Міністерство освіти і науки України
Національний авіаційний університет

На правах рукопису

Терейковський Ігор Анатолійович

УДК 004.056.5:004.8(043.5)

НЕЙРОМЕРЕЖЕВІ МОДЕЛІ, МЕТОДИ І ЗАСОБИ ОЦІНЮВАННЯ ПАРАМЕТРІВ БЕЗПЕКИ ІНТЕРНЕТ-ОРІЄНТОВАНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

05.13.21 – системи захисту інформації

Дисертація
на здобуття наукового ступеня
доктора технічних наук

Науковий консультант
доктор технічних наук, професор
Корченко Олександр Григорович

Київ 2015

СПИСОК УМОВНИХ СКОРОЧЕНЬ

АНМ – асоціативна нейронна мережа

АРТ – нейронна мережа адаптивної резонансної теорії

БШП – багатошаровий персептрону

ВШ – вхідний шар нейронів

ДШП – двохшаровий персептрон

ІС – інформаційна система

ЛМ – ланцюг Маркова

НК – неочікувана кібератака

НМ – нейронна мережа

НММ – нейромережева модель

НМС – нейромережева система

ПБ – параметр безпеки

ПК – поступова кібератака

РБФ – нейронна мережа з радіальними базисними функціями

СВА – система виявлення атак

СЗІ – система захисту інформації

СНМ – семантична нейронна мережа

СШН – схований шар нейронів

ТК – топографічна карта Кохонена

ША – шаблон атаки

ШВ – вихідний шар нейрнів

ШД – шар додавання

ШНП – шаблон нормальної поведінки

ШО – шар образів

ШП – шаблон поведінки

ШПЗ – шкідливе програмне забезпечення

ШФ – шар фільтрації

ЗМІСТ

СПИСОК УМОВНИХ СКОРОЧЕНЬ 2
ВСТУП 6
1. АНАЛІЗ НЕЙРОМЕРЕЖЕВИХ ЗАСОБІВ ОЦІНЮВАННЯ ПАРАМЕТРІВ БЕЗПЕКИ ІНТЕРНЕТ-ОРІЄНТОВАНИХ ІНФОРМАЦІЙНИХ СИСТЕМ 16
1.1. Проблема оцінювання параметрів безпеки інформаційних систем 16
1.2. Актуальні задачі оцінювання параметрів безпеки Інтернет-орієнтованих інформаційних систем 23
1.3. Дослідження можливостей застосування нейронних мереж для оцінювання параметрів безпеки 31
1.4. Нейромережеві моделі та методи оцінювання параметрів безпеки інформаційних систем 41
1.5. Висновки до першого розділу 54
2. РОЗВИТОК ТЕОРЕТИЧНИХ ПОЛОЖЕНЬ ПОБУДОВИ НЕЙРОМЕРЕЖЕВИХ ЗАСОБІВ ОЦІНЮВАННЯ ПАРАМЕТРІВ БЕЗПЕКИ ІНТЕРНЕТ-ОРІЄНТОВАНИХ ІНФОРМАЦІЙНИХ СИСТЕМ 57
2.1. Базові підходи до оцінювання параметрів безпеки за допомогою нейромережевих засобів 57
2.2. Критерії оптимізації виду нейромережевої моделі 70
2.3. Вдосконалення математичного забезпечення процесу навчання багатошарового перспетрону 75
2.4. Верифікація нейромережевих моделей оцінювання параметрів безпеки 87
2.5. Висновки до другого розділу 91
3. МОДЕЛІ ОЦІНЮВАННЯ ПАРАМЕТРІВ БЕЗПЕКИ 94

ІНТЕРНЕТ-ОРІЄНТОВАНИХ ІНФОРМАЦІЙНИХ СИСТЕМ
3.1. Модель процесів інтеграції параметрів безпеки, що використовуються нейромережевими засобами розпізнавання кібератак 94
3.2. Марківська модель одноперіодичного шаблону поведінки 106
3.3. Марківська модель багатоперіодичного шаблону поведінки 108
3.4. Модель на основі багатошарового перспетрону 111
3.5. Модель мережі MPNN 127
3.6. Модель створення ефективних нейромережевих засобів оцінювання параметрів безпеки 131
3.7. Висновки до третього розділу 137
4. МЕТОДИ ПОБУДОВИ НЕЙРОМЕРЕЖЕВИХ ЗАСОБІВ ОЦІНЮВАННЯ ПАРАМЕТРІВ БЕЗПЕКИ ІНТЕРНЕТ-ОРІЄНТОВАНИХ ІНФОРМАЦІЙНИХ СИСТЕМ 140
4.1. Метод застосування продукційних правил для подання експертних знань 140
4.2. Метод визначення часових характеристик використання нейромережевих засобів 150
4.3. Метод проектування шаблону поведінки параметрів безпеки 161
4.4. Метод визначення ефективності розробки нейромережевих засобів оцінювання параметрів безпеки 171
4.5. Висновки до четвертого розділу 176
5. НЕЙРОМЕРЕЖЕВІ СИСТЕМИ ОЦІНЮВАННЯ ПАРАМЕТРІВ БЕЗПЕКИ ІНТЕРНЕТ-ОРІЄНТОВАНИХ ІНФОРМАЦІЙНИХ СИСТЕМ 178
5.1. Комплексна методологія нейромережевого оцінювання параметрів безпеки інформаційних систем 178
5.2. Система оцінювання параметрів безпеки для розпізнавання кібератак на ресурси Інтернет-орієнтованих інформаційних систем 189

5.3. Система розпізнавання шкідливого програмного забезпечення та класифікації листів електронної пошти 196
5.4. Система розпізнавання мережевих кібератак 228
5.5. Висновки до п’ятого розділу 241
ВИСНОВКИ 244
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 249
ДОДАТОК А. Опис Windows-застосунку для розпізнавання скриптового шкідливого програмного забезпечення 280
ДОДАТОК Б. Акти впроваджень результатів дисертаційного дослідження 308

ВСТУП

Актуальність теми. Сучасний стан розвитку вітчизняних ІС, інтегрованих до глобальної мережі Інтернет, характеризується підвищеним рівнем вимог до безпеки інформації, який вже складно забезпечити за допомогою систем захисту, в підсистемах контролю та управління яких використовуються виключно класичні методи оцінки ПБ. Разом з тим, в різних галузях науки та техніки проявляється інтерес до використання методів та моделей теорії НМ. Популярність НМ можна пояснити доведеною ефективністю їх застосування в задачах класифікації та кластеризації образів, апроксимації функцій, прогнозування, оптимізації, управління, створення інформаційно-обчислювальних систем з асоціативною пам'яттю, які частково або в комплексі доводиться вирішувати при оцінюванні ПБ для виявлення кібератак. На сьогодні відомі спроби використання НМ в різноманітних комерційних та вільнодоступних СЗІ. Так, НМ використовуються для розпізнавання кібератак в міжмережевих екранах компанії Cisco та для розпізнавання вірусів в антивірусах Norton Antivirus виробництва корпорації Symantec та F-Prot виробництва компанії CYREN GlobalView Security Lab. Також за допомогою НМ визначаються DDOS-атаки в вільнопоширюваному модулі, призначеному для інтегрування в програмний комплекс Snort. Крім того, компанією Facebook задекларовано використання нейромережевих засобів (НМЗ) розпізнавання спаму. Вказані засоби набули певного поширення в СЗІ вітчизняних ІС, однак високий рівень помилкових спрацювань, необхідність використання потужного апаратного забезпечення, тривалість та складність пристосування до нових видів кібератак та умов застосування значно обмежують їх практичну цінність. Крім того, недоліком поширених закордонних комерційних НМЗ розпізнавання кібератак являється висока вартість та відсутність детальної науково-технічної документації.

Питанням розробки нейромережевих моделей та методів

параметричного оцінювання стану ІС в різний час займалися такі вчені як Є. Бодянський, Д. Деннінг, О. Додонов, О. Корченко, А. Лукацький, О. Петров, О. Резнік, О. Руденко, С. Форестер, В. Харченко, В. Хорошко та ін. Однак в галузі захисту інформації побудову таких моделей та методів засновано на різнорідних підходах, вони носять точковий характер застосування та практично не взаємопов'язані, що ускладнює їх використання при створенні ефективних систем розпізнавання кібератак.

Таким чином, посилення вимог до ефективності систем розпізнавання кібератак на ресурси Інтернет-орієнтованих інформаційних систем, перспективність використання нейромережевих засобів оцінювання параметрів безпеки для розпізнавання кібератак, малодоступність практичного аспекту захисту інформації для науково-критичного аналізу внаслідок широкого використання розробок рівня "ноу-хау", недостатня взаємопов'язаність відомих нейромережевих методів та засобів оцінювання параметрів безпеки, невідповідність їх характеристик до змін умов застосування, нових видів кібератак та можливості функціонування при обмежених обчислювальних ресурсах обумовлюють актуальність обраної науково-прикладної проблеми дисертаційного дослідження – створення комплексної методології розробки широкодоступних ефективних нейромережевих засобів оцінювання параметрів безпеки Інтернет-орієнтованих інформаційних систем, які за рахунок теоретично обґрунтованого вибору характеристик дозволяють оперативно розпізнавати нові види кібератак при обмежених обчислювальних ресурсах та варіативності умов застосування.

Зв'язок роботи з науковими програмами, планами, темами.. Одержані результати дисертаційної роботи виконані відповідно до планів наступних держбюджетних науково-дослідних робіт:

– Національного авіаційного університету по темі «Організація систем захисту інформації від кібератак» (№0111U000171);

– Національного технічного університету України «Київський

політехнічний інститут» по темі «Методи організації високоефективних спеціалізованих сховищ даних науково-освітнього призначення на основі кластерних обчислювальних технологій» (№0210U000261);

– Кіровоградського національного технічного університету по темі «Розробка методів підвищення оперативності передачі та захисту інформації у телекомунікаційних системах» (№0113U003086).

Мета і задачі дослідження.

Мета роботи полягає у підвищенні ефективності систем розпізнавання кібератак в Інтернет-орієнтованих інформаційних системах на основі створення комплексної методології розробки нейромережевих засобів оцінювання параметрів безпеки інформаційних систем, які за рахунок теоретично обґрунтованого вибору характеристик дозволяють оперативно розпізнавати нові види кібератак при обмежених обчислювальних ресурсах та варіативності умов застосування.

Для досягнення поставленої мети необхідно розв'язати наступні **задачі:**

– проаналізувати сучасні нейромережеві засоби оцінювання параметрів безпеки інформаційних систем;

– розвинути теоретичні положення побудови нейромережевих засобів оцінювання параметрів безпеки інформаційних систем, що дозволяють навчатись за допомогою експертних даних, зменшувати похибки класифікації, враховувати особливості сучасних видів кібератак, умови використання та верифікувати отримані рішення;

– побудувати моделі, що враховують запропоновані теоретичні рішення та використовуються в нейромережевих засобах оцінювання параметрів безпеки;

– розробити методи створення нейромережевих засобів оцінювання параметрів безпеки, що враховують запропоновані теоретичні рішення та побудовані моделі;

– розробити нейромережеві системи оцінювання параметрів безпеки інформаційних систем, які дозволяють розпізнавати шкідливе програмне

забезпечення, класифікувати листи електронної пошти та розпізнавати мережеві кібератаки.

Об'єктом дослідження – процеси оцінювання параметрів безпеки ресурсів Інтернет-орієнтованих інформаційних систем для розпізнавання кібератак.

Предметом дослідження – нейромережеві моделі, методи та засоби процесів оцінювання параметрів безпеки ресурсів Інтернет-орієнтованих інформаційних систем для розпізнавання кібератак.

Методи дослідження. Використано методи теорії захисту інформації, НМ, марківських процесів, прикладної статистики, оптимізації та комп'ютерного моделювання.

Наукова новизна отриманих результатів. У результаті проведених досліджень науково обгрунтовано комплексну методологію розробки широкодоступних ефективних нейромережевих засобів оцінювання параметрів безпеки ресурсів Інтернет-орієнтованих інформаційних систем, які дозволяють оперативно розпізнавати нові види кібератак при обмежених обчислювальних ресурсах та варіативності умов застосування.

При цьому отримано наступні нові наукові результати:

– отримали подальший розвиток теоретичні положення побудови нейромережевих засобів оцінювання параметрів безпеки, які за рахунок вперше розроблених підходів до розпізнавання поступових та неочікуваних кібератак, визначення оптимального виду нейромережевої моделі, доцільності застосування та ефективності розробки нейромережевих засобів, класифікації подібних кібератак, застосування продукційних правил для подання експертних знань, проведеній верифікації нейромережевих моделей, запропонованих критеріїв оцінки ефективності нейромережевих засобів, критеріїв вибору оптимального виду нейромережевої моделі та застосуванню розробленого функціоналу приведеної помилки навчання багатошарового персептрону дозволяють вдосконалювати нейромережеві засоби шляхом їх адаптації до поступових і неочікуваних кібератак, умов застосування, навчання за допомогою експертних даних, зменшувати похибки класифікації

та надають можливість верифікації отриманих рішень;

– отримали подальший розвиток моделі нейромережевих засобів оцінювання параметрів безпеки, які за рахунок застосування розроблених теоретичних положень побудови нейромережевих засобів**,** експертного оцінювання вагомості параметрів безпеки, введення в модель MPNN нейронного шару фільтрації з лінійною біполярною з насиченням функцією активації, розроблених аналітичних залежностей для розрахунку параметрів ланцюгів Маркова, призначених для прогнозування параметрів безпеки на стаціонарних інтервалах та для оцінки оптимальної кількості схованих нейронів, кількості обчислювальних навчальних операцій, обсягу пам’яті і помилки навчання багатошарового персептрону дозволяють: визначити перелік параметрів безпеки, які доцільно оцінювати нейромережевими засобами; створювати шаблони поведінки, адаптовані до складного характеру параметрів безпеки; зменшити ресурсоємність процесу визначення оптимальної структури багатошарового персептрону; апріорно оцінювати обчислювальні потужності, необхідні для реалізації нейромережевої моделі; за допомогою експертних даних навчати нейромережеву модель; формалізувати процес створення ефективних нейромережевих засобів, що є основою для підвищення ефективності методів їх розробки;

– вперше розроблено метод подання експертних знань для нейромережевих засобів оцінювання параметрів безпеки, в якому за рахунок розробленого математичного забезпечення детермінування параметрів статистично подібних кібератак, продукційних правил представлення навчальних прикладів та структури і вагових коефіцієнтів синаптичних зв’язків нейромережевої моделі типу MPNN, забезпечується оперативність розпізнавання та розширення множини видів кібератак, характеристики яких не представлені в статистичних даних;

