесу, що моделюється. Наприклад, чи можливо відобразити в навчальній виборці сигнатури всіх вірусів, або сигнатури мережевих атак певного типу.

– Пропорційність навчальних прикладів, що відповідають різним аспектам процесу, що моделюється. Наприклад скільки навчальних прикладів відповідають мереженій атаці типу А, а скільки прикладів – атаці типу В.

2. Загальні обмеження процесу навчання обумовлюються:

– Максимальним терміном навчання.

– Необхідністю представлення в навчальних даних очікуваного вихідного сигналу НМ. Цим визначається тип навчання – з вчителем або без вчителя.

– Можливістю автоматизації процесу навчання, яка визначається кількістю та важливістю емпіричних параметрів. Вказана можливість багато в чому визначає умови застосування НМ. Мережі в яких процес навчання не автоматизовано можуть використовуватись тільки в лабораторних умовах.

– Можливістю донавчання в процесі експлуатації.

– Вимогами до якості навчання, яке звичайно оцінюють по величині максимальної та середньої помилки розпізнавання навчальних та тестових даних. При цьому тестові дані повинні не значно відрізнятись від навчальних.

– Можливістю навчання НМ в лабораторних умовах. Наприклад, в лабораторних умовах потенційно можливо навчити НМ розпізнавати мережеві атаки певного типу. Доцільність навчання в лабораторних умовах пояснюється потребами оптимального механізму створення та оновлення бази знань НМ.

3. На практиці вимоги до обчислювальних потужностей визначаються максимальною кількістю прикладів (обсяг пам'яті), яку може запам'ятати мережа для досягнення необхідної достовірності прийняття рішення. В свою чергу достовірність прийняття рішення характеризується допустимими величинами максимальної та середньої помилки мережі на реальних даних які в загальному випадку можуть виходити за межі множини навчальних даних. Відповідно виникає задача екстраполяції результатів навчання НМ за межі навчальних прикладів. Відзначимо, що обчислювальна потужність мережі залежить від її типу

та алгоритму навчання. Ще однією вимогою може бути незмінність виходу мережі для різних прикладів з однаковими параметрами.

4. Вимоги до вихідної інформації НМ вказують на те в якому вигляді має бути представлена ця інформація. Наприклад, при розпізнавані кібератак може виникнути необхідність не тільки визначення ситуації “кібератака А присутня”, але й розрахунку ймовірності цієї ситуації. Ще однією вимогою може бути необхідність визначення вербальних залежностей між вхідною та вихідною інформацією.

5. Обмеження технічної реалізації НМ стосуються: швидкості прийняття рішення, інтеграції в існуючі ЗЗІ, обсягу та складності програмної реалізації. Для зменшення обсягу можливо розділити програмний код для навчання мережі від коду, що відповідає за її функціонування.

6. Сфера застосування визначає ЗЗІ в яких буде використовуватись НМ. На сьогодні достатньо дослідженим є використання НМ для розпізнавання образів та при проведенні оптимізаційних розрахунків. Відзначимо, що системи розпізнавання образів принципово відрізняються від систем аналізу тексту тим, що в них кількість вихідних та кількість комбінацій вхідних параметрів принципово обмежена. В системах аналізу тексту ця кількість принципово необмежена. Відповідно в СВА та СВВ слід використовувати НМ призначені для розпізнавання образів. В системах керування параметрами ЗЗ слід застосувати НМ призначені для проведення оптимізаційних розрахунків. В перспективі доцільно застосувати НМ з метою реалізації паралельних розрахунків в КС, що дозволить значно підвищити їх стійкість від багатьох типів атак з метою відмови в обслуговуванні.

2.2. Використання експертних систем

Основним призначенням експертних систем є розробка програмних продуктів, які при вирішенні задач, складних для людини, отримують результати, що не поступаються за якістю та ефективністю розв’язку, розв’язкам, отриманим

людиною-експертом. Експертні системи використовуються для вирішення так званих неформалізованих задач, загальним для яких є: задачі не можуть бути задані у числовій формі; цілі не можна виразити у термінах точно визначеної функції цілі; алгоритмічного розв'язку задачі не існує, або, в іншому випадку, його не можна використовувати через обмеженість ресурсів (час, пам'ять). Окрім того, у неформалізованих задачах присутня помилковість, неповнота, неоднозначність та протиріччя як вихідних даних, так і знань про задачу, що вирішується.

Експертна система (ЕС) – це програмний продукт, що використовує експертні знання для забезпечення високоефективного розв'язання неформалізованих задач у вузькій предметній області. Основу експертних систем складає база знань з предметної області, яка накопичується в процесі як побудови, так і експлуатації експертної системи. Дослідники в області ЕС для назви своєї дисципліни часто використовують також термін «інженерія знань», що позначає привнесення принципів та інструментарію досліджень із області штучного інтелекту в рішення складних прикладних проблем, які вимагають знань експертів [1]. Існують певні технології розробки ЕС, що складаються з таких шести етапів: ідентифікація, концептуалізація, формалізація, реалізація, тестування і дослідно-експериментальна експлуатація. На етапі ідентифікації визначаються задачі, які підлягають розв'язанню, виявляються проміжні цілі розробки, визначаються експерти за напрямом і типи користувачів.

Етап концептуалізації призначений для змістовного аналізу проблемної області, виявлення наявної інформації та визначення множини альтернативних методів розв'язання поставлених задач. На етапі формалізації обирається інструментарій і визначаються способи зберігання та представлення всіх типів знань, формалізуються основні поняття, визначаються способи інтерпретації знань, моделюється робота системи, оцінюється адекватність понять, методів розв'язання, засобів подання і маніпулювання знаннями. На етапі виконання здійснюється наповнення експертом бази знань. Розповсюджені такі підходи до розробки ЕС: системи на основі правил, системи з використанням нейронних

мереж та нечіткої логіки (або нейронечіткі систем), експертні системи на основі мереж довіри Байєса та інші. Оскільки ЕС призначені для неформа- лізованих завдань, то вони не відкидають і не заміняють традиційного підходу до розробки програм, орієнтованих на рішення формалізованих завдань. На практиці часто доводиться оцінювати гіпотези, для яких є лише неповна чи недостатня інформація. Іноді важко зробити точні оцінки, але, не зважаючи на невизначеність експерти приймають правильні рішення. Щоб ЕС були корисними, вони також мають вміти це робити.

2.3. Формування навчальної вибірки НММ(БД)

Для формування навчальної вибірки пропонується використати базу даних KDD-99. Наведено словесний опис та фрагменти програмного коду для підготовки вхідних даних із цієї бази даних до виду вхідних параметрів НМ. Однією із цілей підготовки є зменшення обсягу навчальної вибірки НМ. Підходи до оптимізації виду та параметри НММ не описуються.

Розрахунок максимально допустимої тривалості формування навчальної вибірки.

На цьому етапі для кожної $m_j^{(tn)} \in M^{(tn)}$ розраховується максимально допустима тривалість формування навчальної вибірки $T_{j,max}$. Оскільки в MPNN як навчальні приклади використовуються експертні дані, то для НМЗ на базі цієї НММ тривалість формування навчальної вибірки дорівнює тривалості розроблення продукційних правил.

Для розрахунку $T_{j,max}$ використовується вираз

$$T_{j,max} = T_f - t_j \ , \qquad (2.1)$$

де t_j – термін навчання j-ї НММ $m_j^{(tn)} \in M^{(tn)}$.

Результатом виконання етапу є сформована множина

$$T_{l,max}, \ldots T_{L,max}, \tag{2.2}$$

де L – кількість елементів $M^{(tn)}$.

Визначення терміну формування навчальної вибірки.

Етап орієнтовано на аналіз характеристик об'єкта захисту та умов оцінювання для визначення терміну формування навчальної вибірки з мінімально допустимою кількістю навчальних прикладів:

$$T_d = f(O, У), \tag{2.3}$$

де T_d – термін формування навчальної вибірки.

Перевірка терміну формування навчальної вибірки.