– вперше розроблено метод визначення часових характеристик використання нейромережевих засобів, в якому завдяки розробленим аналітичним залежностям для визначення очікуваного терміну розробки,

допустимих термінів формування навчальної вибірки та навчання нейромережевої моделі, запропонованим співвідношенням між очікуваним і допустимим терміном розробки та очікуваним і допустимим терміном навчання, розробленій множині допустимих видів нейромережевих моделей отримана можливість визначення доцільності застосування нейромережевих засобів оцінювання параметрів безпеки для виявлення кібератак на заданий об’єкт захисту;

– вперше розроблено метод проектування шаблону поведінки, який використовується для навчання нейромережевих моделей, в якому за рахунок застосування багатоперіодичних рядів динаміки, розробленого математичного забезпечення для розрахунку періодичних складових та розробленої негомогенної марківської моделі забезпечується зменшення похибки шаблону, що є основою для зменшення терміну формування навчальної вибірки та зменшення похибок класифікації нейромережевих моделей при розпізнаванні поступових кібератак;

– вперше розроблено метод визначення ефективності розробки нейромережевих засобів оцінювання параметрів безпеки, який за рахунок застосування запропонованих критеріїв оцінки ефективності, що відображають ступінь виконання основних вимог до побудови та застосування нейромережевих засобів, запропонованих вагових коефіцієнтів важливості критеріїв ефективності та розробленого генерального критерію ефективності нейромережевих засобів дозволяє, відповідно до визначених критеріїв, обрати найбільш ефективний засіб;

– вперше розроблено комплексну методологію нейромережевої оцінювання параметрів безпеки, яка за рахунок взаємопов’язаного використання розроблених підходів до верифікації нейромережевих засобів, визначення оптимального виду нейромережевої моделі, розроблених моделей створення ефективних нейромережевих засобів оцінювання параметрів безпеки, інтеграції параметрів безпеки та методів подання експертних знань, проектування шаблонів поведінки, визначення часових характеристик

використання та ефективності розробки нейромережевих засобів забезпечує можливість їх верифікації, дозволяє розширити функціональні можливості та, відповідно до розробленого генерального критерію, обрати найбільш ефективний нейромережевий засіб;

– отримали подальший розвиток структурні рішення нейромережевих систем оцінювання параметрів безпеки для розпізнавання кібератак, які за рахунок використання модулів класифікації параметрів кібератак, формування подібних кібератак, формування параметрів розробленої марківської моделі шаблону поведінки, підсистеми первинного визначення параметрів кібератак, модулів інтеграції параметрів безпеки, визначення обчислювальних обмежень, розрахунку критеріїв оптимізації виду та показників ефективності нейромережевої моделі, формування продукційних правил підсистеми експертного оцінювання параметрів нейромережевих засобів, модулів розробки MPNN, визначення доцільності застосування, оптимізації виду та верифікації нейромережевих моделей підсистеми розробки нейромережевих моделей, дозволяють зменшити похибку класифікації кібератак, верифікувати отримані результати і забезпечити оперативну адаптацію до умов застосування та нових типів кібератак.

Практичне значення одержаних результатів. Отримані у дисертаційній роботі наукові результати є методологічною базою для розробки і впровадження ефективних інструментальних засобів у вигляді програмних або програмно-апаратних модулів оцінювання параметрів безпеки Інтернет-орієнтованих інформаційних систем для розпізнавання кібератак, які мають підвищену оперативність, адаптованість до умов застосування, нових видів кібератак та низьку обчислювальну ресурсоємність.

Практична цінність полягає у наступному:

– розроблені алгоритми визначення параметрів багатошарового персептрону, які базуються на створеній моделі багатошарового персептрону, дозволяють в 1,5-6 разів зменшити обчислювальні витрати на навчання та до 2 разів зменшити сумарну похибку його навчання, що

забезпечує зменшення ресурсоємності та похибки класифікації інструментальних засобів оцінки параметрів безпеки для розпізнавання кібератак, які створені на його основі;

– на основі розробленого методу подання експертних знань для навчання нейромережевої моделі створено прикладне програмне забезпечення, яке на основі динамічної оцінки параметрів мережевого трафіку може оперативно адаптуватись до розпізнавання нових типів мережевих кібератак;

– на основі розробленого методу проектування шаблону поведінки створено прикладне програмне забезпечення для прогнозування параметрів навантаження веб-серверу, яке дозволяє до 2 разів підвищити точність прогнозу зазначених параметрів;

– розроблене спеціалізоване програмне забезпечення, що базується на створених нейромережевих методах та моделях, дозволило підвищити захищеність інформаційних ресурсів та підвищити оперативність створення алгоритмів функціонування апаратних засобів захисту інформації, що підтверджується актами впровадження у діяльність Київського національного університету будівництва і архітектури (акт впровадження від 24.02.2014) та Інституту проблем моделювання в енергетиці ім. Г. Є. Пухова НАН України (акт впровадження від 12.01.2015).

– розроблені програми, що реалізують запропоновані моделі та методи, впроваджені в навчальний процес на кафедрі безпеки інформаційних технологій Національного авіаційного університету (акт впровадження від 17.02.2015) та на кафедрі системного програмування та спеціалізованих комп’ютерних систем Національного технічного університету України «Київський політехнічний інститут» (акт впровадження від 24.02.2015).

Особистий внесок здобувача. Всі основні результати дисертаційної роботи отримані здобувачем самостійно. У роботах, опублікованих із співавторами, здобувачу належить: метод оцінки ефективності нейромережевих засобів [2, 58, 59], підхід до верифікації нейромережевих

методів розпізнавання кібератак [3], метод застосування продукційних правил [6], підхід до застосування нейромережевих моделей [12], визначення типу та параметрів архітектури нейронної мережі [13, 16, 51], метод оцінки нейромережевих засобів щодо можливостей виявлення кібератак [14], критерій оптимізації та оптимізаційна моделі [25, 50], концепція застосування спектрального аналізу статистичних даних [34], методологія використання нейромережевих технологій в системах розпізнавання атак [40], марківська модель динаміки параметрів технічного стану та марківська оптимізаційна модель [41-44], негомогенна марківська модель [55], метод оптимізації нейромережевої моделі [57].

Апробація результатів дисертації. Основні результати дисертації доповідались, обговорювались та отримали позитивні оцінки на наступних конференціях: Міжрегіональний семінар наукової міжвідомчої Ради НАН України «Технічні засоби захисту інформації» (2004-2008); V Міжнародна науково-практична конференція «Проблеми впровадження інформаційних технологій в економіці» (Ірпінь, 2004); Международная научно-практическая конференция «Единое информационное пространство '2004», (Днепропетровск, 2004); 70 наукова конференція молодих вчених, аспірантів та студентів (Київ, 2004); VI Всеукраїнська науково-практична конференція «Комп'ютерне моделювання та інформаційні технології в науці, економіці та освіті» (Кривий Ріг, 2005); Четверта науково-технічна конференція «Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні» (Київ, 2006); IX Международная научно-практическая конференция «Безопасность информации в информационно-телекоммуникационных системах» (Киев, 2006); Міжнародна науково-теоретична конференція «Актуальні проблеми державного управління та документо-інформаційного забезпечення апарату влади» (Київ, 2006); II Міжнародна науково-технічна конференція «Сучасні інформаційно-комунікаційні технології /COMINFO' 2006/» (Київ, 2006); III Міжнародна науково-методична конференція «Болонський процес: трансформація

навчального процесу у технологію навчання» (Київ, 2007); IV Міжнародна науково-методична конференція «Сучасні тенденції розвитку вищої освіти, трансформація навчального процесу у технологію навчання» (Київ, 2007); VI Міжнародна науково-практична конференція «Проблеми впровадження інформаційних технологій в економіці» (Ірпінь, 2007); Науково-практична конференція «Інформаційна безпека» (Київ, 2009); Міжнародна науково-практична конференція «Моделювання об'єктів, процесів та систем» (Київ, 2011); V Міжнародна науково-технічна конференція ACSN-2011 «Сучасні комп'ютерні системи та мережі: розробка та використання» (Львів, 2011); VIII Наукова конференція Державного університету інформаційно-комунікаційних технологій «Сучасні тенденції розвитку технологій в інфокомунікаціях та освіті» (Київ, 2011); VII Міжнародна науково-технічна конференція «Сучасні інформаційно-комунікаційні технології» COMINFO' 2011 (Київ, 2011); VIII Міжнародна науково-практична інтернет-конференця «Проблеми впровадження інформаційних технологій в економіці» (Ірпінь, 2012); Перша всеукраїнська науково-практична конференція Moodle Moot Ukraine 2013 «Теорія і практика використання системи управління навчанням Moodle» (Київ, 2013); II Всеукраїнська науково-практична конференція «Соціальні комунікації: стан, проблеми, тенденції» (Київ, 2014); Друга міжнародна науково-практична конференція MoodleMoot Ukraine 2014 «Теорія і практика використання системи управління навчанням Moodle» (Київ, 2014); Всеукраїнська науково-практична конференція «Стратегії розвитку інформаційного культурно-освітнього та економічного простору України» (Київ, 2014); IV міжнародна науково-технічна конференція ITSEC (Київ, 2014); The Sixth World Congress «Aviation in the XXI-st Century», «Safety in Aviation and Space Technologies» (Kyiv, 2014).

Публікації. За тематикою дослідження опубліковано 59 наукових праць, серед них 1 одноосібна монографія, 44 статей у фахових наукових виданнях (з них 31 одноосібна), 13 у збірниках праць конференцій, 12 статей опубліковано у виданнях, які включені до міжнародних наукометричних баз.

РОЗДІЛ 1
АНАЛІЗ НЕЙРОМЕРЕЖЕВИХ ЗАСОБІВ ОЦІНКИ ПАРАМЕТРІВ БЕЗПЕКИ ІНТЕРНЕТ-ОРІЄНТОВАНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

1.1. Проблема оцінювання параметрів безпеки інформаційних систем

Однією із основних ознак розвитку сучасного суспільства є подальше зростання залежності від якості й надійності комп'ютеризованих ІС, що застосовуються в різноманітних галузях людської діяльності. Відповідне посилення стратегічної спрямованості інформаційних ресурсів обумовлює необхідність підвищення вимог до рівня їх інформаційної безпеки. Проблема загострюється тим, що особливості найбільшої глобальної мережі Інтернет, з якою інтегровано більшість ІС, і використання загальнодоступного програмного забезпечення призводять до значних обсягів випадкових і непередбачених негативних впливів на вказані системи. Зазначимо, що Інтернет-орієнтація ІС розглядається в ракурсі необхідності захисту ресурсів таких систем від кібератак в процесі реалізації базових технологічних процесів отримання, зберігання, транспортування, обробки та представлення інформації. При цьому під поняттям кібератак будемо розуміти реалізацію у кібернетичному просторі загроз безпеці його компонентів (а саме конфіденційності, цілісності та доступності) з урахуванням їх вразливостей [30]. В той же час кіберпростір – це віртуальний простір, отриманий у результаті взаємодії користувачів, програмного та апаратного забезпечення, мережевих технологій для підтримки та управління процесами перетворення інформації з метою забезпечення інформаційних потреб суспільства [30].

В загальному випадку під поняттям інформаційна безпека розуміють стан захищеності життєво важливих інтересів людини, суспільства і держави, при якому запобігається нанесення шкоди через: неповноту, невчасність та невірогідність інформації, що використовується; негативний інформаційний вплив; негативні наслідки застосування інформаційних технологій; несанкціоноване поширення,

використання, порушення цілісності, конфіденційності та доступності інформації [13, 217]. В той же час під поняттям безпеки інформації (information security) розуміють стан інформації, в якому забезпечується збереження визначених політикою безпеки властивостей інформації. При цьому під поняттям захисту інформації в інформаційних системах розуміють діяльність, яка спрямована на забезпечення безпеки оброблюваної в ІС інформації та ІС в цілому, і дозволяє запобігти або ускладнити можливість реалізації загроз, а також знизити величину потенційних збитків внаслідок реалізації загроз.

В [206] з позицій спричинення недоліків та деструктивних впливів від інформаційної безпеки відділяють функціональну. При цьому у визначенні функціональної безпеки акцент ставиться на правильності функціонування і вважається, що вона в основному пов’язана з ненавмисне реалізованими деструктивними факторами, помилки носять випадковий характер, а еталонний стан об'єкта, відхилення від якого вказує на помилку – відомий. Однак складність та багатофакторність процесів функціонування ресурсів сучасних Інтернет-орієнтованих ІС в більшості випадків вказує на неможливість окремого оцінювання параметрів функціональної та інформаційної безпеки.

Підтверджується висновок про недоцільність відокремлення параметрів оцінювання і в [39], де стверджується, що рівень функціональної безпеки системного засобу визначається тим, наскільки адекватно співвідноситься набір механізмів захисту з умовами конкретного використання і тим, наскільки коректно дані механізми реалізовані. При цьому вказується, що як недостатність механізмів захисту, так і некоректність їх реалізації визначають вразливість системних засобів. Розділу дестабілізуючих факторів на навмисні та випадкові не передбачено. Крім того, в [119] використано визначення інформаційна безпека, яке включає в себе захист інформації від випадкових або навмисних впливів штучного або природного характеру.

Разом з тим, в закордонних публікаціях, присвячених оцінці безпечності інформаційних технологій [253, 254] містяться вимоги до функціональних вимог безпеки. Тому, незважаючи на деяку неоднозначність сучасної термінології, можна

вважати, що процедура оцінювання ПБ ІС зводиться до визначення величин параметрів безпеки, що свідчать про наявність/відсутність кібератак. При цьому під поняттям множини параметрів безпеки будемо розуміти множина параметрів, які відображають стан безпеки об’єктів захисту ІС. В загальному випадку множина ПБ Інтерент-орієнтоваих ІС формуються на основі аналізу:

– параметрів вхідних та вихідних мережевих з’єднань по різноманітним протоколам;

– потенційно небезпечного програмного коду, який передається в ІС;

– параметрів, що відображають функціонування системного та прикладного програмного забезпечення ІС;

– функціональні параметри апаратного забезпечення ІС;

– параметри, що характеризують зміст інформації, котра передається в ІС.

Джерелом статистичних даних для формування такої множини ПБ являються: системні журнали операційних систем робочих станцій та серверів ІС, бази даних засобів захисту інформації (мережевих екранів, антивірусів, систем захисту від спаму, DLP-систем), а також бази даних параметрів кібератак (КДД-99).