На цьому етапі для кожної $m_j^{(tn)} \in M^{(tn)}$ порівнюються $T_{j,max}$ і T_d. Множина НМЗ, які доцільно використовувати для оцінки ПБ, формується за допомогою виразів:

$$\text{якщо } T_{j,max} > T_d \rightarrow m_j^{(tn)} \in Mz, \tag{2.4}$$

$$\text{якщо } T_{j,max} > T_d \rightarrow m_j^{(tn)} \in Mz, \tag{2.5}$$

де Mz – множина НМЗ, які доцільно використовувати для оцінки ПБ.

У результаті реалізації методу формується множина НМЗ, які доцільно використовувати для оцінювання ПБ з метою розпізнавання кібератак.

Зазначимо, що використання запропонованого методу багато в чому ускладнюється необхідністю залучення висококваліфікованих експертів, знання яких потрібно для оцінювання можливості формування в прийнятний термін мінімально допустимої навчальної вибірки. Разом з тим достатньо відомі бази даних, у яких подані образи кібератак, які можуть бути використані як навчальні приклади нейромережевих засобів розпізнавання. Очевидно, якщо кількість таких образів більша від мінімально допустимої кількості навчальних прикладів, то

дев’ятий та десятий етапи виконувати не потрібно, а оцінка одинадцятого пункту позитивна. Тому в спрощеному варіанті методу слід оцінити можливість формування мінімальної навчальної вибірки на підставі доступних баз даних образів кібератак.

2.4. Використання PNN для формування експертних правил

Аналіз [1, 9, 12] відомих прикладів правил визначення безпечного/ небезпечного стану КС, що застосовуються в СВА, вияв дві властивості які недостатньо враховуються в структурі та математичному забезпеченні класичної мережі PNN:

1. Кожному окремому типу мережевої атаки може відповідати одна комбінація параметрів захищеності. Тобто кількість класів, що розпізнаються, може дорівнювати кількості навчальних прикладів. Таким чином, кількість нейронів в ШД буде дорівнювати кількості нейронів в ШО. Очевидно, що в таких випадках використання ШД буде недоцільним. Вихідний сигнал від нейронів ШО може безпосередньо подаватись до нейрону ШВ. Відповідно змінена структура мережі PNN показана на рис. 2.1.

2. У багатьох випадках умова, яка використовується в продукційних правилах (1) має наступний вигляд:

$$p_1 \in [P_1^{min}, P_1^{max}] \wedge p_2 \in [P_2^{min}, P_2^{max}] \wedge \cdots, \qquad (2.6)$$

де p_1, p_2 .., – підконтрольні параметри, $[P_1^{min}, P_1^{max}]$ „… – задані діапазони величин підконтрольних параметрів.

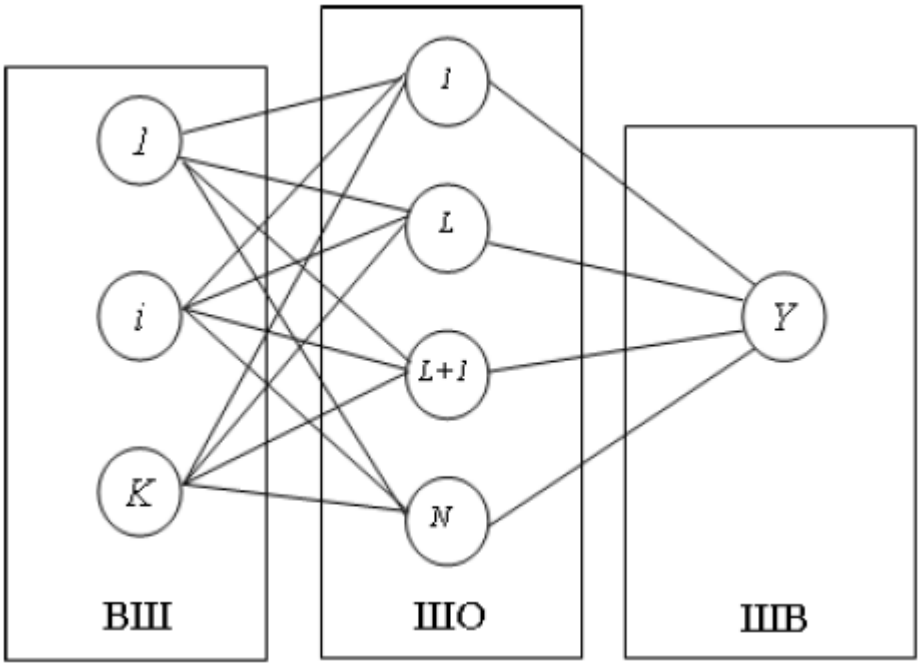

Рисунок. 2.1. - Структура мережі PNN без ШД

Безпосереднє визначення такого правила в класичній моделі PNN неможливе, оскільки лінійна активаційна функція ШО не в змозі відобразитискладову

$$p_1 \in [P_1^{min}, P_1^{max}]. \quad (2.7)$$

Разом з тим, умову (5) можна представити за допомогою системи вигляду:

$$\begin{cases} p_1 = P_1^{min} \wedge p_2 = P_2^{min} \wedge \cdots \\ p_1 = P_1^{min} + \Delta_1 \wedge p_2 = P_2^{min} + \Delta_2 \wedge \cdots \\ p_1 = P_1^{max} \wedge p_2 = P_2^{max} \wedge \cdots \end{cases}, \quad (2.8)$$

де $\Delta_1, \Delta_2, \ldots$ - задані коефіцієнти.

Однак використання виразу (2.8) призводить до вагомого ускладнення НМ за рахунок значного збільшення кількості нейронів ШО. Можливим шляхом адаптації моделі PNN до умови (2.6) є введення до її складу проміжного (фільтруючого) шару нейронів, завданням якого буде фільтрація вхідного сигналу відповідно виразу (2.7). Вказаний фільтруючий шар (ШФ) має знаходитись між ВШ та ШО. Структура модифікованої мережі PNN, показана на рис. 2.2.

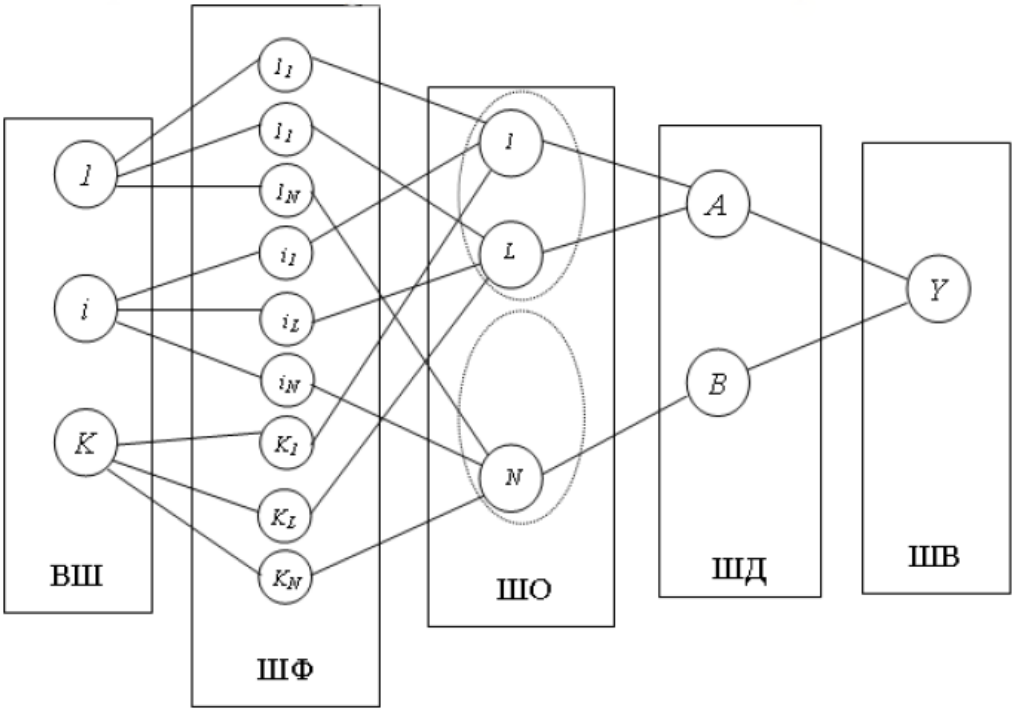

Рисунок. 2.2. - Структура модифікованої мережі PNN

Зазначимо, що кожен нейрон ВШ пов’язаний з такою кількістю нейронів ШФ, яка дорівнює кількості нейронів ШО. Прицьому кожен нейрон ШФ пов’язаний тільки з одним нейроном ШО, для якого власне і реалізується фільтрація вхідного сигналу. Для зручності нейрони ШО пронумеровані як i_L, де i – номер пов’язаного з ним вхідного нейрону, а L– номер пов’язаного з ним нейрону шару образів. Відповідно [10] для реалізації фільтру (2.7) в проміжних нейронах слід застосувати функцію активації вигляду

$$\begin{cases} 0 \text{ при } x \le P^{min} \\ Kx + A \text{ при } P^{min} \le x \le P^{max} \\ 0 \text{ при } x \ge P^{\max} \end{cases}, \qquad (2.9)$$

де К та А – деякі коефіцієнти.

Ця функція отримала назву лінійної біполярної з насиченням. Графік даної функції показано на рис. 2.3.

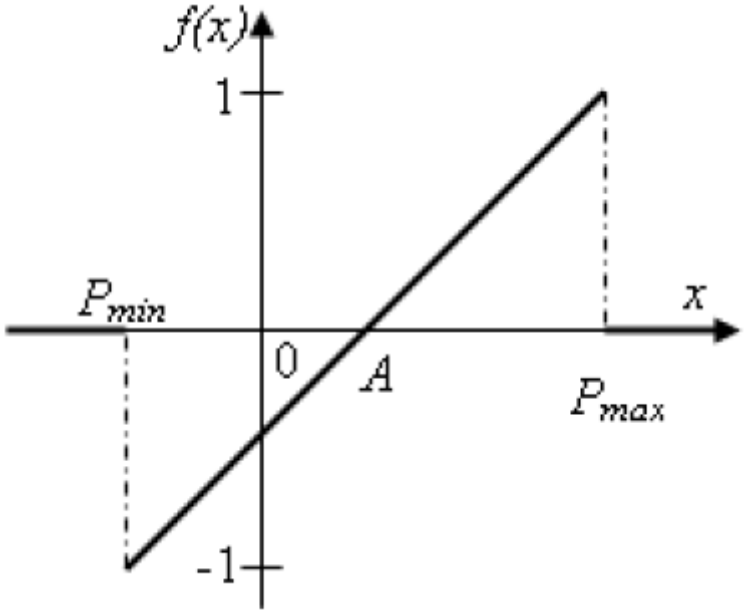

Рисунок. 2.3. - Графік лінійної біполярної функції активації з насиченням

У першому наближенні можна вважати, що К=1, а А=0. Також зазначимо, що по суті нейрони ШФ в модифікованій мережі PNN відіграють роль вагових коефіцієнтів зв'язків між ВШ та ШО у класичній мережі. Тому всі вагові коефіцієнти зв'язків в модифікованій мережі PNN дорівнюють 1. Крім адаптації структури та математичного забезпечення необхідність використання продукційних правил виду (2.6) призводить і домодифікації алгоритму навчання мережі:

− додати в ШО новий нейрон, який буде відповідати новому навчальному прикладу – продукційному правилу;

− в залежності від класифікації навчального прикладу встановити для нового нейрону вихідний зв´язок з відповідним нейроном ШД.

− додати в ШФ нейрони що будуть відповідно виразу (2.7) перетворювати сигнали, які передаються від вхідних нейронів до нового нейрону ШО;

− встановити зв'язки між новим нейроном ШО та новими нейронами ШФ;

− встановити зв'язки між новими нейронами ШФ та відповідними вхідними нейронами.

− встановити в мережі PNN всі вагові коефіцієнти рівними 1.

− співвіднести для нього вагові коефіцієнти вхідних зв´язків з величинами параметрів які відповідають заданому прикладу безпечного стану або реалізації атаки.

В підсумку узагальнений метод подання експертних знань в модифікованій мережі PNN, призначеній для розпізнавання мережевих атак на КС за рахунок застосування продукційних правил, складається з наступних етапів:

1. Використовуючи технології обробки експертної інформації [7, 8] визначити:

− множину параметрів захищеності КС.

− множину станів яку повинна розпізнавати НМ. В найпростішому випадку НМ буде розпізнавати всього два стани КС – безпечний та небезпечний.

− множину продукційних правил.

2. Визначити множину вхідних параметрів НМ, що співвідносяться з параметрами захищеності. При цьому можливо застосувати запропонований в [10] метод кодування параметрів захищеності до виду прийнятного НМ.

3. Визначити в ШД стільки нейронів, скільки класів повинна розпізнавати НМ.

4. Використовуючи розроблений алгоритм навчання визначити структуру та вагові коефіцієнти зв’язків.

2.4. Висновки до другого розділу

Другий розділ присвячено розробці методики використання експертних знань в системах розпізнавання мережевих кібератак. В результаті проведених досліджень доведено:

- розглянута методика використання експертних знань для побудови побальшої нейромережевої системи розпізнавання мережевих кібератак.
- серед класичних нейромережевих архітектур найбільш придатними для розпізнавання мережевих кібератак є HMM та модель PNN. Використання цих архітектур пояснюється їх високими обчислювальними здатностями, апробацією при вирішенні подібних задач та можливістю реалізації навчання "з вчителем";
- розроблена методика адаптації архітектури HMM для вирішення задач ЗІ, в тому числі і до задачі розпізнавання мережевих кібератак. Адаптація полягає у визначенні кількості схованих нейонних шарів та кількості схованих нейронів. Показано потенційну можливість формування явних правил прийняття рішень про наявність/відсутність вірусів;
- розроблена методика адаптації архітектури мережі HMM до вирішення задачі розпізнавання мережевих кібератак. Розроблене відповідне математичне забезпеченя. Доведено доцільність використання модель PNN тільки на ранніх етапах розробки системи розпізнавання кібератак.

РОЗДІЛ 3
РЕАЛІЗАЦІЯ СИСТЕМИ РОЗПІЗНАВАННЯ КІБЕРАТАК

3.1. Побудова архітектури системи розпізнавання кібератак

Насамперед слід відзначити, що концептуально теорія НМ об’єднує багато досить різнотипних моделей. Спільною рисою цих моделей є методологія обробки даних, яка полягає в:

− отриманні зовнішнього сигналу;

− передачі отриманого сигналу до штучних нейронів по синаптичним (зваженим) зв´язкам;

− обробці в штучному нейроні отриманого сигналу шляхом застосування активаційної функції.

При цьому кожен синаптичний зв’язок може мати свій унікальний ваговий коефіцієнт, попередньо визначений при навчанні НМ в процесі подання навчальних прикладів. Саме багатоітераційна процедура подання деяким типам НМ навчальних прикладів – статистичних даних – є підґрунтям для їх інтерпретації як аналізатора статистичної інформації. До цих типів НМ передусім відносяться: багатошаровий персептрон, карта Кохонена, машина Больцмана. Однак достатньо відомі та апробовані інші типи НМ, котрі навчаються методом «з вчителем» шляхом запам’ятовування представлених навчальних прикладів, які в певному сенсі можна вважати аналогом продукційних правил типу (3.1). Адже по своїй суті окремий навчальний приклад це комбінація продукційних правил:

$$\text{Якщо } X_1 = a_1 \wedge X_2 = a_2 \wedge \cdots \rightarrow Y \quad , \qquad (3.1)$$

де – і-ий вхідний параметр, – очікуваний вихід НМ.