Проведемо декомпозицію проблеми оцінювання ПБ Інтернет-орієнтованих ІС для розпізнавання кібератак. Зазначимо, що акцент ставиться на ІС загального призначення, які в основному використовуються в сферах організаційного управління, промисловості та економіці. Відповідно [1, 2, 48], характерними властивостями Інтернет-орієнтованих ІС являються:

– підтримка типових Інтернет-сервісів (веб-сайту, електронної пошти),

– складність опису (досить велика кількість функцій, процесів, елементів даних і складні взаємозв'язки між ними);

– різнорідність методів та моделей, використаних при побудові її компонентів;

– наявність сукупності тісно взаємодіючих компонентів (підсистем), що мають свої локальні задачі й цілі функціонування (наприклад, традиційних додатків, пов'язаних з обробкою транзакцій і рішенням регламентних задач, і додатків аналітичної обробки (підтримки прийняття рішень), що використовують нерегламентовані запити до даних великого об'єму;

– функціонування в неоднорідному середовищі на декількох апаратно-програмних платформах;

– необхідність постійної інтеграції в ІС існуючого і новоствореного програмного забезпечення;

– використання програмного забезпечення, створеного роз'єднаними і різнорідними групами розробників з різним рівнем кваліфікації і традиціями використання тих або інших інструментальних засобів;

– використання програмного забезпечення, яке в багатьох випадках не має офіційної підтримки та несе в собі потенційну загрозу безпеці за рахунок помилок, та люків, які можуть проявлятись тільки в певних умовах експлуатації;

– широке використання в програмного забезпечення архітектури розподілених об’єктів;

– складності адміністрування як в штатних умовах експлуатації, так і при модифікації програмного забезпечення;

– формування управлінських рекомендацій на основі обробки великих обсягів різнорідної інформації;

– необхідність постійної актуалізації інформаційних ресурсів;

– тісна інтегрованість з іншими Інтернет-орієнтованими ІС, частина з яких несуть в собі потенційні загрози як навмисного, так і ненавмиснго характеру;

– тісна взаємодія з зовнішнім Інтернет-середовищем, яке включає в себе велику кількість інформаційно-програмних деструктивних засобів;

– стандартизація та уніфікація процедур взаємодії між різними функціональними блоками ІС;

– інтелектуалізація процедур обробки даних, що значно ускладнює оцінювання управляючих сигналів ІС;

– адаптованість до користувачів з різною кваліфікацією, що значно зменшує ефективність захисту від деструктивних впливів;

– взаємодією з віддаленими користувачами;

– децентралізованість управління як системою захисту, так і всією ІС в

цілому, що в багатьох випадках визначає запізнення та зменшення ефективності управлінських впливів.

Простір означених властивостей показано на рис. 1.1, а їх аналіз дозволяє стверджувати, що основними факторами, які визначають особливості оцінки ПБ вітчизняних Інтернет-орієнтованих ІС, являються:

– складність розпізнавання кібератак внаслідок складності встановлення явного зв'язку між порушенням інформаційної безпеки і певним видом кібератак, складністю визначення вразливостей програмного забезпечення та наслідків реалізації кібератак, можливістю виникнення порушення інформаційної безпеки без явно вираженої кібератаки, високої різноваріантності кібератак;

– виникнення нових видів кібератак по причині постійного вдосконалення методів та засобів здійснення кібератак, використання нових Інтернет-сервісів [1, 19, 59],

– необхідність функціонування при обмежених обчислювальних ресурсах, спричинених використанням бюджетного апаратного забезпечення та розташування на одній апаратній платформі Інтернет-серверів разом з СЗІ [2, 63, 66, 108],

– варіативність умов застосування, що обумовлюється реконфігурацією ІС, зміною програмно-апаратного забезпечення об'єктів ІС, модифікацією Інтернет-сервісів, кваліфікацією адміністративного персоналу та ін.

При цьому результати [77, 78, 82] вказують на те, що типові порушення захищеності Інтернет-орієнтованих ІС виникають по причині деструктивного впливу:

- ШПЗ, розміщеного на веб-сторінках;
- ШПЗ, яке розповсюджується за допомогою електронної пошти;
- витоків текстової інформації з використанням засобів електронної пошти;
- нецільових електронних листів (спаму);
- віддалених мережевих кібератак на Інтернет-сервери.

Наслідками деструктивних впливів може бути як порушення інформаційної безпеки типових Інтернет-сервісів, так і порушення інформаційної безпеки всіх інших функціональних блоків ІС.

Рис. 1.1. Властивості Інтернет-орієнтованих ІС

Також в [64, 88. 92] вказується на необхідність захисту від деструктивних впливів за допомогою спеціалізованих програмних засобів захисту, функціонування яких вимагає оцінки ПБ. Перелік найбільш важливих для Інтернет-орієнотваих ІС засобів захисту показано на рис. 1.2.

Перелік основних причин порушення захисту ІС добре співвідноситься з переліком найбільш актуальних, не вирішених задач оцінки ПБ – розпізнавання віддалених мережевих кібератак на ІС [37, 55, 76, 88, 115], розпізнавання ШПЗ [11,

12, 13, 19, 25, 106, 109, 111] та класифікації електронних листів для розпізнавання спаму та витоків інформації [25, 193, 214, 270].

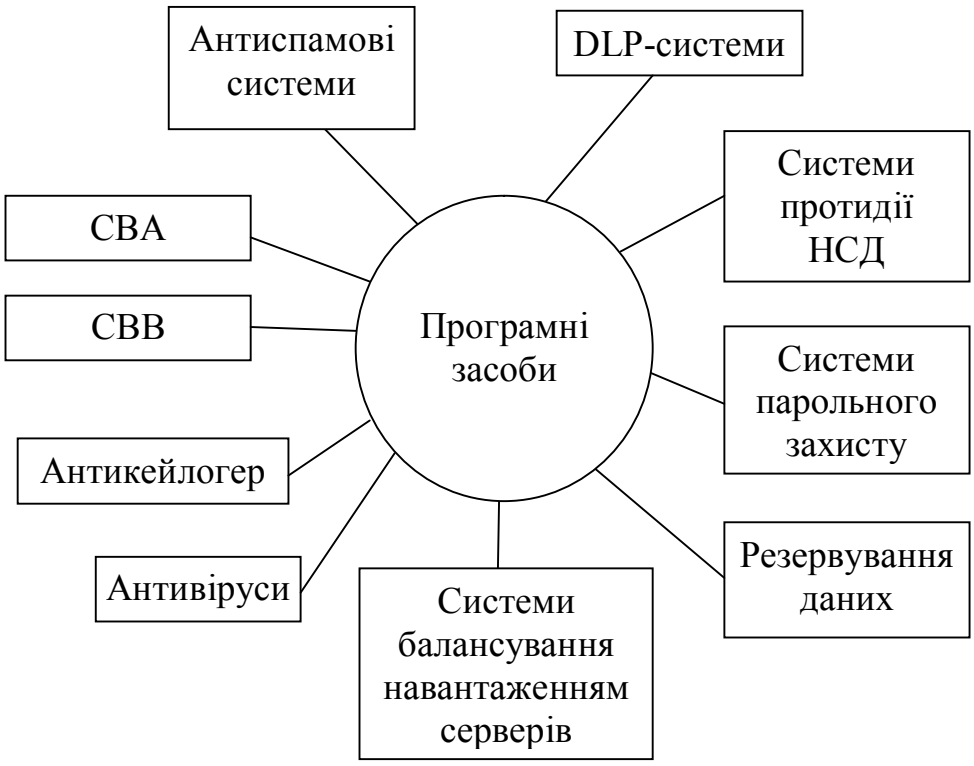


Рис. 1.2. Програмні засоби захисту в яких оцінюються параметрів безпеки

В перерахованих роботах вирішувати вказані задачі пропонується за рахунок розробки відповідних моделей моніторингу ПБ. При цьому вказується на обмеженість перспектив застосування моделей, в яких використовуються явні алгоритмічні правила прийняття рішень. Тому, на погляд автора, що збігається з результатами [72, 101, 124, 125, 140, 228, 229, 231, 232], створювати методи та моделі оцінювання ПБ для розпізнавання кібератак слід на основі застосування методів теорії штучних НМ, які довели свою ефективність при розв'язанні подібних економічних, фінансових та технічних задач. При цьому в літературі не наведено комплексної методології до розробки нейромережевих засобів ЗІ. Для окреслення напрямків досліджень проведено аналіз перспектив вирішення актуальних задач оцінювання ПБ для розпізнавання кібератак, досліджено можливості застосування НМ для оцінювання ПБ та проаналізовано можливості сучасних НМС ЗІ.

1.2. Актуальні задачі оцінювання параметрів безпеки інформаційних систем

Розпізнавання мережевих кібератак. Відповідно [93], під поняттям мережевої кібератак будемо розуміти кібератаку, реалізація якої пов’язана з деструктивним впливом, що здійснюється за допомогою мережевих каналів зв’язку. Для розпізнавання мережевих кібератак використовуються СВА, які представляють комплекс засобів, призначених для моніторингу подій, що відбуваються в ІС, для подальшого аналізу з метою визначення ознак порушення безпеки об'єкту моніторингу [7, 16, 21, 22, 37, 76, 85, 115, 125, 228]. Для прийняття рішень в СВА використовуються два основних методи – визначення аномалій та визначення зловживань. Робота аналізатора при визначенні аномалії базується на припущенні, що ознакою атаки є відхилення поточних величин ПБ від ШНП. Для визначення ШНП застосовуються статистичні моделі [89]. В деяких СВА формується комплексний показник аномалій. При формуванні даного показника для визначення взаємозв'язків між показниками використовуються коваріаційні матриці. Також використовується підхід до визначення аномалій з використанням методу прогнозу подій, який дозволяє виявити кібератаку на ранніх етапах її здійснення. Суть методу полягає в прогнозуванні кібератаки на основі аналізу попередніх подій, пов'язаних з об'єктом захисту [104]. До недоліків відносять високий рівень помилкових спрацьовувань, в основному за рахунок недосконалості моделей ШНП, які не дозволяють достатньо точно визначити прогнозовані величини ПБ [109]. СВА, що використовують метод визначення зловживань аналізують послідовність подій, пов'язаних з діяльністю об'єкта захисту та порівнюють їх з зразками відомих атак. Такі зразки називають ША, а сам метод – визначення атак на основі сигнатур. По причині не повноти інформації та наявності шумів при реєстрації ПБ труднощі викликає задача розрахунку відповідності ША реальним подіям, що стосуються об'єкта захисту. Для вирішення цієї задачі застосовуються різноманітні методи – експертний, аналізу переходів, моделювання атак. При застосуванні експертного методу відомі кібератаки описуються в вигляді деякого набору правил, виконання

яких сигналізує про реалізацію кібератак. Метод аналізу переходів передбачає представлення мережевої кібератаки у вигляді послідовності переходів об'єктів захисту із одного стану в інший [119]. При застосуванні методу моделювання кібератак попередньо сформовані послідовності подій, характерні для реалізації кібератаки, порівнюються з поточними показниками. В результаті порівняння формується висновок про ймовірність здійснення кібератаки. Часто використовуються статистичні моделі зміни ПБ ІС під час кібератаки. В цілому метод визначення зловживань дозволяє достатньо ефективно виявляти кібератаки відомих типів при низькому показнику хибних спрацювань, але не дозволяє виявити кібератаку, зразок якої не відомий. Важливою та на сьогоднішній час не вирішеною задачею є формування ША. Загалом ШНП та ША називають ШП. Хоча розробці ШП присвячено досить багато робіт [7, 16, 37, 55, 85, 115, 124, 125, 133], але практичний досвід та результати [118, 121] вказують на те, що ці шаблони не достатньо адаптовані до типової динаміки ПБ Інтернет-орієнтованих ІС, яка в більшості випадків не може бути адекватно описана за допомогою однорідних моделей. З метою підтвердження вказаної гіпотези, на основі [96, 133] та статистичних даних, зібраних автором, проаналізовано ряд типових випадків процесу зміни ПБ об’єктів захисту Інтернет-орієнтованих ІС. Так, на рис. 1.3, 1.4 показано наведені в [133] графіки зміни кількості TCP/IP пакетів, отриманих веб-сервером.