Таким чином в НМ, що навчається шляхом запам’ятовування навчальних прикладів, можливо подати експертні знання у вигляді продукційних правил. При

цьому вважається [12], що з точки зору застосування в СВА серед таких мереж високий потенціал має ймовірнісна НМ (PNN – Probabilistic Neural Network). Класифікація невідомих прикладів реалізується мережею PNN на основі оцінок їх схожості з навчальними прикладами за допомогою методу Байєса [11,12]. Невідомий приклад відноситься до класу, у якого щільність розподілу в області даного прикладує найбільшою. Для оцінки щільності розподілу в області певного навчального прикладу використовується функція Гауса з центром в точці, якій відповідає даний приклад. Класична мережа PNN складається із чотирьох шарів нейронів – вхідного, образів, додавання та вихідного. Кількість нейронів вхідного шару (ВШ) дорівнює кількості контрольованих параметрів аналіз яких дозволяє розпізнати мережеву атаку. Кількість нейронів шару образів (ШО) дорівнює кількості навчальних прикладів, а кількість нейронів шару додавання (ШД) дорівнює кількості класів, які розпізнаються. ВШ та ШО складають повнозв'язну структуру, а кожен нейрон ШО пов'язаний тільки з тим нейроном ШД якому відповідає клас образу. Для зв'язків, що входять в нейрон ШО, вагові коефіцієнти встановлюються такими ж, як нормалізовані складові частини відповідного навчального прикладу. Вагові коефіцієнти зв'язків, що входять до нейронів ШД та до нейрона вихідного шару (ШВ) дорівнюють 1. Таким чином, структура і вагові коефіцієнти зв’язків мережі PNN безпосередньо визначаються навчальними даними. Структура мережі PNN, призначеної для класифікації двох станів A і B показана на рис. 3.1. В цій мережі нейрони ШО з номерами від 1 до L відповідають навчальним прикладам, які спів-відносяться з безпечним станом, а нейрони з номерами від L+1 до N – співвідносяться з реалізацією мережевої атаки. Вихідний сигнал j-го нейрону шару образів розраховується так:

$$\theta_j^0 = \sum_{i=1}^{N} \exp\left(\frac{-\left(w_{i,j} - x_i\right)^2}{2\sigma^2}\right), \qquad (3.2)$$

де x_i − і-а компонента невідомого образу, $w_{i,j}$ – ваговий коефіцієнт зв’язку між і-им вхідним нейроном та j-им нейроном шару образів, N − кількість компонент вхідного вектора-образу, σ − радіус функції Гауса. У нейронах ШД

використовується лінійна функція активації. Вихідний сигнал j-го нейрону ШД розраховується так

$$\theta_j^s = \sum_{i=1}^{N} \theta_i^0 , \qquad (3.3)$$

де N − кількість нейронів ШО, пов'язаних з j-им нейроном ШД, − активність і-ого нейрону шару образів, пов'язаного з j-им нейроном ШД.

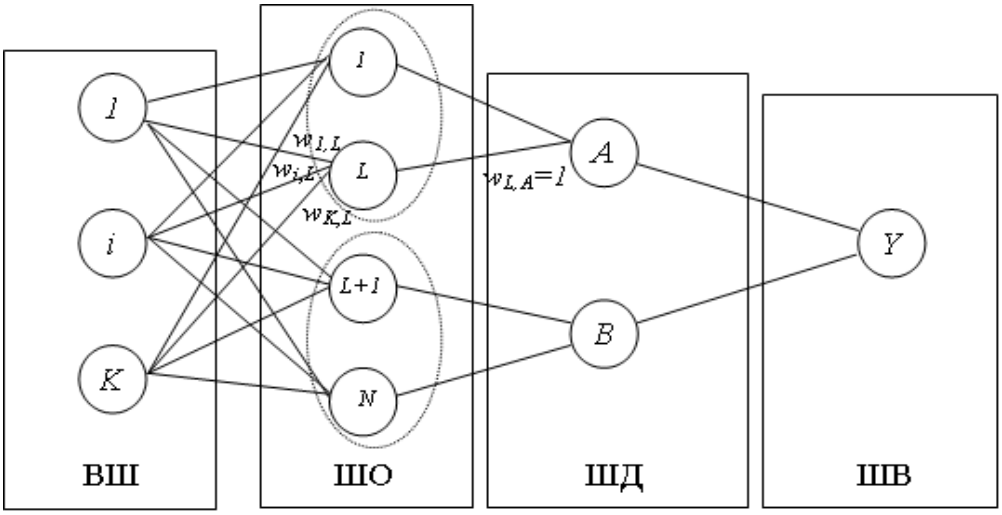

Рисунок. 3.1. - Структура мережі PNN

Вихідний сигнал нейрону ШД дорівнює ймовірності віднесення вхідного образу до класу, що відповідає даному нейрону. Задачею вихідного нейрону є тільки визначення нейрону ШД з максимальною активністю. В багатьох випадках вихідний нейрон відсутній, а визначення нейрону ШД з максимальною активністю реалізують засобами, які не входять до складу нейронної мережі. Зазначимо, що відповідно [10] для підвищення ефективності процесу розрахунку вихідного сигналу мережу PNN доцільно представитив матричній формі. При цьому елементами матриць будуть вагові коефіцієнти зв’язків між сусідніми шарами нейронів. Якщо ж зв’язок між нейронами не передбачено, то вважається що його ваговий коефіцієнт дорівнює 0.

Єдиним емпіричним параметром такої моделі мережі PNN є величина радіуса функції Гауса, що використовується у виразі (3.2) для розрахунку вихідного сигналу нейрону ШО. При цьому в теоретичних роботах [8, 10]

зазначається, що для багатьох практичних випадків в першому наближенні можна приняти $1\sigma = 0$.

В [12] описану мережу PNN пропонується використовувати для розпізнавання мережевих атак за рахунок класифікації одного із двох можливих станів КС:

А – безпечний стан;

В – небезпечний стан – реалізація мережевої атаки.

Безпечний стан співвідноситься з нейроном ШД А, а стан реалізації мережевої атаки – з нейроном ШД В.

На нейрони ВШ подають інформацію, яка відповідає нормалізованим величинам контрольованих параметрів КС, значення яких можуть сигналізувати про наявність/відсутність мережевої атаки – частота мережевих запитів, завантаженість лінії зв´язку, кількість неправильних пакетів, протокол, по якому передаються дані, завантаженість процесора, IP-адреса, з якої передаються дані і т. ін. Кількість вхідних нейронів дорівнює кількості контрольованих параметрів захищеності.

Для внесення в НМ знань про правило класифікації безпечного стану або реалізації атаки достатньо:

− визначити в ШД два нейрони А та В, котрі співвідносяться з безпечним та небезпечним станом КС;

− внести в ШО новий нейрон;

− співвіднести для нього вагові коефіцієнти вхідних зв´язків з величинами параметрів які відповідають заданому прикладу безпечного стану або реалізації атаки;

− встановити для нового нейрону вихідний зв´язок з відповідним нейроном ШД А або В.

Для прикладу на рис. 3.1 показано вагові коефіцієнти $w_{L,1}$, $w_{i,L}$, $w_{K,L}$ та $w_{L,A}$

за рахунок яких в мережу PNN внесено приклад і, який відповідає безпечному стану КС.

3.2. Вибір засобів програмування

Для даної задачі було запропоновано обрати мову програмування – Python, та використовувати вже існуючі модулі для більшої простоти і наглядності коду в програмному продукті. Оскільки Python — інтерпретована мова, математичні алгоритми, часто працюють в ньому набагато повільніше ніж у компільованих мовах, таких як C або навіть Java.

Необхідне технічне забезпечення в себе включає:

- JetBrains PyCharm Community Edition 5.0.1
- Python 2.7/3.1

Необхідні додаткові установки до проекту:

- Pandas
- Numpy
- Time
- Sklearn
- Matplotlib
- Scipy

Описання підключаємих модулів:

Pandas – це бібліотека мови програмування Python для аналіза та обробки даних. Для даного проекту ця бібліотека використовується в аналізі бази KDD-99 з розширення csv, створення та редагування схожих файлів.

Numpy – це розширення мови програмування Python, що додає підтримку

великих багатовимірних масивів та матриць. Разом з цим розширення має велику кількість багаторівневих математичних функцій для операцій з цими ж масивами. NumPy намагається вирішити цю проблему для великої кількості обчислювальних алгоритмів забезпечуючи підтримку багатовимірних масивів і безліч функцій і операторів для роботи з ними. Таким чином будь-який алгоритм який може бути виражений в основному як послідовність операцій над масивами і матрицями працює також швидко як еквівалентний код написаний на C. NumPy можна розглядати як гарну вільну альтернативу MATLAB, оскільки мова програмування MATLAB зовні нагадує NumPy: обидві вони інтерпретовані, і обидві дозволяють користувачам писати швидкі програми поки більшість операцій проводяться над масивами або матрицями, а не над скалярами. Перевага MATLAB у великій кількості доступних додаткових тулбоксів, включаючи такі як пакет Simulink. Основні пакети, що доповнюють NumPy, це: SciPy — бібліотека, що додає більше MATLAB-подібної функціональності; Matplotlib — пакет для створення графіки в стилі MATLAB. Внутрішньо як MATLAB, так і NumPy базується на бібліотеці LAPACK, призначеної для вирішення основних задач лінійної алгебри.

Time – це модуль був використаний для заміру часу роботи певних функцій та програми вцілому.

Sklearn – ця бібліотека дає можливість реалізації цілого ряду алгоритмів для навчання «з учителем» та без нього.

Matplotlib – це бібліотека двовимірної графіки для Python із допомогою якої можна створювати якісні рисунки та графіки різних розширень.

Scipy – відкрита біблиотека високоякісних наукових інструментів для Python. SciPy містить модулі для оптимізації, інтегрування, спеціальних функцій, обробки сигналів, обробки зображень, генетичних алгоритмів, розв'язування звичайних диференціальних рівнянь та інших задач. Бібліотека розробляється для тієї ж аудиторії, що і MATLAB та Scilab. Для візуалізації при використанні SciPy часто застосовують бібліотеку Matplotlib, яка є аналогом засобів виводу графіки MATLAB.

3.3. Експериментальні дослідження

Для перевірки ефективності запропонованого методу проведено експериментальні дослідження в яких модифікована мережа PNN застосовувалась для виявлення мережевих атак.

У якості джерела статистичних даних для формування навчальної та тестової множини НМ використана база даних KDD-99, котра містить близько 500000 записів – образів мережевих з'єднань [15]. Кожен запис складається з 42 полів. В полях від 1 до 41 записані такі параметри мережевого з'єднання як тривалість, тип протоколу, мережевий сервіс, кількість отриманих байтів, кількість переданих байтів, статус з'єднання і т.д. В 42 полі записана інформація, що характеризує стан захищеності ІС – або відсутність атаки (normal), або її тип. У базі представлено 22 види атаки, які розділяються на 4 основних класи – відмова в обслуговуванні (DoS), несанкціоноване отримання прав доступу незареєстрованим користувачем (R2L), несанкціоноване підвищення привілеїв (U2R), зареєстрованим користувачем та сканування портів (Probe).

Оскільки основною передумовою застосування експертних знань в НМ є недостатня повнота навчальних даних, то основну увагу було зосереджено на розпізнаванні атак U2R, для яких кількість записів в KDD-99 найменша. Типовий приклад запису для атаки виду buffer_overflow– 113, tcp, telnet, SF, 6274, 16771, 0, 0, 0, 5, 0, 1, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 1, 1, 1.00, 0.00, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, buffer_overflow, для атаки виду loadmodule– 31, tcp, telnet, SF, 142, 1278, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 5, 3, 0.60, 0.60, 0.20, 0.00, 0.00, 0.00, 0.00, 0.00, loadmodule; для атаки виду perl –54, tcp, telnet, SF, 260, 2635, 0, 0, 0, 0, 0, 1, 0, 1, 0, 2, 2, 1, 0, 0, 0, 0, 1, 1, 0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 255, 1, 0.00, 0.01, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, perl і для атаки виду rootkit – 0, udp, other, SF, 32, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00,

255, 1, 0.00, 0.02, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, rootkit. З залученням експертів в галузі захисту інформації розроблено 10 продукційних правил для розпізнавання атак вказаного типу. При цьому правила стосуються розпізнавання neptune, smurf, pod, teardrop, land, back, buffer_overflow, loadmodule, perl, rootkit.

Приклад 1. Продукційне правило для розпізнавання кібератаки neptune:

Якщо тривалість з'єднання (duration) = 0 ∧ протокол (protocol_type) – tcp ∧ сервіс (service)– (private ∨ mnet ∨ sql_net) ∧ flag – (SF ∨ S0) ∧ кількість отриманих байт (src_bytes) – 0 ∧ кількість переданих байт (dst_bytes) – 0 ∧ land – 0 ∧ wrong_fragment – 0 ∧ urgent – 0 ∧ hot – 0 ∧ num_failed_logins – 0 ∧ logged_in = 0 ∧ num_compromised = 0 ∧ root_shell – 0 ∧ su_attempted= 0 ∧ num_root = 0 ∧ num_file_creations = 0 ∧ num_shells = 0 ∧ num_access_files = 0 ∧ num_outbound_cmds = 0 ∧ is_host_login= 0 ∧ is_guest_login = 0 ∧ count= від 6 до 141 ∧ srv_count = від 1 до 19 ∧ serror_rate = 1 ∧ srv_serror_rate = 1 ∧ rerror_rate = 0 ∧ srv_rerror_rate= 0 ∧ same_srv_rate = від 0.01 до 1 ∧ diff_srv_rate = від 0.05 до 0.33 ∧ srv_diff_host_rate = 0 ∧ dst_host_count= від 1 до 131 ∧ dst_host_srv_count = від 1 до 19 ∧ dst_host_same_srv_rate = від 0.01 до 1 ∧ dst_host_diff_srv_rate = від 0 до 0.27 ∧ dst_host_same_src_port_rate =від 0.01 до 1 ∧ dst_host_srv_diff_host_rate = від 0 до 0.33 ∧ dst_host_serror_rate = 1 ∧ dst_host_srv_serror_rate = 1 ∧ dst_host_rerror_rate = 0 ∧ dst_host_srv_rerror_rate = 0.

Приклад 2. Продукційне правило для розпізнавання кібератаки smurf:

Якщо тривалість з'єднання (duration) = 0 ∧ протокол (protocol_type) – imcp ∧ сервіс (service)– ecr_i ∧ flag – SF ∧ кількість отриманих байт (src_bytes) – 1032 ∧ кількість переданих байт (dst_bytes) – 0 ∧ land – 0 ∧ wrong_fragment – 0 ∧ urgent – 0 ∧ hot – 0 ∧ num_failed_logins – 0 ∧ logged_in = 0 ∧ num_compromised = 0 ∧ root_shell – 0 ∧ su_attempted = 0 ∧ num_root = 0 ∧ num_file_creations = 0 ∧

num_shells = 0 ∧ num_access_files = 0 ∧ num_outbound_cmds = 0 ∧ is_host_login = 0 ∧ is_guest_login = 0 ∧ count = від 300 до 500 ∧ srv_count = від 300 до 500 ∧ serror_rate = 0 ∧ srv_serror_rate = 0 ∧ rerror_rate = 0 ∧ srv_rerror_rate = 0 ∧ same_srv_rate = 1 ∧ diff_srv_rate = 0 ∧ srv_diff_host_rate = 0 ∧ dst_host_count = 255 ∧ dst_host_srv_count = 255 ∧ dst_host_same_srv_rate = 1 ∧ dst_host_diff_srv_rate = 0 ∧ dst_host_same_src_port_rate 1 ∧ dst_host_srv_diff_host_rate 0 ∧ dst_host_serror_rate = 0 ∧ dst_host_srv_serror_rate = 0 ∧ dst_host_rerror_rate = 0 ∧ dst_host_srv_rerror_rate = 0.