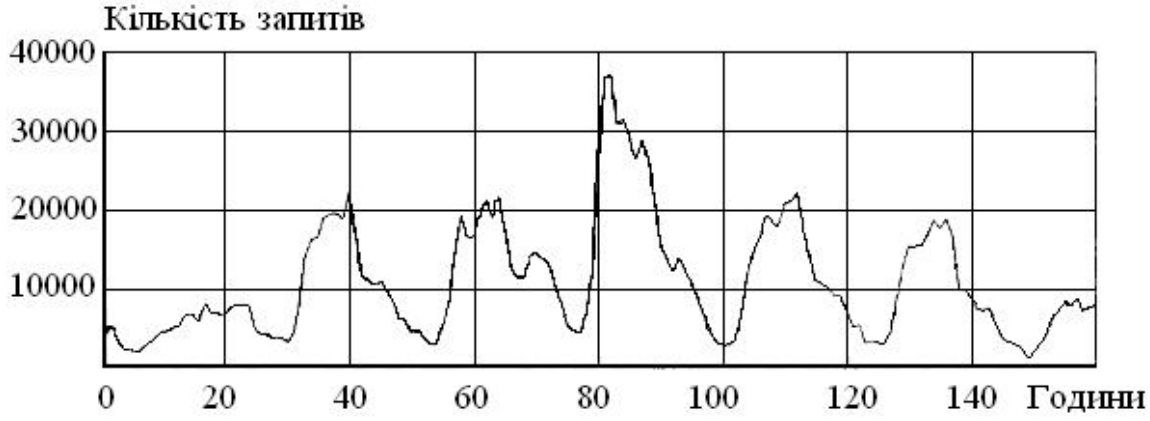


Рис. 1.3. Зміна кількості TCP/IP пакетів при вікні спостережень 1 година

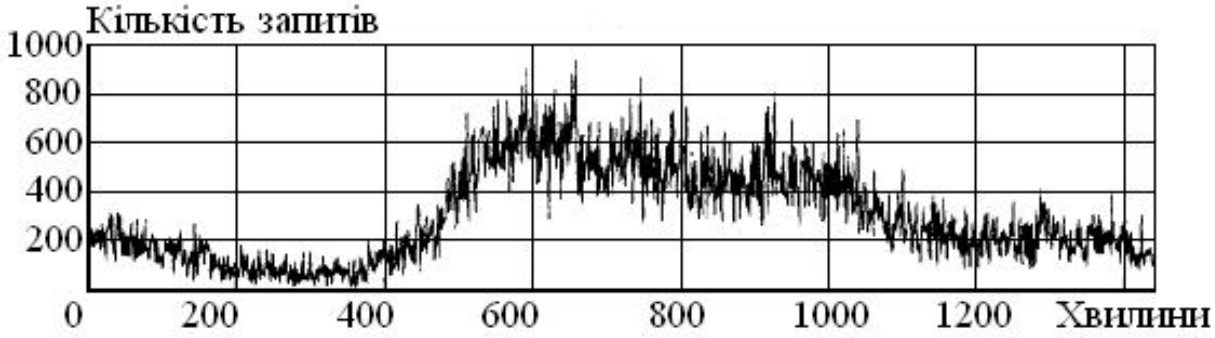


Рис. 1.4. Зміна кількості TCP/IP пакетів при вікні спостережень 1 хвилина

Аналіз графіків рис. 1.3, 1.4 показує, що при нормальних умовах експлуатації екстремальні величини кількості запитів, які поступають до веб-серверу, значно відрізняються від середніх значень. В загальному випадку максимальні значення в 5-10 разів перевищують середні. Крім того, прослідкується певна циклічність процесу зміни кількості запитів. Ще однією особливістю процесу є те, що циклічність процесу зміни кількості запитів відбувається в декількох масштабах часу. Також із аналізу рис. 1.4 можливо зробити висновок, про наявність флуктації трафіку при невеликих вікнах спостережень (1-60 с.), в межах яких кількість переглядів сторінок та кількість відвідувачів веб-сайту не змінюється. Цей факт зумовлений тим, що типова веб-сторінка значно більша від максимального розміру мережевого пакету. Тому одиночний запит веб-сторінки призводить до стрімкого, короткочасного зростання кількості мережевих пакетів [129]. Досить схожі результати, отримані і в [92]. На відміну від [129] в цій роботі наведено зміну обсягу вхідного та вихідного мережевого трафіку комп'ютера-сервера корпоративної ІС, який забезпечував не тільки функціонування веб-серверу, але й серверу баз даних, файлового серверу та серверу друку. В [96] доведено автомодельність процесу зміни трафіку з коефіцієнтом самоподібності 0.7 – 0.85. Також показано наявність циклічного ефекту для різних часових діапазонів, що співпадає з результатами [129]. Зазначимо, що в [96] статистичні дані було зібрано за допомогою спеціалізованих програм. Даний підхід доцільний при застосуванні в експериментальних цілях, але

не завжди прийнятний на практиці внаслідок організаційних обмежень та збільшення використання обчислювальних ресурсів комп'ютера-сервера. Крім того, в статистиці відсутня інформація стосовно багатьох параметрів, які широко використовуються в ШП, наприклад, кількість звернень до веб-серверу з однієї IP-адреси (хости). Для виправлення вказаних недоліків автором було зібрано та проаналізовано статистичні дані функціонування веб-серверу, що забезпечував доступ до інформаційного сайту кредитної установи. Веб-сервер розміщувався на окремому комп'ютері та був з'єднаний з мережею Internet виділеним каналом. Статистика зібрана за допомогою аналізу log-файлів веб-серверу та лічильника відвідувань, розміщеного на всіх сторінках сайту. Мінімальний інтервал спостережень 1 с. Термін спостережень 1 рік 2 місяці. Графік динаміки переглядів веб-сторінок на протязі доби показано на рис. 1.5.

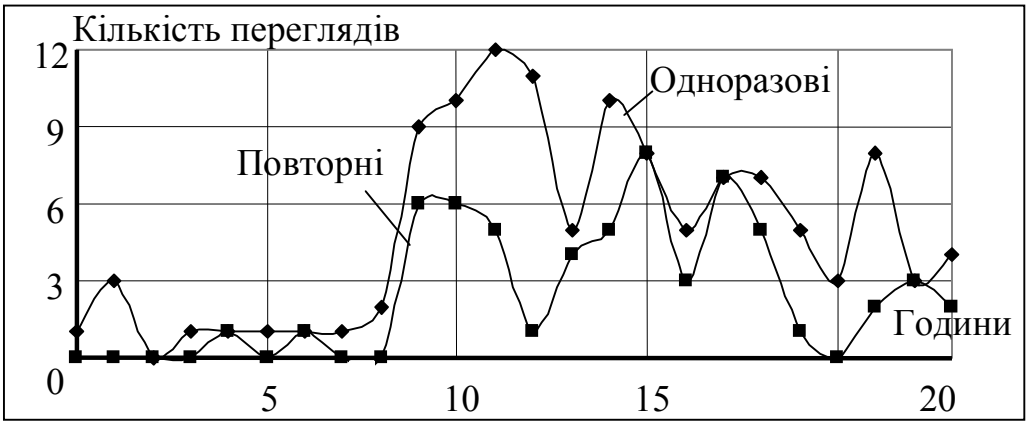


Рис. 1.5. Динаміка одноразових та повторних переглядів веб-сторінок

Аналіз рис. 1.5 підтверджує наявність циклічності в випадкових процесах, пов'язаних з переглядом користувачами сторінок Веб-сайту. Таким чином, доведено, що зміна більшості параметрів ШП для Інтерент-орієнтованих ІС носить циклічний характер на різних часових інтервалах. В [96, 133] для врахування циклічності мережевого трафіку в заданому інтервалі спостережень пропонуються відносно прості лінійні моделі, в яких флуктація описується за допомогою двох допоміжних

коефіцієнтів. Перший коефіцієнт розраховується як відношення максимальної зафіксованої вхідної інтенсивності запитів до середньої інтенсивності запитів на протязі заданого інтервалу спостережень. Другий коефіцієнт розраховується як відношення часу, на протязі якого миттєва вхідна інтенсивність запитів перевищувала середню інтенсивність, до загального терміну спостережень. Зазначена низька точність таких моделей та рекомендується використовувати їх тільки для приблизних розрахунків. Також зроблена спроба створення моделі динаміки трафіку з різними степенями самоподібності. Однак запропонована модель дозволяє лише розраховувати коефіцієнти самоподібності процесу тільки з одним циклом. Крім того, не показано можливості безпосереднього моделювання динаміки процесу. Тому доцільність її застосування в ШП викликає сумніви. Разом з тим, спільний висновок [96, 133,252] полягає в неадекватності пуасонівської моделі динаміки більшості функціональних параметрів. Вказані причини підтверджують актуальність та важливість вдосконалення ШП. Вдосконалити ШП можливо завдяки використанню апарату марківської апроксимації, який широко застосовується для прогнозування параметрів в задачах діагностики технічних систем [8, 9, 18, 99, 138, 144, 172, 206]. При цьому в літературі не наведено методів побудови марківських моделей ШП ПБ, які б адекватно враховували типові статистичні залежності ПБ Інтернет-орієнтованих ІС та могли б бути використані для навчання НММ призначених для розпізнавання мережевих кібератак. При цьому марківські моделі повинні бути неоднорідними, а для визначення перехідних точок необхідно враховувати можливу циклічність процесу на різних часових інтервалах. На наш погляд, що збігається з висновками [!!!], врахувати циклічність процесу доцільно за допомогою добре апробованого методу Фур'є спектрального аналізу даних.

Розпізнавання шкідливого програмного забезпечення. Відповідно [96], шкідливим будемо називати будь-яке ПЗ, яке може заподіяти шкоду ІС. За останні декілька років саме ШПЗ стало однією із основних причин порушень захищеності ІС [11, 12, 13, 19, 25, 109, 111, 129, 225]. При цьому для більшості Інтернет-орієнтованих ІС характерним є організація електронного документообігу при адміністративному обмеженні повноважень користувачів на інсталяцію програмного

забезпечення та доступу до файлів. В цьому випадку ймовірними шляхами зараження є використання листів електронної пошти, офісних документів та перегляд веб-сайтів. Для вказаних ресурсів ШПЗ, як правило, розроблюється за допомогою скриптових мов програмування. Таким чином, ШПЗ, створене за допомогою скриптових мов програмування, є однією із основних загроз інформаційній безпеці ІС [96, 113]. Зазначимо, що до скриптового ШПЗ також відносяться шкідливі макроси, які використовують можливості макромов, вбудованих в системи обробки даних. Найбільш поширені макровіруси та макротрояни, написані на мові VBA і пристосовані для функціонування в середовищах MS Office, AutoCAD, 1С "Предприятие" та 1С "Бухгалтерия". Крім того, в сучасних версіях ОС Windows вбудовано скриптовий інтерпретатор Windows Scripting Host, який дозволяє виконувати скрипти (макроси), написані на мовах VBScript та JScript. Запуск макроса може бути здійснено користувачем при відкритті файлу. Ця особливість використовується для активізації скриптових вірусів та троянів, що розповсюджуються в вигляді файлів, прикріплених до листів електронної пошти. Результати [54, 101, 177, 189, 191, 192, 194, 200, 229] вказують, що більшість поштових вірусів та троянів функціонують в середовищі інтерпретатора WScript і написані на мові програмування VBScript. Тому більшість поштових вірусів та троянів не відрізняються від макровірусів та макротроянів MS Office. ШПЗ, призначене для нанесення шкоди при перегляді Веб-сайтів, має певні відмінності. Воно вбудоване в веб-сторінку та виконується в середовищі браузеру. Для створення такого ШПЗ використовуються мови програмування: Java, C++, VBA, JavaScript, JScript, VBScript та ActiveScript.

В основному для захисту від скриптового ШПЗ використовуються антивірусні сканери та поведінкові аналізатори. Найчастіше для виявлення ШПЗ використовується метод пошуку сигнатур. Особливістю пошуку сигнатур скриптового ШПЗ є те, що сканер може аналізувати програмний код скрипта в текстовому вигляді. Порівняно з стандартними файловими вірусами означена особливість значно спрощує роботу сканера та дозволяє проводити аналіз функціональності макроса. Однак найбільш розповсюджений метод пошуку

сигнатур дозволяє розпізнавати тільки відоме ШПЗ та відкриває шлях для обходу антивірусного захисту поліморфному та зашифрованому (обфусифікованому) ШПЗ. Іншими важливими недоліками методу сигнатур є відносно низька швидкість пошуку всіх видів ШПЗ та необхідність постійного оновлення бази сигнатур. Крім того, в більшості антивірусних сканерів задекларовано використання евристичних методів пошуку ШПЗ, реалізація яких практично не документується. Проте аналіз [1, 19, 62, 82] вказує на те, що в більшості випадків базою цих методів є статистичний аналіз послідовності виконання програмного коду об'єкта, який перевіряється. Відзначимо, що навіть по явно завищеним рекламним заявам розробників сучасні евристичні методи дозволяють виявити тільки близько 50% ШПЗ з невідомою сигнатурою. Що стосується поведінкових аналізаторів, то вони досить вузько застосовуються на практиці, так як більшість дій, характерних для ШПЗ, можуть виконуватися і звичайними програмами. Ще одним засобом захисту від скрипового ШПЗ є модулі блокування скриптів (макросів), що входять до складу СЗІ офісних пакетів та браузерів. Однак їх широкому застосуванню заважає значне складність їх оперативного налаштування та значне обмеження функціональних можливостей використання офісних документів і веб-сторінок. Наприклад, блокування браузером спливаючих вікон значно спотворює інформацію веб-сторінки, а дозвіл їх перегляду є прогалиною в захисті. Отже, задача розпізнавання скриптового ШПЗ далека від свого вирішення, що підтверджується незалежним тестуванням антивірусних засобів [121, 122].

Захист електронної пошти. Електронна пошта є одним із найбільш поширених та важливих сервісів сучасних ІС. На сьогодні вважається, що ефективність та безпечність функціонування електронної пошти в основному визначається рівнем захисту від спаму та від витоків інформації [194].

Під терміном спам розуміють масово розповсюджувані листи, зміст яких носить рекламний або шахрайський характер. В сучасних умовах розсилка спаму – це високо прибутковий бізнес, підкріплений відповідним ринком і стабільним попитом. В російськомовній зоні Інтернет обсяг спаму складає близько 70-90% від загального обсягу всієї електронної пошти [193, 270]. Існуючі програмно-технічні

методи боротьби зі спамом функціонують по принципу розпізнання – блокування (знищення) спаму [250]. Основною проблемою є класифікація одержаних листів на цільові та спам. Проаналізуємо особливості сучасних методів класифікації електронних листів. Метод чорного, білого і сірого списків. Базою методу є аналіз зворотної адреси відправника листа. Основний недолік даного методу полягає в тому, що адреса не обов'язково вказує на джерело спаму. Наприклад, спам-лист може прийти з динамічної IP-адреси, або розсилка здійснена без відома власника адреси. Використання сірого списку доцільне тільки при невеликому обсязі листування з обмеженим колом осіб. В протилежному випадку ведення сірого списку потребує великих затрат на переконфігурацію. Крім того, сучасні спам-засоби дозволяють генерувати підтвердження відправки спам-листа. Метод фіксації масових розсилок електронних листів. Листи класифікуються як спам, якщо обсяг відправки пошти з однієї адреси (з однієї підмережі) за короткий термін часу перевищує граничну величину. Недоліками методу є необхідність контролю за всім простором поштових відправлень Інтернет та неефективність при невеликих спам-розсилках. Технологія верифікації відправника Sender Permitted From. Адміністратор домену публікує дані, які описують можливі джерела електронної пошти з адресами відправника з цього домену. Опубліковані дані називаються SPF-записом або SPF-політикою. Приймаючий поштовий сервер класифікує спам на основі порівняння адреси відправника з SPF-записом. Однак спамер може самостійно зареєструвати велику кількість доменів з коректними SPF-записами і розсилати спам з цих доменів. Також спамери можуть використовувати безкоштовні домени 3-го і більш рівнів. Крім того, підтримка SPF призводить до значних обчислювальних витрат системи пересилки електронної пошти. Метод розпізнання спаму по ключовим словам (словосполученням), які визначаються користувачем у вигляді набору правил. Даний метод не знайшов широкого розповсюдження через складнощі при формуванні вказаних правил. Метод байєсовської фільтрації. Кожному слову або тегу, що зустрічається в електронній переписці, присвоюється два значення: ймовірність його присутності в спамі та ймовірність його присутності в звичайних листах. Для кожного нового листа за допомогою формули Байєса розраховується

загальна спам-оцінка листа. Якщо величина спам-оцінки більша від граничного значення, то лист класифікується як спам. Ефективність методу безпосередньо залежить від правильності спам-оцінок слів листа. Для цього здійснюється статистичний аналіз як спаму, так і звичайних листів кожного користувача. Таким чином, метод байєсовської фільтрації передбачає деяке запізнення, пов'язане з накопиченням кожним користувачем достатнього об'єму статистичного матеріалу. Ще одним недоліком даного методу є висока ймовірності пропуску спаму, якщо в листі мало слів з високою спам-оцінкою.

У більшості сучасних антиспамових систем реалізовані комплексні методи захисту, які декларують фільтрацію 98% спаму. Однак навіть у найсучасніших поштових службах реакція на новий вид спам-листів складає 20-30 хвилин. При цьому розсилка багатьох мільйонів спам-листів здійснюються за 1-2 години. Тому з великою вірогідністю поштові служби проведуть невірну класифікацію спаму. Таким чином, практично всі існуючі системи розпізнавання спаму не можуть адекватно реагувати на сучасні методи формування і розповсюдження спам-листів. Схожою виглядає ситуація захисту електронної пошти від витоків інформації, яка в більшості ІС реалізується за допомогою DLP-систем. Принцип роботи даних систем полягає у сигналізації та/або блокуванні електронних листів, які містять конфіденційні дані. Труднощі виявлення конфіденційної текстової інформації подібні до розпізнавання спаму і полягають в недостатньому врахуванні змісту листів. Виправити даний недолік можливо за рахунок застосування НММ, що вже довели свою ефективність в системах обробки текстової інформації.

1.3. Дослідження можливостей застосування методів теорії нейронних мереж для оцінювання параметрів безпеки

Під терміном НМ розуміють мережу елементів (штучних нейронів), пов'язаних між собою синаптичними зв'язками [10, 40, 41, 46, 48, 97, 126, 145, 218]. Основними конструктивними параметрами НМ є кількість вхідних, схованих і вихідних нейронів, структура зв'язків (топологія мережі), правила розповсюдження

та комбінування сигналів, правила обчислення вихідного сигналу нейрона та правила навчання, що коректують зв'язки в мережі. Сукупність вказаних параметрів визначають (архітектуру НМ) вид НММ [40-42, 48, 49, 63,78, 97, 112, 116, 185, 218].

Окреслюючи сферу застосування НМ слід врахувати, що можливості мережі значною мірою залежать від виду НММ. Результати [10, 40-42, 48, 49, 51, 58-60, 62, 65, 78, 86, 90, 97, 99, 116, 117, 121, 126, 128, 134, 136, 137, 141, 145, 158, 169, 207, 209, 213, 218] вказують на те, що розвиток сучасних НМ йде шляхом пристосування базових архітектур до вирішення практичних задач. При цьому ряд архітектур вже втратили свої передові позиції і використовуються тільки в якості допоміжних. Базуючись на висновках [10, 40-42, 48, 49, 51, 58-60, 62, 65, 78, 86, 90, 97, 99, 116, 117, 121, 126, 128, 134, 135-137, 141, 145, 155, 158, 169, 185, 207, 209, 210, 213, 218, 269] та результатах п. 1.2 для розгляду виберемо НМ на базі БШП, РБФ, ймовірністні НМ, АРТ, АНМ та ТК. Однак вибрані мережі недостатньо пристосовані до аналізу тексту, що важливо для розпізнавання спаму та витоків тестової інформації. Тому розглянута СНМ, яка успішно використовується в галузі обробки текстової інформації [258-268].

Нейронні мережі на базі багатошарового персептрону. БШП, представляє собою НМ з прямим розповсюдженням сигналу, яка складається із декількох послідовно з'єднаних між собою шарів штучних нейронів [10, 40, 41, 65, 78, 116]. Навчання БШП виконується методом "з вчителем". При цьому вагові коефіцієнти змінюються так, щоб мінімізувати середньоквадратичний функціонал помилки НМ

$$\varepsilon^2(W) = \sum_{i=1}^{N} \left(y_i - y_i^r \right)^2 \to min . \qquad (1.1)$$

де y_i, y_i^r – очікуваний та реальний вихідний сигнал *i*-го вихідного нейрону;

N – кількість вихідних нейронів;

ε – загальна помилка навчання

W – матриця вагових коефіцієнтів синаптичних зв’язків.

Зазначимо, що використання (1.1) призводить до збільшення відносної помилки

навчання для прикладів з невеликими величинами очікуваного вихідного сигналу. Кількість вхідних параметрів БШП має бути обмеженою. Вхідні параметри можуть бути як дискретні, так і неперервні. Вважається, що мінімальна кількість навчальних прикладів має бути в 10-20 разів більша від кількості вхідних параметрів. Максимальна кількість навчальних прикладів залежить від кількості схованих нейронів. До переваг БШП відносять можливість навчання на корельованих та зашумлених навчальних даних. Однак для якісного навчання необхідно пропорційно представити в навчальних даних всі аспекти піддослідного процесу. Процес навчання багато ітераційний та довготривалий. Особливістю навчання БШП є невелика кількість емпіричних параметрів величини яких можуть бути визначені в процесі адаптації НМ до поставленої задачі. Тому результати навчання на однакових прикладах практично незмінні. Результати [140, 210] вказують на потенційну можливість донавчання БШП в процесі застосування, однак наведені підходи потребують доопрацювання. Цим же пояснюється і необхідність вдосконалення методики автономного функціонування БШП. Результати навчання БШП інтерпретуються у вигляді ймовірності та піддаються вербалізації. В теорії [140, 210] інтелектуальні можливості БШП, які оцінюються по критеріям якості навчання, екстраполяції навчальних результатів та обсягу пам’яті вважаються найвищими серед класичних НМ. Технічна реалізація БШП визначається достатньою швидкістю прийняття рішення, що пояснюється необхідністю проведення розрахунків, пов’язаних тільки з прямим проходженням сигналу. Традиційної сферою застосування БШП є системи розпізнавання образів.

Для моделювання часових рядів створено такі модифікації БШП, як мережі Елмена та Джордана, а для аналізу зображень – згорткові НМ. Основою архітектурною відмінністю НМ Джордана є наявність зворотніх зв’язків, які використовуються для подачі на вхід мережі затриманий на один або декілька тактів вихідний сигнал. Мережа Елмана відрізняється від мережі Джордана тільки тим, що зворотні зв’язки йдуть не від вихідних, а від схованих нейронів. Вважається [140, 210] , що наявність зворотніх зв’язків потенційно дозволяє ефективніше, по відношенню до БШП, врахувати передісторію піддослідних процесів та підвищити

робастість до часових спотворень. Однак використання зворотніх зв'язків призводить до значного збільшення терміну навчання. Тому в класичному вигляді є сенс використовувати тільки відносно невеликі мережі Елмена та Джордана. Крім того, використання зворотніх зв'язків збільшує нестійкість процесу навчання, що в призводить до складності автоматизації процесу навчання НМ. Зазначимо, що НМ Елмана знайшла застосування при розпізнавання фонем в процесі аналізу аудіоінформації та для класифікації відеоінформації.

Згорткові НМ являються модифікованим БШП, структура якого пристосована для розпізнавання двохвимірних зображень з високим рівнем шуму. Їх основною сферою застосування є системи розпізнавання рукописного тексту. Крім того, відомі спроби використання в системах комп'ютерного зору та розпізнавання мови. В цих системах, по відношенню до БШП, згорткові НМ мають менші розміри та краще враховують топологію вхідних даних. Однак складний алгоритм навчання призводить до збільшення терміну навчання. В літературі не знайдено методики пристосування як згорткових НМ, так і мереж Єлмена та Джордана до автономного функціонування.

Мережа з радіальними базисними функціями. В найбільш простій формі РБФ складається із вхідного, схованого та вихідного нейронних шарів [65, 97, 116, 185, 218]. Кожен з схованих нейронів призначений для зберігання окремого еталонного образу, який відповідає окремому класу. Після нелінійного перетворення сигнали від нейронів СШН потрапляють у вихідний шар нейронів, що мають лінійні функції активації. Вихід РБФ можливо трактувати як ймовірність віднесення невідомого образу до певного еталону. Процес навчання НМ багатоітераційний. Однак знайти теоретичний функціонал оптимальної кількості ітерацій не вдалось, хоча і вважається що їхня кількість менша ніж у БШП. Слід зазначити практичну незмінність результатів навчання на фіксованій навчальній вибірці. Методики вербалізації РБФ не знайдено. Вхідні параметри РБФ можуть бути як дискретні, так і неперервні. Максимальна кількість навчальних прикладів залежить від кількості схованих нейронів. Базуючись на аналізі структури РБФ можна зробити висновок про низьку якість навчання на корельованих та зашумлених даних. Також необхідно

пропорційно представити в навчальних даних всі аспекти піддослідного процесу. По відношенню до БШП, РБФ дозволяє моделювати довільну функцію за допомогою всього одного проміжного шару, що в деякій мірі спрощує архітектуру. Крім того, РБФ навчається на порядок швидше, а її програмна реалізація простіша. Водночас мережа РБФ має і цілий ряд суттєвих недоліків. В першу чергу, це велика кількість емпіричних параметрів, що використовуються при визначенні вагових коефіцієнтів нейронів. Теоретичні методи не гарантують точного визначення оптимальної кількості схованих нейронів, яка є однією із найважливіших характеристик НМ. По цим причинам РБФ погано пристосована до автономного застосування. Ще одним важливим недоліком є погана екстраполяція результатів за межею області відомих даних. Тому в навчальній вибірці слід представити весь діапазон можливих вхідних даних та еталонних образів. Наведені в [97] результати порівняння обчислювальних можливостей РБФ та БШП вказують на те, що для моделювання складних функцій мережа РБФ потребує більшого числа нейронів. Як наслідок, програмна реалізація РБФ буде проводити класифікацію довше та витрачати більше ресурсів, ніж БШП. Традиційною сферою застосування РБФ є розпізнавання образів.

Топографічна карта Кохонена. Структурно ТК складається із вхідного і вихідного (топографічного) нейронних шарів [65, 78, 86, 97, 116, 218]. Кількість нейронів вхідного шару дорівнює кількості компонент вхідних образів. Кожен вхідний нейрон пов'язаний з кожним топографічним нейроном, який відповідає певному класу образів. Обсяг навчальної вибірки, в якій можуть бути несистематичні помилки, повинен в 5-10 раз перевищувати кількість вхідних параметрів. Навчальні дані можуть мати як дискретний так і неперервний характер. В процесі багатоітераційного навчання розраховуються вагові коефіцієнти топографічних нейронів та відбувається розподіл бібліотечних образів на класи (кластери), кількість яких визначається необхідною точністю розпізнавання. При цьому мережа організуються так, що нейрони, які відповідають образам, розміщеним близько в просторі входів, розміщуються близько і на топографічній карті. Кількість навчальних ітерацій має бути як мінімум в 10 разів більша, ніж кількість навчальних прикладів. Після закінчення навчання на вхід ТК можна

подавати нові образи для розпізнавання. Крім того, ТК може використовуватись як детектор нових явищ [63, 76]. Значним недоліком ТК є велика кількість емпіричних параметрів, від яких залежить якість навчання [97]. Порівняння обчислювальних можливостей НМ типу ТК з БШП вказує на значно менший термін навчання, що разом з можливістю оперативної адаптації меж кластерів до зміни вхідної інформації зумовлює ефективність їх використання з метою розвідувального аналізу даних. Проте узагальнюючі можливості БШП, а також обсяг його пам'яті набагато вищі. Традиційною сферою застосування ТК є візуалізація класифікованих даних в системах розпізнавання образів та аналізу текстової інформації. Відомі спроби застосування ТК в системах розпізнавання звукової інформації та для розв'язання оптимізаційних задач.

Ймовірнісні нейронні мережі. Базовим типом ймовірнісних НМ є PNN, в якій для віднесення невідомого образу x до k-го класу застосовується вираз

$$h_k c_k f_k(x) > h_i c_i f_i(x), \exists i \in \{N\}, \qquad (1.2)$$

де $\{N\}$ – множина всіх класів,

i – довільний клас,

h_k (h_i) – апріорна ймовірність класифікації образу як класу k (i),

c_k (c_i) – ціна помилки класифікації образу як класу k (i),

$f_k(x)$ і $f_i(x)$ – функції щільності ймовірності для класів k та i.

Оцінка функції $f_k(x)$ і $f_i(x)$ визначається на основі безітераційного запам'ятовування навчальних прикладів з застосуванням вагової функції Гауса. Тому процес навчання PNN відбувається швидко. Ще однією перевагою PNN є наявність тільки одного управляючого параметру (радіусу функції Гауса), методика розрахунку якого показана в [93]. Вхідні параметри можуть бути як дискретні, так і неперервні. Кількість навчальних прикладів дорівнює кількості схованих нейронів. Також до переваг PNN відноситься якісна класифікація на невеликій навчальній вибірці, низька чутливість до помилок в навчальних даних, ймовірнісний зміст класифікації, простота реалізації, пристосованість до автономного функціонування.

Загальними недоліками PNN є якісна класифікація тільки в діапазоні навчальних даних, потенційно висока обчислювальна ресурсоємність. Традиційною PNN використовується для виділення найбільш інформативних параметрів.

Асоціативні нейронні мережі. До класичних АНМ відносяться мережі Хопфілда, Хеммінга та Коско [138, 140, 210]. Їх основною перевагою відносно НМ з прямим розповсюдженням сигналу є динамічність та ітераційність обробки даних, що має позитивно впливати на обчислювальні можливості. Навчання АНМ відбувається шляхом безпосередньої обробки навчальних даних. Однак обсяг пам'яті АНМ менший, в порівнянні з БШП. Так, в [65] наведено формулу для розрахунку максимального обсягу збережених образів в мережі Хопфілда при умові безпомилкового розпізнавання всього обсягу пам'яті:

$$p_{\max} \le (0{,}05 \times N). \qquad (1.3)$$

де N – кількість нейронів в мережі.

Навчальні образи повинні бути слабо корельовані між собою. Інакше можливо виникнення перехресних асоціацій при їх пред'явленні на вході мережі. До особливостей мережі Хеммінга відносять можливість розпізнавання тільки бінарних образів, визначення тільки номеру еталону при класифікації та неможливість розпізнавання зашумлених сигналів [65, 97, 116, 218]. Мережу Коско вважають розвитком НМ Хопфілда, призначеним для вирішення задачі встановлення асоціації між вхідними та еталонними образами [126, 185, 218]. Обсяг пам'яті таких НМ дещо вищий, але недостатня апробованість ускладнює їх використання в СЗІ. Традиційною сферою використання АНМ є вирішення задач класифікації зашумлених даних та виділення прототипів. В [65, 97, 218] доведена доцільність обробки статистичних даних за допомогою асоціативних мереж перед їх використанням в БШП з метою зменшення кількість вхідних параметрів. Крім того, відомі спроби застосування АНМ для оптимального розподілу ресурсів та активної кластеризації. До загальних недоліків АНМ відносять обмеженість пам'яті, квадратичну залежність кількості зв'язків від розмірності вхідного сигналу,

непередбачуваність функціонування за рахунок помилкової і нестабільної класифікації та неможливість навчання на корельованих образах [97, 116].