Приклад 3. Продукційне правило для розпізнавання кібератаки pod:

Якщо тривалість з'єднання (duration) = 0 ∧ протокол (protocol_type) – imcp ∧ сервіс (service)– ecr_i ∧ flag – SF ∧ кількість отриманих байт (src_bytes) – 1480 ∧ кількість переданих байт (dst_bytes) – 0 ∧ land – 0 ∧ wrong_fragment – 1 ∧ urgent – 0 ∧ hot – 0 ∧ num_failed_logins – 0 ∧ logged_in = 0 ∧ num_compromised = 0 ∧ root_shell – 0 ∧ su_attempted = 0 ∧ num_root = 0 ∧ num_file_creations = 0 ∧ num_shells = 0 ∧ num_access_files = 0 ∧ num_outbound_cmds = 0 ∧ is_host_login = 0 ∧ is_guest_login = 0 count = від 300 до 500 ∧ srv_count = від 300 до 500 ∧ serror_rate = 0 ∧ srv_serror_rate = 0 ∧ rerror_rate = 0 ∧ srv_rerror_rate = 0 ∧ same_srv_rate = 1 ∧ diff_srv_rate = 0 ∧ srv_diff_host_rate = 0 ∧ dst_host_count = від 80 до 225 ∧ dst_host_srv_count = від 1 до 20 ∧ dst_host_same_srv_rate = від 0.01 до 1 ∧ dst_host_diff_srv_rate = від 0.01 до 0.05 ∧ dst_host_same_src_port_rate = 0 ∧ dst_host_srv_diff_host_rate = 0 ∧ dst_host_serror_rate = 0 ∧ dst_host_srv_serror_rate = 0 ∧ dst_host_rerror_rate = 0 ∧ dst_host_srv_rerror_rate = 0.

Приклад 4. Продукційне правило для розпізнавання кібератаки teardrop:

Якщо тривалість з'єднання (duration) = 0 ∧ протокол (protocol_type) – udp ∧ сервіс (service) – private ∧ flag – SF ∧ кількість отриманих байт (src_bytes) – 28 ∧

кількість переданих байт (dst_bytes) – від 0 до 28 ∧ land – 0 ∧ wrong_fragment – від 1 до 3 ∧ urgent – 0 ∧ hot – 0 ∧ num_failed_logins – 0 ∧ logged_in = 0 ∧ num_compromised = 0 ∧ root_shell – 0 ∧ su_attempted = 0 ∧ num_root = 0 ∧ num_file_creations = 0 ∧ num_shells = 0 ∧ num_access_files = 0 ∧ num_outbound_cmds = 0 ∧ is_host_login = 0 ∧ is_guest_login = 0 count = від 300 до 500 ∧ srv_count = від 300 до 500 ∧ serror_rate = 0 ∧ srv_serror_rate = 0 ∧ rerror_rate = 0 ∧ srv_rerror_rate = 0 ∧ same_srv_rate = 1 ∧ diff_srv_rate = 0 ∧ srv_diff_host_rate = 0 ∧ dst_host_count = від 74 до 250 ∧ dst_host_srv_count = від 1 до 100 ∧ dst_host_same_srv_rate = від 0.01 до 0.5 ∧ dst_host_diff_srv_rate = від 0.1 до 0.5 ∧ dst_host_same_src_port_rate = від 0 до 1 dst_host_srv_diff_host_rate = 0 ∧ dst_host_serror_rate = 0 ∧ dst_host_srv_serror_rate = 0 ∧ dst_host_rerror_rate = 0 ∧ dst_host_srv_rerror_rate = 0.

Приклад 5. Продукційне правило для розпізнавання кібератаки land:

Якщо тривалість з’єднання (duration) = 0 ∧ протокол (protocol_type) – tcp ∧ сервіс (service)– finger ∧ flag – S0 ∧ кількість отриманих байт (src_bytes) – 0 ∧ кількість переданих байт (dst_bytes) – 0 ∧ land = 1 ∧ wrong_fragment – 0 ∧ urgent – 0 ∧ hot – 0 ∧ num_failed_logins – 0 ∧ logged_in = 0 ∧ num_compromised = 0 ∧ root_shell – 0 ∧ su_attempted = 0 ∧ num_root = 0 ∧ num_file_creations = 0 ∧ num_shells = 0 ∧ num_access_files = 0 ∧ num_outbound_cmds = 0 ∧ is_host_login = 0 ∧ is_guest_login = 0 ∧ count = від 1 до 78 ∧ srv_count = від 1 до 2 ∧ serror_rate = 1 ∧ srv_serror_rate = 1 ∧ rerror_rate = від 0 до 0.67 ∧ srv_rerror_rate = 0 ∧ same_srv_rate = від 0.01 до 1 ∧ diff_srv_rate = від 0 до 1 ∧ srv_diff_host_rate = від 0 до 1 ∧ dst_host_count = від 1 до 16 ∧ dst_host_srv_count = від 1 до 9 ∧ dst_host_same_srv_rate = від 0.07 до 1 ∧ dst_host_diff_srv_rate = від 0 до 0.2 ∧ dst_host_same_src_port_rate = від 0.08 до 1 ∧ dst_host_srv_diff_host_rate = від 0 до 1 ∧ dst_host_serror_rate = від 0.11 до 1 ∧ dst_host_srv_serror_rate = від 0 до 0.07 ∧ dst_host_rerror_rate = 0 ∧ dst_host_srv_rerror_rate = 0.

Приклад 6. Продукційне правило для розпізнавання кібератаки back:

Якщо тривалість з'єднання (duration) = 0 ∧ протокол (protocol_type) – tcp ∧ сервіс (service)– http ∧ flag – (SF ∨ RSTR) ∧ кількість отриманих байт (src_bytes) – 54540 ∧ кількість переданих байт (dst_bytes) – від 7000 до 8500 ∧ land – 0 ∧ wrong_fragment – 0 ∧ urgent – 0 ∧ hot – від 1 до 2 ∧ num_failed_logins – 0 ∧ logged_in = 1 ∧ num_compromised = від 0 до 1 ∧ root_shell – 0 ∧ su_attempted = 0 ∧ num_root = 0 ∧ num_file_creations = 0 ∧ num_shells = 0 ∧ num_access_files = 0 ∧ num_outbound_cmds = 0 ∧ is_host_login = 0 ∧ is_guest_login = 0 ∧ count = від 2 до 5 ∧ srv_count = від 2 до 7 ∧ serror_rate = 0 ∧ srv_serror_rate = 0 ∧ rerror_rate = 0 ∧ srv_rerror_rate = від 0 до 0.4 ∧ same_srv_rate = 1 ∧ diff_srv_rate = 0 ∧ srv_diff_host_rate = від 0 до 0.7 ∧ dst_host_count = від 2 до 128 ∧ dst_host_srv_count = від 2 до 128 ∧ dst_host_same_srv_rate = 1 ∧ dst_host_diff_srv_rate = 0 ∧ dst_host_same_src_port_rate = від 0.03 до 0.5 ∧ dst_host_srv_diff_host_rate = 0 ∧ dst_host_serror_rate = 0 ∧ dst_host_srv_serror_rate = 0 ∧ dst_host_rerror_rate = від 0 до 0.2 ∧ dst_host_srv_rerror_rate = від 0 до 0.2.

Приклад 7. Продукційне правило для розпізнавання кібератаки buffer_overflow:

Якщо тривалість з'єднання (duration) = 0 ∧ протокол (protocol_type) – tcp ∧ сервіс (service)– ftp_data ∧ flag – SF ∧ кількість отриманих байт (src_bytes) – 0 ∧ кількість переданих байт (dst_bytes) – від 2000 до 6000 ∧ land – 0 ∧ wrong_fragment = 0 ∧ urgent = 0 ∧ hot = 0 ∧ num_failed_logins = 0 ∧ logged_in = 1 ∧ num_compromised = 0 ∧ root_shell – від 0 до 1 ∧ su_attempted = 0 ∧ num_root = від 0 до 1 ∧ num_file_creations = 0 ∧ num_shells = 0 ∧ num_access_files = 0 ∧ num_outbound_cmds = 0 ∧ is_host_login = 0 ∧ is_guest_login = від 1 до 3 ∧ count = від 1 до 3 ∧ srv_count = 0 ∧ serror_rate = 0 ∧ srv_serror_rate = 0 ∧ rerror_rate = 0 ∧ srv_rerror_rate = 1.00 ∧ same_srv_rate = 0 ∧ diff_srv_rate = 0 ∧ srv_diff_host_rate = від 1 до 4 ∧dst_host_count = від 1 до 84 ∧dst_host_srv_count = 1.00 ∧ dst_host_same_srv_rate = 0.00 idst_host_diff_srv_rate = 0.00

∧dst_host_same_src_port_rate = 1.00 ∧dst_host_srv_diff_host_rate = 0.02 ∧dst_host_serror_rate = 0 ∧dst_host_srv_serror_rate = 0 ∧dst_host_rerror_rate = 0 ∧dst_host_srv_rerror_rate = 0.

Приклад 8. Продукційне правило для розпізнавання кібератаки loadmodule:

Якщо тривалість з'єднання (duration) = від 0 до 103 ∧ протокол (protocol_type) – tcp ∧ сервіс (service)– (telnet ˅ ftp ˅ ftp_data) ∧ flag – SF ∧ кількість отриманих байт (src_bytes) – від 0 до 302 ∧ кількість переданих байт (dst_bytes) – від 600 до 9000 ∧ land – 0 ∧ wrong_fragment – 0 ∧ urgent – 0 ∧ hot – від 1 до 4 ∧ num_failed_logins – 0 ∧ logged_in = від 0 до 1 ∧ num_compromised = від 0 до 4 ∧ root_shell – від 0 до 1 ∧ su_attempted = 0 ∧ num_root = від 0 до 3 ∧ num_file_creations = від 0 до 4 ∧ num_shells = від 0 до 2 ∧ num_access_files = від 0 до 1 ∧ num_outbound_cmds = 0 ∧ is_host_login = 0 ∧ is_guest_login = 0 ∧ count = від 1 до 4 ∧ srv_count = від 1 до 3 ∧ serror_rate = 0 ∧ srv_serror_rate = 0 ∧ rerror_rate = 0 ∧ srv_rerror_rate = 0 ∧ same_srv_rate = від 0.5 до 1 ∧ diff_srv_rate = від 0 до 1 ∧ srv_diff_host_rate = 0 ∧ dst_host_count = від 1 до 6 ∧ dst_host_srv_count = від 1 до 5 ∧ dst_host_same_srv_rate = від 0.25 до 1 ∧ dst_host_diff_srv_rate = від 0 до 0.75 ∧ dst_host_same_src_port_rate = від 0.17 до 1 ∧ dst_host_srv_diff_host_rate = від 0 до 0.67 ∧ dst_host_serror_rate = 0 ∧ dst_host_srv_serror_rate = 0 ∧ dst_host_rerror_rate = 0 ∧ dst_host_srv_rerror_rate = 0.

Приклад 9. Продукційне правило для розпізнавання кібератаки perl:

Якщо тривалість з'єднання (duration) = від 45 до 54 ∧ протокол (protocol_type) – tcp ∧ сервіс (service)– telnet ∧ flag – SF ∧ кількість отриманих байт (src_bytes) – 260 ∧ кількість переданих байт (dst_bytes) – від 2300 до 2600 ∧ land – 0 ∧ wrong_fragment – 0 ∧ urgent – 0 ∧ hot – 0 ∧ num_failed_logins – 0 ∧ logged_in = 1 ∧ num_compromised = 0 ∧ root_shell = 1 ∧ su_attempted = 0 ∧

num_root = 2 ∧ num_file_creations = 2 ∧ num_shells = 1 ∧ num_access_files = 0 ∧ num_outbound_cmds = 0 ∧ is_host_login = 0 ∧ is_guest_login = 0 ∧ count = 1 ∧ srv_count = 1 ∧ serror_rate = 0 ∧ srv_serror_rate = 0 ∧ rerror_rate = 0 ∧ srv_rerror_rate = 0 ∧ same_srv_rate = 1 ∧ diff_srv_rate = 0 ∧ srv_diff_host_rate = 0 ∧ dst_host_count = 255 ∧ dst_host_srv_count = від 1 до 2 ∧ dst_host_same_srv_rate = 0 ∧ dst_host_diff_srv_rate = 0.01 ∧ dst_host_same_src_port_rate = 0 ∧ dst_host_srv_diff_host_rate = 0 ∧ dst_host_serror_rate = 0 ∧ dst_host_srv_serror_rate = 0 ∧ dst_host_rerror_rate = 0 ∧ dst_host_srv_rerror_rate = 0.

Приклад 10. Продукційне правило для розпізнавання кібератаки rootkit:

Якщо тривалість з'єднання (duration) = від 0 до 700 ∧ протокол (protocol_type) – (tcp ˅ udp) ∧ сервіс (service)– (telnet ˅ ftp ˅ ftp_data) ∧ flag – SF ∧ кількість отриманих байт (src_bytes) – від 0 до 1700 ∧ кількість переданих байт (dst_bytes) – від 0 до 24000 ∧ land – 0 ∧ wrong_fragment – 0 ∧ urgent – від 0 до 1 ∧ hot – від 0 до 1 ∧ num_failed_logins – 0 ∧ logged_in = від 0 до 1 ∧ num_compromised = від 0 до 6 ∧ root_shell – від 0 до 1 ∧ su_attempted = від 0 до 14 ∧ num_root = від 0 до 1 ∧ num_file_creations = 0 ∧ num_shells = 0 ∧ num_access_files = 0 ∧ num_outbound_cmds = 0 ∧ is_host_login = 0 ∧ is_guest_login = 1 ∧ count = 1 ∧ srv_count = 0 ∧ serror_rate = 0 ∧ srv_serror_rate = 0 ∧ rerror_rate = 0 ∧ srv_rerror_rate = 1 ∧ same_srv_rate = 0 ∧ diff_srv_rate = 0 ∧ srv_diff_host_rate = від 1 до 255 ∧ dst_host_count = від 1 до 41 ∧ dst_host_srv_count = від 0.01 до 1 ∧ dst_host_same_srv_rate = 0.02 ∧ dst_host_diff_srv_rate = від 0 до 1 ∧ dst_host_same_src_port_rate = 0 ∧ dst_host_srv_diff_host_rate = 0 ∧ dst_host_serror_rate = 0 ∧ dst_host_srv_serror_rate = 0 ∧ dst_host_rerror_rate = 0 ∧ dst_host_srv_rerror_rate = 0.

Приклади продукційних правил для розпізвання всіх підтипів кібератак можна знайти в Додатку 3.

Використавши вказані продукційні правила та наведений вище метод, побудовано модифіковану мережу PNN призначену для виявлення атак типу U2R. Основні параметри мережі такі: кількість вхідних параметрів мережі К = 41, кількість нейронів ШД дорівнює 2 (нейрон А відповідає атаці, нейрон В – нормальному стану), кількість нейронів ШО дорівнює 28, а кількість нейронів ШФ дорівнює 1148. Апробація розробленої нейромережевої моделі на даних KDD-99 показала абсолютну точність розпізнавання всіх видів атак класу U2R, що відповідає загальновизнаному твердженню – СВА які базуються на використанні експертних знань безпомилково розпізнають відомі атаки. Для порівняльного аналізу запропонованого методу використано роботи [3-5], в яких наведено результати застосування різноманітних методів розпізнавання мережевих атак на сигнатурах представлених в базі даних KDD-99. Так в роботах [4, 5] для розпізнавання атак використано багатошаровий персептрон і мережу Кохонена. В роботі [4] показано, що точність розпізнавання атак класу U2R мережею Кохонена становить: для buffer_overflow – 0.0458, для loadmodule – 0.0208, для perl – 0.2857, а для rootkit – 0.0063. При цьому багатошаровий персептрон, по причині малого обсягу навчальних даних, взагалі не вдалось навчити розпізнавати атаки типу U2R. В роботі [5] наведено дещо інші дані. Точність розпізнавання атак класу U2R мережею Кохонена становить близько 0.21. Також в роботах [3, 5] для розпізнавання застосовано спеціальну адаптивну модель, яка базується на статистичному аналізі головних компонент. Точність розпізнавання цієї моделі не перевищує 0.5, що пояснюється не великою кількістю статистичних даних. Порівняння результатів [3-5] з результатами представленої роботи вказує на те, що запропонований метод дозволить розширити повноту класифікації атак, сигнатури яких не достатньо представлені базах даних.