Мережі адаптивної резонансної теорії. Основною особливістю мережі АРТ є можливість динамічного запам'ятовування нових образів без повного перенавчання та втрати інформації про образи, що вже були в ній збережені. Для цього в НМ використовується специфічний по відношенню до класичних НМ дворівневий алгоритм порівняння вхідного образу з вмістом пам'яті [140, 210]. АРТ може використовуватись для розвідувального аналізу даних та здатна навчатись на корельованих навчальних прикладах, з фіксованою кількістю вхідних параметрів. Обсяг навчальної вибірки обмежується тільки обчислювальними можливостями програмно-апаратної реалізації. При цьому навчання мережі АРТ реалізується методом "без вчителя", а очікуваний вихід в навчальних прикладах не використовується. Для якісного навчання в навчальних прикладах слід відобразити всі аспекти піддослідного процесу. Екстраполяція результатів навчання за межі навчальної вибірки малодостовірна. Результат класифікації можливо представити у вигляді ймовірності. Аналіз літератури не виявив алгоритму вербалізації АРТ. Різноманітні модифікації АРТ можуть працювати як з дискретним, так і з неперервними вхідними параметрами. Обсяг пам’яті АРТ по відношенню до БШП дещо менший. До позитивних рис АРТ відносять швидкий доступ до бібліотечних образів, стабільність і закінченість процесів навчання та розпізнавання, короткий термін навчання, зрозумілість функціонування та простоту програмної реалізації. Недоліками АРТ є неможливість довготривалої класифікації зашумлених образів, чутливість навчання до порядку пред'явлення вхідних векторів та велика обчислювальна складність процесу класифікації [78, 116]. Традиційно АРТ використовуються для класифікації образів та розпізнавання зображень.

Семантична нейронна мережа. Даний вид НММ є розвитком активних семантичних мереж та НМ Маккаллока-Питтса та може використовуватись на всіх етапах розбору тексту [239, 240]. Особливістю СНМ є те, що проміжним нейронам призначається відповідність деяких елементів семантики предметної області або моделі тексту. Елемент може представляти окремий символ, сукупність деяких

символів тексту або сукупність понять і відношень між поняттями, що можна абстрагувати як єдине ціле. В випадку наявності відповідного елементу в тексті нейрон приймає значення "істина", а в протилежному випадку – "не правда". Зв'язки між нейронами представляють собою відношення між елементами семантики. Фактори впевненості представляються у вигляді градієнтних величин, що оброблюються і передаються нейронами. Зміст тексту, представлений станом СНМ, оброблюється як потік градієнтних даних, що передається між нейронами. Навчальні дані СНМ повинні представляти базу даних та базу знань імітаційної моделі предметної області. При цьому кількість навчальних прикладів необмежена. Вони можуть бути частково зашумлені та корельовані. Для якісної класифікації в навчальних прикладах мають бути відображені більшість можливих варіантів тексту. Термін навчання досить короткий, а процес навчання та до навчання можна автоматизувати. Крім задачі класифікації вхідної символьної послідовності, СНМ може вирішувати задачу формування коректних словозмін вказаної послідовності.

Проведений аналіз сучасних видів НММ дозволяє стверджувати, що з точки зору їх застосування характеристики задач можливо розділити на категорії, що відповідають: навчальним даним, обмеженням процесу навчання, обчислювальним потужностям, вихідній інформації, технічній реалізації, сфері застосування.

Деталізуємо вказані категорії.

1. Основними характеристики навчальних даних являються:

– Кількість параметрів, що характеризують навчальний приклад.

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

– Кількість доступних навчальних прикладів. Наприклад, в задачах розпізнавання змісту тексту кількість навчальних прикладів можна вважати необмеженою. Для інших задач (розпізнавання мережевих кібератак) кількість навчальних прикладів може бути приблизно рівною кількості вхідних параметрів.

– Наявність помилок (шуму) в навчальних прикладах.

– Наявність кореляції навчальних прикладів.

– Можливість попередньої обробки вхідних даних для та видалення шуму.

– Можливість відображення в навчальній виборці всіх аспектів

піддослідного процесу. Наприклад, чи можливо відобразити в навчальній виборці сигнатури всіх типів аномальної поведінки або сигнатури всіх вірусів.

– Пропорційність навчальних прикладів, що відповідають різним аспектам піддослідного процесу. Наприклад, скільки навчальних прикладів відповідають аномальній поведінці типу А, а скільки прикладів – поведінці типу В.

2. Обмеження процесу навчання обумовлюються:

– Максимальним терміном навчання.

– Необхідністю представлення в навчальних даних очікуваного вихідного сигналу НМ. Таким чином, 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 на 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. В той же час неможливо навчити НМ класифікувати електронні листи відповідно інтересам конкретного користувача. Доцільність навчання в лабораторних умовах пояснюється потребами оптимального механізму створення та оновлення бази знань НМ.

– Вимогою до незмінності вихідного сигналу мережі для різних прикладів з однаковими параметрами.

3. На практиці вимоги до обчислювальних потужностей визначаються максимальною кількістю прикладів (обсяг пам'яті), яку може запам'ятати НМ для досягнення необхідної точності розпізнавання. В свою чергу точність розпізнавання характеризується величинами максимальної та середньої помилки НМ на даних які можуть виходити за межі множини навчальної вибірки. Відповідно виникає задача

екстраполяції результатів навчання НМ за межі навчальної вибірки прикладів.

4. Вимоги до вихідної інформації НМ вказують на те, в якому вигляді має бути представлена ця інформація. Наприклад, при розпізнавані вірусів може виникнути необхідність не тільки визначення ситуації типу “несправність в програмному забезпеченні”, але й розрахунку ймовірності цієї ситуації або графічного відображення таких ситуацій на площину, що дозволить провести остаточну класифікацію користувачеві. 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 СЗІ, обсягу та складності програмної реалізації.

6. Сфера застосування визначає системи, в яких буде використовуватись НМ. На сьогодні достатньо дослідженим є використання НМ для розпізнавання образів, проведення оптимізації та аналізу тексту. 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. Крім того, сфера застосування визначається пристосованістю мережі до автономного функціонування. Для цього в архітектурі НМ повинно бути передбачено можливість повної автоматизації процесу донавчання на експлуатації.

1.4. Нейромережеві моделі та методи оцінювання параметрів безпеки інформаційних систем

Базою для аналізу сучасних нейромережевих моделей та методів, що застосовуються в СЗІ стали роботи [1, 4, 7, 15, 19, 37, 48, 58, 59, 86, 87, 100, 111, 134, 146, 199, 212, 247-250, 252]. В більшості проаналізованих робіт є певна невідповідність термінологічного аспекту описаної розробки: нейромережевий метод, модель, система, технологія, засіб. Як правило, наводиться комплексний опис розробки, хоча назва роботи вказує, наприклад, на створення НММ. Тому аналіз цих робіт проведено з єдиних позицій визначення основних характеристик

НММ та методів. Наведемо отримані дані.

Методи простої та семантичної класифікації мережевих атак. Методи розроблено в межах нейромережевої технології виявлення мережевих комп'ютерних атак за допомогою програмного комплексу «Snort», описаної в роботі [37]. Технологія передбачає застосування двох нейромережевих методів виявлення атак – **простої класифікації (ПСК)** та **семантичної класифікації (ССК)**. В якості вхідних параметрів використовуються параметри мережевих пакетів транспортного рівня стеку протоколів TCP/IP. В методі ПСК використано БШП з 10 вхідними нейронами та 2 нейронами у вихідному шарі. Для оптимізації кількості схованих нейронів пропонується застосування конструктивних алгоритмів. Наведено вираз для розрахунку корекції вагових коефіцієнтів нейронів вихідного шару:

$$\Delta w_{jk}(i) = -\eta(y_n(i) - f(x_i))\varphi'(v_n(i))y_n, \qquad (1.4)$$

де η – коефіцієнт швидкості навчання,

n – номер нейрону у вихідному шарі,

i – номер навчальної ітерації,

v_n – інформаційне поле, отримане на вході функції активації,

y_n – вихідний сигнал n-го вихідного нейрону,

φ' – похідна функції активації,

$f(x_i)$ – бажаний відгук і-го нейрону.

Зазначимо відсутність детального опису процесу оптимізації структури БШП. В методі ССК пропонується використання топографічної ТК, вибір якої обґрунтовується її невисокою ресурсоємністю. В обох методах передбачено обробку вхідних параметрів з метою зменшення кількості вхідних параметрів НМ.

Метод нейромержевої фільтрації спаму (НФС), наведений в роботі [249]. Доводиться оптимальність використання адитивних НМ. Вид НММ обрано з позицій максимізації точності розпізнавання, можливості автоматизації навчання та можливості представлення результатів в графічному вигляді. Тобто використано

процедуру багатокритеріальної оптимізації процесу визначення архітектури НМ. В якості вхідних параметрів НММ використано частоти зустрічі в спамі та в цільових електронних листах інформативних слів. Також запропонована процедура багатокритеріальної оптимізації параметрів НМ, в якій використано критерії максимізації обчислювальної потужності та мінімізації терміну навчання.

Метод визначення фрагментів програмного коду (ВФПК), описаний в роботі [24]. Метод застосовується для визначення переліку та оцінки значень вхідних параметрів НМ, що використовуються в системах детектування шкідливого програмного забезпечення. Також в роботі [248] наведено опис та результати експериментів по розпізнаванню ШПЗ, проведених за допомогою БШП. Аналіз наведених результатів підтверджує перспективність запропонованого методу. Можна зробити висновок про використання в методі процедури попередньої обробки вхідних параметрів НМ, яка підвищує їх інформативність.

Нейромережева системи виявлення вторгнень (НСВВ), описана в роботі [250]. Система орієнтована на використання БШП для розпізнавання мережевих атак. Наведено результати експериментів, що підтверджують ефективність системи при розпізнаванні атак, сигнатури яких представлені в базі KDD-99. Вибір типу НМ обґрунтовано з позицій максимальної обчислювальної потужності. Також проведена однокритеріальна оптимізація архітектури БШП.

Нейромережевих підхід виявлення SQL-ін'єкцій (НПВІ) представлений роботі [252]. Запропоновано розглядати проблему визначення зловмисних SQL-запитів у вигляді проблеми прогнозування часових рядів. Відповідно вказаній пропозиції пропонується використати рекурентні НМ типу Джордана та Елмана. Тобто тип НМ обрано відповідно критерію апробованості в задачах прогнозування часових рядів. Також наведено процедуру попередньої обробки вхідних параметрів та процедуру однокритеріальної оптимізації структури НМ. Використано критерій максимізації обчислювальної потужності. Наведені результати експериментальних досліджень, котрі були проведені на основі даних порталу Php-Nuke, підтверджують перспективність запропонованого підходу.

Бінарний нейромережевих метод (БНМ), описаний в роботі [111]. Метод

застосовується для вирішення задачі виявлення мережевих атак. В основі методу лежить спеціальна бінарна нейронна мережа, яка має дві важливі властивості. По-перше, модель пристосована для вирішення завдань, у яких вхідна інформація має складну, багатозв'язкову і навіть фрактальну структуру. По-друге, метод навчання моделі є прямою обчислювальною процедурою і не зводиться до пошуку глобального екстремуму складної нелінійної функції, що не накладає ніяких принципових обмежень на розмірність завдання. Таким чином, в методі передбачено вибір виду НММ по критерію апробованості в задачах певного типу та по критерію мінімізації тривалості навчання. На жаль, в роботі відсутні експериментальні дані, що ускладнює порівняльний аналіз. В методі не передбачено проводити оптимізацію структури НМ, застосування та процедури обробки вхідних даних.

Метод виділення мережевих атак із типового мережевого трафіку (ВМА), описаний в роботі [100]. Метод застосовується для розпізнавання мережевих атак. Запропоновано застосування БШП з 2 СШН. ВШ такого БШП складається із 9 нейронів, а ШВ – із 1 нейрону. Зазначено, що вибір БШП з такою структурою пояснюється вимогами гнучкості та функціональності. Тобто використано багатокритеріальну оптимізацію структури НМ. Вказано на попередню обробку статистики, що використовувалась для навчальної та тестової вибірки.

Спосіб виявлення DDoS-атак (СВДА), наведений в роботі [146]. Запропоновано використання нечітких НМ. Пропозиція ґрунтується на перспективності НМ такого типу. Акцент ставить на розпізнаванні DDoS-атаки типу SYN Flood. Для формалізації знань експертів про DDoS-атаки було створено 5 лінгвістичних змінних, кожна з яких характеризує одну з компонент вектора параметрів мережевого трафіку, що використовується для формування вхідних параметрів НМ. До вказаних лінгвістичних змінних відносяться: X_1 – час отримання пакетів, X_2 – процент пакетів з різних зовнішніх ір-адрес, X_3 – процент пакетів з різних портів, X_4 – процент пакетів з пошкодженими заголовками, S – степінь впевненості. Розроблено предикатні правила виду: Якщо X_1 = «великий» $\rightarrow Y \rightarrow$

«висока». Запропоновано представити нечіткий класифікатор у вигляді НМ з прямим розповсюдженням сигналу, що навчається за допомогою модифікованого алгоритму зворотнього розповсюдження помилки. Модифікація полягає у пристосуванні класичного алгоритму до нечітких нейронів «І» та «АБО». Таким чином, основною відмінністю СВДА є можливість застосування для навчання НМ експертних знань.

Метод використання нейронної мережі гібридної структури типу CounterPropagation (НМГС), описаний в роботах [19, 199]. Метод призначено для виявлення мережевих атак на веб-сервер. Особливістю мережі CounterPropagation є комбінація ТК з БШП. Вхідними даними методу є параметри мережевого трафіку, що передається по протоколам IP, TCP, HTTP, HTTPS, CGI, SQLNet. В методі передбачена процедура попередньої обробки вхідних параметрів НМ за рахунок представлення їх у вигляді графічних образів (піфограм), котрі використовуються в когнітивній графіці. Метою попередньої обробки є мінімізація розмірності вхідних даних. Графічне представлення визначило необхідність застосування в методі шару Кохонена. Використання персептронного шару обґрунтоване з позицій обчислювальної ефективності. Таким чином, в методі передбачено багатокритеріальну оптимізацію виду та однокритеріальну оптимізацію параметрів НММ. Також в методі застосована процедура оптимізації параметрів навчання НМ, яка дозволяє до 10 разів зменшити величину помилки розпізнавання атак.