3.4. Висновки до третього розділу

Третій розділ присвячено розробці продукційних правил на основі експертних знань. Розглянуто мову програмування та методи які використовувалися в процесі розробки нейронної мережі. Доведено, що мова програмування Python є найбільш зручною для такої розробки, т.як має величезну базу бібліотек, які полегшують процес алгоритмізації моделі PNN.

Експериментально доведено, що мережеві кібератаки типу DoS(підтипи: land, pod), U2R(підтипи: buffer_overflow, loadmodule, perl, rootkit) та R2L(підтипи: guess_passwd, ftp_write, imap, phf, multihop, wazermaster, wazerclient, spy) мають високу ймовірність помилки розпізнавання, т.як в базі даних KDD-99 мають недостатню кількість прикладів. Для розпізнавання таких видів кібератак рекомендується використовувати іншу базу даних. Навідміну від кібератак типу DoS(підтипи: neptune, smurf, teardrop, back) та Probe(підтипи: portsweep, ipsweep, satan, nmap) мають досить велику кількість навчальних прикладів для точного розпізнавання кібератаки таких типів.

ВИСНОВОК

Пи написані даної дипломної роботи було розглянуто задачу розробки методики розпізнавання мережевих кібератак за допомогою нейронних мереж. В процесі вирішення отримано наступні результати:

1. Показано, що важливим недоліком сучасних систем захисту являється недостатня ефективність систем розпізнавання мережевих кібератак, підвищити яку можливо за рахунок використання методів теорії штучних нейронних мереж.

2. Розроблена методика застосування нейронних мереж в задачах розпізнаваня мережевих кібератак. Методика дозволила визначити, що для розпізнавання мережевих кібератак доцільно застосувати нейромережеві архітектури типу PNN.

3. Розроблена методика визначення вхідних та вихідних параметрів НМ типу PNN при їх використанні в системах розпізнавання мережевих кібератак написаних на мові програмування Python 2.7. З використанням визначеної номенклатури вхідних та вихідних параметрів побудовані моделі PNN.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Cannady J. Artificial Neural Networks for Misuse Detection. − In Proceedings of the 21st National Information Systems Security Conference. − 1998, October.
2. Haykin S. Neural networks. − A comprehensive foundations. McMillan College Publ.Co. N.Y., − 1994. 696 pp.
3. Архипов А, Ишутин А. Применение моделей обнаружения аномалий для выявления атак // Четверта науково-технічна конференція. Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. Тези доповідей. − 2006. − 71-72 с.
4. Барский А.Б. Нейронные сети: распознавание, управление, принятие решений. − М.: Финансы и статистика, 2004. – 176 с.
5. Галушкин А.И. Теория нейронных сетей − М.: ИПРЖР, 2000. – 416 с.
6. Вороновский Г.К., Махотило К.В., Петрашев С.Н., Сергеев С.А. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальності. − Харьков: Основа, 1997. – 112 с.
7. Головко В. А. Нейронные сети: обучение, организация и применение. – М.: ИПРЖР, 2001. – 256 с.
8. Дорогов А.Ю., Алексеев А.А. Нейронные сети с ядерной организацией. // Оборонная техника. – 1998. – №7-8. – 43-46 с.
9. Ежов А.А., Шумский С.А. Нейрокомпьютинг и его применения в экономике и бизнесе. – М.: МИФИ, 1998. – 224 с.
10. Заенцев И.В. Нейронные сети: основные модели. - Воронеж: Воронежский государственный университет, 1999. – 76 с.
11. Зиновьев А.Ю. Визуализация многомерных данных. – М.: СК Пресс, 2005. – 180 с.
12. Каллан Р. Основные концепции нейронных сетей. – М.: Вильямс, 2003. – 288 с.
13. Круглов В.В., Борисов В.В. Искусственные нейронные сети. – М.: Горячая линия-Телеком, 2002. – 382 с.

14. Кузнецов Г.В., Иванов А.М. Классификация и анализ систем и методов обнаружения атак. // Захист інформації. – 2004, №4 – 4-11 с.
15. Кузнецов Г.В., Иванов А.М. Методы анализа даннях для обнаружения атак в компьтерных сетях банковских структур // Защита информации. Сб.н.тр. - К.: НАУ. – 2004. – 45-50 с.
16. Лукацкий А.В. Обнаружение атак. – СПб.: БХВ-Петербург, 2003. – 624 с.
17. Огарок А., Комашинский Д., Школьников Д. Виртуальные войны. Искусственный интеллект на защите от вирусов и программных закладок // Конфідент – 2003. – №2 (50). – 64-69, 97 с.
18. Абрамов Е. С. Разработка и исследование методов построения систем обнаружения атак: дис.канд. те хн. наук: 05.13.19 / Абрамов Е. С. Таганрог, 2005. 199 с.
19. Артеменко А.В., Головко В. А. Анализ нейросетевых методов распознавания компьютерных вирусов /Материалы секционных заседаний. Молодежный инновационный форум «ИНТРИ» – 2010. — Минск: ГУ «БелИСА», 2010. – 239 с.
20. Гамаюнов Д. Ю. Обнаружение компьютерных атак на основе анализа поведения сетевых объектов: авторефер. дисс. на соискание научн. степени канд. техн. наук: спец 05.13.11 – математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей / Д.Ю. Гамаюнов – Москва, 2007. – 11 с.
21. Ежов А. А. Нейрокомпьютинг и его применения в экономике и бизнесе / А. А. Ежов, С. А. Шумский. М.: МИФИ, 1998. 224 с.
22. Гірницька Д. А. Визначення коефіцієнтів важливості для експертного оцінювання у галузі інформаційної безпеки / Д.А. Горніцька, В.В. Волянська, А.О. Корченко // Захист інформації. – 2012. – Том 14, №1 (54). – С. 108-121.
23. Емельянова Ю. Г. Нейросетевая технология обнаружения сетевых атак на информационные ресурсы / Ю. Г. Емельянова, А. А. Талалаев, И. П.

Тищенко, В. П. Фраленко // Программные системы: теория и приложен ия. – 2011. – №3(7). – 3–15 с.

24. Корченко А. А. Модель эвристических правил на логико-лингвистически х связках для обнаружения аномалий в компьютерных системах / А. А. Корченко // Захист інформації – 2012. – № 4. – 109-115 с.
25. Руденко О.Г. Штучні нейронні мережі. Навч. посіб. / О.Г. Руденко, Є.В. Бодянський. – Харків: ТОВ "Компанія СМІТ", 2006. – 404 с.
26. Терейковський І. Нейронні мережі в засобах захисту комп'ютерної інформації / І. Терейковський. К.: ПоліграфКонсалтинг. 2007. – 209 с.
27. KDD cup 99 Intrusion detection data set [Електронний ресурс]. Електрон. текстові дані (752 Мб). – Darpa: Irvine, CA 92697-3425, 1999. – Режим доступу: http://kdd.ics.uci.edu/databases/kddcup99.
28. Брюховецкий А.А. Обнаружение вредоносных программ на основе инфрмативных признаков сетевого трафика / А.А. Брюховецкий, А.В. Скатков // Тези доповідей міжнародної конференція з автоматичного управління, присвячена 100-річчю з дня народження академіка О.Г. Івахненка.
29. Терейковський І.А. Вдосконалення алгоритму навчання багатошарового персептрону призначеного длярозпізнавання мережевих атак / І.А. Терейковський // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні − 2012. — Вип. 2(24). — С. 65−70.
30. Bezobrazov S., Golovko V. Neural Networks for Artificial Immune Systems: LVQ for Detectors Construction // IDAACS'2007: proceedings of the 4 IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. — Dortmund, 2010. — P. 180-184.
31. Айвенс К. Компьютерные сети / Айвенс К.; пер. с. англ. – СПб.: Питер, 2006. - 304 с.
32. Безруков Н. Н. Компьютерная вирусология / Н. Н. Безруков. – К.: Инкомбук, 1990. – 450 с.

33. Вилков А.С. Информационная безопасность персональных ЭВМ и мониторинг компьютерных сетей / А.С. Вилков. – М. : МИНИТ ФСБ России, 2005. – 210 с.
34. Гарнаев А. Ю. Microsoft Office 2000 / А. Ю. Гарнаев. – СПб. : БХВ, 2000. – 656 с.