Метод побудови сукупного класифікатора трафіку (ПСКТ), запропонований в роботі [86]. Метод призначений для ієрархічної класифікації комп'ютерних атак на інформаційно-телекомунікаційні мережі. Особливістю ПСКТ є використання математичного методу головних компонент для стиснення статистичних даних, що використовуються в якості навчальної вибірки НМ. В методі використано об'єднання з 22 нейромережевих детекторів, кожен із яких навчений розпізнавати певний тип атаки, наведений в базі даних KDD-99. Детектор представляє собою трьохшарову НМ з 12 вхідними нейронами та 2 вихідними нейронами, один із яких відповідає за наявність, а другий за відсутність атаки. В якості СШН використано шар Кохонена. Зазначимо, що обґрунтування архітектури

та параметрів нейромережевого детектора не наведено. При виявленні детектором атаки вихід першого вихідного нейрону дорівнює 1. Для унеможливлення ситуації, коли декілька детекторів одночасно сигналізують про власний тип атаки, на другий вихід кожного з них передається мінімальна евклідова відстань між вхідним образом (вхідними параметрами - x_i) і ваговими коефіцієнтами схованих нейронів ($w_{i,j}$):

$$E_j = \min_i \sqrt{(x_1 - w_{1,j})^2 + ...(x_{12} - w_{12,j})^2} \ . \qquad (1.5)$$

Надалі класифікується та атака, детектор якої має мінімальну евклідову відстань. В ПСКТ в неявному вигляді передбачено оптимізацію навчання та функціонування нейромережевого детектора.

Нейромережевий підхід до виявлення мережевих атак (ПВМА) на ІС, наведено в роботі [87]. Акцент ставиться на розпізнавання атак, сигнатури яких представлені в БД KDD-99. Відповідно даних цієї БД, кількість вхідних параметрів – 41. Запропоновано використовувати критерій вибору оптимального виду НММ у вигляді мінімуму обсягу навчальної вибірки. Шляхом аналізу літературних джерел визначено, що до допустимих типів відносяться: ТК, БШП з одним схованим шаром нейронів та РБФ. Зазначено, що для ТК мінімальний обсяг навчальної вибірки (L) повинен в 2 рази перевищувати кількість вхідних нейронів (n). Тобто $L \geq 2n$. Для БШП та РБФ обсяг навчальної вибірки розраховується так $L \approx W / \varepsilon$, де W – кількість синаптичних зв'язків, ε – допустима помилка навчання. Надалі в [12] зроблена спроба визначити оптимальну структуру БШП. Заявлено, що визначена експериментальним шляхом кількість схованих нейронів дорівнює $m = 10$. При цьому кількість вихідних нейронів дорівнює 2. Відповідно, необхідний обсяг навчальної вибірки ТК складає $L = 82$ приклади, а для БШП та РБФ при $\varepsilon = 0{,}1$, $L = (m(n+3)+2)/\varepsilon = 4420$. Тому оптимальним типом нейромережевої моделі обрано ТК. Зазначимо, що правильність розрахованих величин викликає сумніви, адже, відповідно теорії НМ [17], при заданій точності навчання кількість схованих нейронів БШП безпосередньо залежить від величини навальної вибірки. Надалі в

[12] проводиться оптимізація структури ТК. Неявно використано критерій максимізації точності навчання. Також використано аналогічна [12], процедура попередньої обробки вхідних параметрів.

Адаптивна система виявлення атак (АСВА), описана в роботі [148]. Система призначена для розпізнавання мережевих атак та базується на спільній роботі ТК і БШП, що виконують завдання кластеризації і класифікації даних. Виявлення атак, котре проводиться в декілька етапів, стало можливим завдяки тому, що в базу даних експертної системи вносилася інформація про зміни в поведінці конкретного об'єкта на протязі деякого відрізка часу. Доводиться, що оптимізація архітектури дозволить підвищити точність та оперативність розпізнавання. В якості вхідних даних використано параметри мережевого трафіку по протоколу TCP. Для обробки вхідних даних використано метод ковзаючого часового вікна. ТК використана для попередньої обробки даних, що поступають на вхід БШП з метою їх стиснення та підвищення інформативності. Наведено математичний вираз для розрахунку частоти визначення нейрону в позиції (*i,j*) в якості нейрону-переможця:

$$\beta_{i,j} = f_{i,j} + \sum_{x=1}^{r}\left(\frac{f_{i-x,j} + f_{i,j-x} + f_{i+x,j} + f_{i,j+x}}{1+x}\right), \qquad (1.6)$$

де $f_{i,j}$ – кількість разів коли нейрон в позиції (*i,j*) був нейроном-переможцем,

r – відстань між центрами кластерів,

x – довжина вхідного вектора.

Надалі ця частота використовується для визначення центрів та границь кластерів. Структура БШП оптимізована з точки зору відповідності обсягу контрольованих ресурсів.

Нейромережева технологія виявлення та класифікації мережевих атак (ВКМА), описана в роботі [249]. В технології запропоновано використання трьохшарової НМ, що навчається методом зворотнього поширення помилки. При цьому для розпізнавання кожного виду мережевої атаки застосовується окрема НМ. Як вхідні параметри використовуються параметри мережевого трафіку по стеку

протоколів TCP/IP. 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. Описи підходів до оптимізації виду та параметрів НММ відсутні.

Система виявлення аномальної поведінки обчислювальних процесів (ВАОП), розроблена в роботі [7]. Система призначена для виявлення атак на компоненти інформаційної системи, які функціонують на базі мікроядерних операційних систем. Детально розроблено методику збору та підготовки вхідних параметрів для НМ. Пропонується використання ТК та БШП. Опису процедури оптимізації виду та параметрів НММ не наведено.

Модель кібернейрону (МКН), розроблена та описана в роботі [48]. Модель пропонується використовувати для розпізнавання комп'ютерних вірусів. Основною відмінністю моделі кібернейрону є відсутність функції активації, замість якої використовується таблиця підстановки, а основною перевагою – потенційно висока обчислювальна потужність. Розроблені алгоритми навчання кібернейрону. В якості вхідних параметрів використовуються або фрагменти піддослідного файлу, або хеш-коди вказаних фрагментів. Визначення вказаних фрагментів пропонується реалізувати методом ковзаючого вікна. Завданням НМ являється розпізнавання чистих та заражених фрагментів. Слід зазначити, що модель кібернейрона з'явилась відносно недавно, являється практично не апробованою, а використання табличної активаційної функції теоретично малообґрунтоване. Відповідно, застосування кібернейрону в сфері захисту інформації потребує серйозного доопрацювання.

Метод розпізнавання аномалій мережевого трафіку (РАМТ), розроблений в роботі [1]. Методом передбачене використання НМ типу БШП. В якості вхідних даних НМ використано параметри заголовків IP-дейтаграм. Вибір архітектури НМ базується на твердженні про високі апроксимаційні можливості БШП. БШП складається із трьох шарів нейронів. Кількість нейронів ВШ – 18, що дорівнює кількості параметрів заголовку IP-дейтаграми. Кількість нейронів у ШВ 2. Вихід нейрону №1 відповідає за наявність аномалії, а вихід нейрону №2 за безпечний стан

мережевого трафіку. Наведені вирази для розрахунку кількості нейронів у СШН. Таким чином, методом передбачено оптимізацію параметрів архітектури НМ. Для спрощення створення репрезентативної вибірки розроблено метод уточнюючих сигнатур, суть якого полягає у введенні додаткових штучно створених сигнатур, що описують апріорно аномальний трафік. Таким чином, в методі в неявному вигляді можливо використати експертні дані про мережеві атаки.

Нейромережева штучна імунна система (НШІС), описана в роботі [7, 247]. НШІС призначена для розпізнавання в сканованих файлах ШПЗ. Використано НМ типу ТК. Вибір тину НМ обґрунтовано по критерію мінімізації допустимого обсягу навчальної вибірки (L), який для ТК залежить тільки від кількості нейронів СШН (m): $L \geq 2m$. В свою чергу $m = p + r$, де p – кількість прикладів безпечних програм в навчальній вибірці, а r – кількість прикладів ШПЗ. Процедури попередньої обробки вхідних параметрів та оптимізації процесу навчання не передбачені.

Модель ТК для розпізнавання комп'ютерних вірусів (МТК), розроблена в роботі [4]. Модель призначена для використання в антивірусних сканерах. Передбачено блок попередньої обробки вхідних параметрів. Вибір типу моделі реалізовано шляхом порівняльних числових експериментів. В якості критерію порівняння використано термін навчання. Оптимізація параметрів та процедури навчання нейромережевої моделі не проводилась.

Метод виявлення несанкціонованого доступу до бази даних (ВНДБД) розроблено в роботі [62]. Крім виявлення атак, метод передбачає виявлення вразливостей в БД. Запропоновано використання ДШП. ВШ складається із 4 нейронів, а ШВ із 1. В якості вхідних даних використано: обсяг інформації, що завантажується в базу даних, кількість транзакцій за одну хвилину, кількість операцій модифікації за одну хвилину, ознаки звернень до словника. Попередня обробка вхідних параметрів полягає у їх ранжуванні та нормалізації.

Алгоритм перетворення параметрів трафіку (АППТ) описано в роботі [15]. Алгоритм призначений для отримання із мережевого трафіку вхідних даних для нейромережевої системи виявлення мережевих атак. В якості вхідної інформації зазначеного алгоритму використовуються параметри TCP-сесії. Перетворення

параметрів трафіку застосовується з метою зменшення кількості вхідних параметрів НМ і збільшення їх інформативності та реалізується за допомогою математичного апарату, що базується на методі головних компонент. В алгоритмі оптимізація виду та параметрів НММ не передбачена.

Нейромережева технологія виявлення мережевих атак (ТВМА) на інформаційні ресурси описана в [58, 59, 148]. В технології передбачено модуль стиснення вхідних даних, котрий базується на застосуванні нейромережевого аналогу методу головних компонент – рециркуляційної нейронної мережі з двома шарами нейронів. Шляхом числових експериментів доведено можливість використання запропонованої технології для виявлення мережевих атак, сигнатури яких представлено в базі даних KDD-99.

Базові характеристики проаналізованих нейромережевих методів та моделей наведено в табл. 1.1. Аналіз даних цієї таблиці вказує на те, що більшість відомих НМЗ призначені для розпізнавання мережевих атак. При цьому в якості базових видів НММ використовуються БШП та ТК.

Крім того, в результаті проведеного аналізу визначено, що забезпечення ефективності сучасних нейромережевих методів та моделей йде шляхом забезпечення в них певних можливостей, котрі характеризуються за допомогою наступних критеріїв: $\phi_{по}$ – попередня обробка вхідних параметрів, $\phi_{ота}$ – однокритеріальна оптимізація виду архітектури, $\phi_{бва}$ – багатокритеріальна оптимізація виду архітектури, $\phi_{опа}$ – однокритеріальна оптимізація параметрів архітектури, $\phi_{бпа}$ – багатокритеріальна оптимізація параметрів архітектури, $\phi_{омн}$ – оптимізація методу навчання, $\phi_{веп}$ – можливість використання експертних правил. Наведений перелік доповнено критеріями $\phi_{мна}$ та $\phi_{одв}$, які вказують на можливість застосування в методі класичних і перспективних видів НММ та на можливість принципової оцінки доцільності застосування НМ для вирішення поставленої задачі. Підґрунтям використання параметру $\phi_{мна}$ є наведене в роботах [17, 20] твердження про те, що в СЗІ, як і в більшості відомих застосувань, розвиток нейромережевих методів та моделей йде шляхом пристосування базових та перспективних нейромережевих архітектур до умов поставлених практичних задач. Підґрунтям

використання параметру $\phi_{одв}$ є об'єктивна необхідність чіткого окреслення області застосувань НМ в СЗІ.

Величини запропонованих критеріїв в першому наближенні можна оцінити так: критерій дорівнює -1, коли відповідна можливість в нейромережевому методі або моделі не забезпечується, 0 – коли забезпечується опосередковано і 1 – коли забезпечується безпосередньо.

Таблиця 1.1

Базові характеристики нейромережевих методів та моделей

№	Метод	Розпізнавання				Тип НМ								
		ШПЗ	Атак на БД	Спаму	Мереже-вих атак	БШП	КН	ТК	НМДЕ	АНМ	ННМ	БНМ	РНМ	Всі типи
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	ВФПК	+	-	-	-	+	-	-	-	-	-	-	-	-
2	МКН					-	+	-	-	-	-	-	-	-
3	МТК					-	-	+	-	-	-	-	-	-
4	НШІС													
5	НПВІ	-	+	-	-	-	-	-	+	-	-	-	-	-
6	ВНДБД					+	-	-	-	-	-	-	-	-
7	НФС	-	-	+	-	-	-	-	-	+	-	-	-	-
8	АППТ	-	-	-	+	-	-		-	-	-	-	-	+
9	ПСК					+	-		-	-	-	-	-	-
10	НСВВ													
11	ТВМА													
12	РАМТ													
13	ВМА													
14	ССК					-	-	+	-	-	-	-	-	-

Таблиця 1.1 (продовження)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
15	НМГС	-	-	-	+	+	-	+	-	-	-	-	-	-
16	ПСКТ													
17	ПВМА													
18	АСВА													
19	ВАОП													
20	СВДА					-	-		-	-	+	-	-	-
21	БНМ					-	-		-	-	-	+	-	-
22	ВКМА					-	-		-	-	-	-	+	-

Для проаналізованих випадків величини означених критеріїв наведено в табл. 1.2. При цьому для всіх проаналізованих методів $ф_{мна}$= $ф_{одв}$ =-1. Лише для АППТ $ф_{мна}$=1, а $ф_{одв}$=0.

Тобто в більшості із проаналізованих методів не можна використати всього переліку класичних та перспективних видів НММ і в жодному із методів (моделей) не передбачена оцінка принципової доцільності його застосування. Також зазначимо, що базовий перелік параметрів може бути в подальшому розширений.

Таблиця 1.2

Величини критеріїв, що характеризують нейромережеві методи та моделі

№	Модель, метод	Параметр								
		$ф_{по}$	$ф_{ота}$	$ф_{бва}$	$ф_{опа}$	$ф_{бпа}$	$ф_{омн}$	$ф_{веп}$	$ф_{мна}$	$ф_{одв}$
1	2	3	4	5	6	7	8	9	10	11
1	ВФПК	1	0	-1	0	-1	0	1	-1	-1
2	МКН	1	1	-1	-1	-1	-1	-1	-1	-1
3	МТК	1	1	-1	0	-1	-1	-1	-1	-1
4	НШІС	-1	1	-1	1	-1	-1	-1	-1	-1

Таблиця 1.2 (продовження)

1	2	3	4	5	6	7	8	9	10	11
5	НПВІ	1	1	0	0	-1	0	-1	-1	-1
6	ВНДБД	1	0	-1	0	-1	-1	-1	-1	-1
7	НФС	1	1	0	1	0	1	-1	-1	-1
8	АППТ	1	-1	-1	-1	-1	-1	-1	0	-1
9	ПСК	1	0	-1	0	-1	0	-1	-1	-1
10	НСВВ	0	0	-1	0	-1	0	-1	-1	-1
11	ТВМА	1	0	-1	0	-1	-1	-1	-1	-1
12	РАМТ	-1	1	-1	0	-1	-1	0	-1	-1
13	ВМА	0	0	-1	0	-1	-1	1	-1	-1
14	ВМА	1	0	-1	0	-1	0	-1	-1	-1
15	ВМА	1	1	0	0	-1	-1	-1	-1	-1
16	ПСКТ	1	0	-1	0	-1	0	-1	-1	-1
17	ПВМА	1	1	-1	0	-1	0	-1	-1	-1
18	АСВА	1	1	0	1	1	0	-1	-1	-1
19	ВАОП	1	-1	-1	-1	-1	-1	-1	-1	-1
20	СВДА	0	0	-1	0	-1	0	-1	-1	-1
21	БНМ	-1	0	-1	-1	-1	1	-1	-1	-1
22	ВКМА	1	-1	-1	-1	-1	-1	-1	-1	-1

Практична цінність даних табл. 1.2 полягає у окресленні недоліків та перспектив вдосконалення сучасних нейромережевих методів та моделей. Наприклад, величини $\phi_{по}$=0, $\phi_{бва}$=-1 свідчать про те, що до недоліків методу НСВВ можна віднести недостатню оптимізацію виду НММ. Це свідчить про можливість відповідного вдосконалення вказаного методу. При цьому жоден із розглянутих методів не передбачає повноцінної оптимізації НММ, відповідно умов поставленої задачі та повноцінного використання в такій моделі експертних правил.

Також в результаті проведеного аналізу показано, що в сучасних СВА в основному використовуються класичні види НММ, які в тій чи іншій мірі адаптовані до умов поставленої задачі.

1.5. Висновки до першого розділу

В результаті проведеного аналізу можливо зробити висновок про те, що важливим та актуальним напрямком підвищення ефективності систем розпізнавання кібератак на ресурси Інтернет-орієнтованих інформаційних систем є застосування нейромережевих моделей, методів та засобів оцінювання параметрів безпеки. Не дивлячись на певні досягнення в цій області, ефективному застосуванню таких засобів оцінювання заважають ряд недоліків. Основними із них являються:

– недостатня оперативність реагування на нові види кібератак;

– недостатня адаптація до варіативності умов застосування та функціонування при обмежених обчислювальних ресурсах;

– недостатня точність розпізнавання кібератак;

– недостатня взаємопов’язаність відомих нейромережевих підходів, моделей та методів оцінювання параметрів безпеки для виявлення кібератак;

– відсутність нейромережевих систем, що на основі комплексної нейромережевої методології дозволять вирішувати найбільш актуальні задачі розпізнавання кібератак на Інтернет-орієнтовані інформаційні системи.

Для виправлення цих недоліків визначено два напрямки наукових досліджень:

1. Розвиток теоретичного базису нейромережевого оцінювання параметрів безпеки.

2. Застосування розроблених теоретичних положень для послідовного створення нейромережевих моделей, методів та систем оцінювання параметрів безпеки.

Визначення напрямків розвитку теоретичного базису ґрунтується на

наступних передумовах:

– Ефективне використання нейромережевих засобів потребує розробки типових підходів до застосування нейромережевих моделей для розпізнавання різних видів кібератак.

– Відсутність оперативності розпізнавання нових типів кібератак в основному пов’язане з тривалим накопиченням статистичних даних, необхідних для навчання нейронної мережі. Для забезпечення оперативності можливо для навчання нейронної мережі використовувати експертні дані.

– Забезпечити пристосованість нейромережевих засобів до варіативності умов застосування можливо за рахунок оптимізації виду та параметрів нейромережевої моделі, що лежить в основі таких засобів.

– Для адаптації нейромережевих засобів до функціонування при обмежених обчислювальних ресурсах необхідно як оптимізувати вид та параметри нейромережевої моделі, так і апріорно оцінювати обсяг обчислювальних ресурсів для її реалізації.

– Використання нейромережевих засобів пов’язане з певним набором умов та обмежень, що визначаються умовами задачі оцінювання та характеристиками нейромережевої моделі. Тому необхідно провести як визначення принципової доцільності застосування нейромережевих засобів, так і оцінку ефективності їх розробки.

– Підвищити точність розпізнавання кібератак можливо за рахунок адаптації математичного забезпечення нейромережевих моделей до функціональних залежностей, що відповідають процесам розпізнавання. Крім того, для підвищення точності розпізнавання довготривалих кібератак доцільно використати марківський шаблон поведінки параметрів безпеки, який дозволяє частково нівелювати часову складову процесу розпізнавання.

– Використання нейронних мереж в високовідповідальних засобах розпізнавання кібератак потребує теоретичної верифікації нейромережевих моделей оцінювання параметрів безпеки,

Вирішення другого напрямку досліджень пов’язане з комплексною розробкою моделей, методів та систем, що базуються на запропонованих теоретичних рішеннях та враховують особливості сучасних видів кібератак.

Таким чином, сформульована в дисертаційній роботі проблема створення комплексної методології розробки широкодоступних ефективних нейромережевих засобів оцінювання параметрів безпеки Інтернет-орієнтованих інформаційних систем декомпозується на ряд наступних задач:

– аналіз сучасних нейромережевих засобів оцінки параметрів безпеки інформаційних систем;

– розвиток теоретичних положень побудови нейромережевих засобів оцінювання параметрів безпеки інформаційних систем, що дозволяють навчатись за допомогою експертних даних, зменшувати похибки класифікації, враховувати особливості сучасних видів кібератак, умови використання та верифікувати отримані рішення;

– побудова моделей, що враховують запропоновані теоретичні рішення та використовуються в нейромережевих засобах оцінки параметрів безпеки;

– розробка методів створення нейромережевих засобів оцінювання параметрів безпеки, що враховують запропоновані теоретичні рішення та побудовані моделі;

– розробка нейромережевих систем оцінки параметрів безпеки інформаційних систем, які дозволяють розпізнавати шкідливе програмне забезпечення, класифікувати листи електронної пошти та розпізнавати мережеві кібератаки.

РОЗДІЛ 2

РОЗВИТОК ТЕОРЕТИЧНИХ ПОЛОЖЕНЬ ПОБУДОВИ НЕЙРОМЕРЕЖЕВИХ ЗАСОБІВ ОЦІНЮВАННЯ ПАРАМЕТРІВ БЕЗПЕКИ ІНТЕРНЕТ-ОРІЄНТОВАНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

2.1. Базові підходи до оцінювання параметрів безпеки за допомогою нейромережевих засобів

Розпізнавання неочікуваних та поступових кібератак. Аналіз [247-250] вказує на те, що використання НМ для розпізнавання кібератак пов'язане з визначенням на базі аналізу вектору контрольованих подій ($\overline{\boldsymbol{Z}}$) оператора переходу ІС в різні стани захищеності (Sz):

$$Sz = Net(\overline{\boldsymbol{Z}}), \tag{2.1}$$

де Net – нейромережевий оператор.

Очевидно, що формування Net багато в чому залежить від вектору $\overline{\boldsymbol{Z}}$, який по своїй суті відображає характер кібератаки. Застосування методології технічної діагностики та контролю [71] до аналізу $\overline{\boldsymbol{Z}}$ дозволило, по аналогії з класифікацією відмов технічних систем, виділити два класи кібератак: поступові та неочікувані.

Визначення 1. Неочікуваною кібератакою будемо називати кібератаку, реалізація якої пов'язана з стрибкоподібним та неочікуваним, з точки зору СЗІ, виходом ПБ РІС за безпечні межі.

В більшості випадків НК обумовлюється використанням в процесі реалізації кібератаки схованих вразливостей системи захисту РІС або є результатом виходу з ладу СЗІ. В якості типового прикладу НК можна назвати «атаку нульового дня» [77].

Визначення 2. Поступовою є кібератака, яка виникає в результаті тривалої та очікуваної, з точки зору СЗІ, зміни ПБ до значень, які

перевищують безпечні межі.

Типовим прикладом ПК є DDos-атака на Веб-сервер, спрямована на вичерпання його обчислювальних ресурсів і реалізована за рахунок масових звернень. Слід зазначити, що в багатьох випадках РІС одночасно є ціллю як ПК, так і НК. З цієї точки зору доцільно розглянути комбіновані кібератаки.

Маючи на увазі, що, відповідно [31, 251], кібератака є випадковою подією для оцінки кількісних характеристик рівня захищеності, можливо використати статистичні моделі ПК та НК. В таких моделях основною статистичною характеристикою зовнішніх негативних впливів на безпеку ресурсів ІС є інтенсивність кібератак $\lambda(t)$, де t – термін експлуатації ресурсів ІС. За допомогою вказаних моделей визначають ймовірність забезпечення безпечного стану – $P(t)$, ймовірність успішної кібератаки на протязі заданого терміну експлуатації – $F(t)=1-P(t)$ та щільність розподілу часу безвідмовної роботи $-f(t)$. Основною ознакою моделі НК є постійна величина інтенсивності кібератак $\lambda(t)=const$. Завдяки цьому $P(t)=e^{-\lambda t}$, $F(t)=1-P(t)=1-e^{-\lambda t}$, $f(t)=\lambda e^{-\lambda t}$. Графічна інтерпретація статистичної моделі НК показана на рис. 2.1.

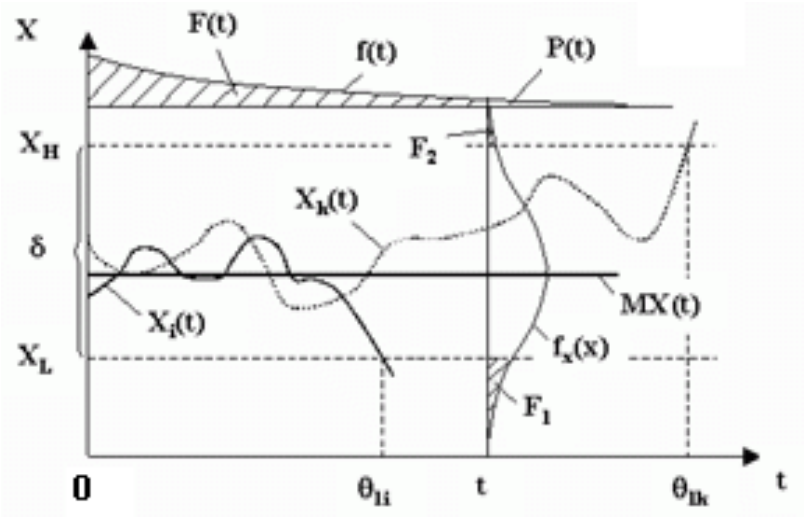


Рис. 2.1. Графічне відображення моделі неочікуваної кібератаки

Складовими частинами такої моделі є: X – ПБ; кібератака реалізується, якщо X виходить за межі встановленого допуску δ, обмеженого верхньою (X_H) та нижньою границями (X_L); $X(t)$ представляє собою стаціонарний випадковий процес, математичне сподівання ($MX(t)$), дисперсія ($DX(t)$) та функція щільності розподілу ($f_x(x,t)$) якого не залежать від терміну експлуатації. Тобто $MX(t)=const$, $DX(t)=const$, $f_x(x,t)=f_x(t)$. Зміна $X_i(t)$ та $X_k(t)$ реалізацій ПБ для *i*-го та *k*-го РІС викликана різними умовами експлуатації та різними негативними впливами. Моменти часу θ_{1i} та θ_{1k} виходу окремих реалізацій X за межі допуску свідчать про кібератаку на відповідний РІС.

Графічна інтерпретація статистичної моделі ПК показана на рис. 2.2. Основними її відмінностями від моделі НК є: кібератака реалізується, якщо величина X перевищує встановлену межу X_{max}; величина ПБ є довільною часовою функцією (на рис. 2.2 використана функція $X(t)=\bar{a}+\bar{\gamma}_x t$, де $\bar{a}$ – математичне сподівання величини X в нульовий момент часу, $\bar{\gamma}_x$ – математичне сподівання швидкості зміни X), математичне сподівання, дисперсія та функція щільності розподілу ПБ залежать від терміну експлуатації.

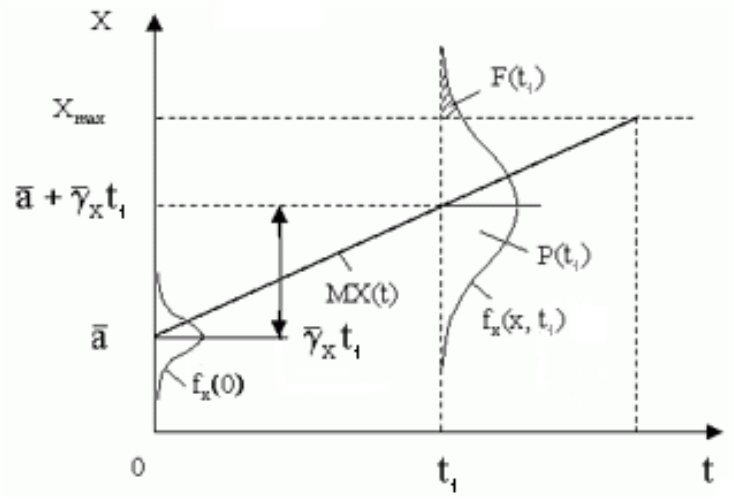


Рис. 2.2. Графічне відображення моделі поступової кібератаки

Зазначимо, що до спрощень показаної на рис. 2.2 моделі відносяться: лінійність функції $X(t)$ та нормальний закон розподілу $f_x(t)$. Також для прикладу на рис. 2.2 схематично показано розрахунок моменту часу t_1, коли $F(t_1) > 0$. Випадок зміни величини ймовірності забезпечення безпеки РІС під впливом комбінованої кібератаки показано на рис. 2.3, де індекси 1, 2, 3 відповідають НК, ПК та комбінованій кібератаці. Також зазначимо, що рис. 2.3 відповідає випадку, коли НК та ПК не залежні між собою. Тому ймовірність забезпечення безпеки при комбінованій кібератаці розраховувалась так: $P_3(t) = P_1(t)P_2(t)$.

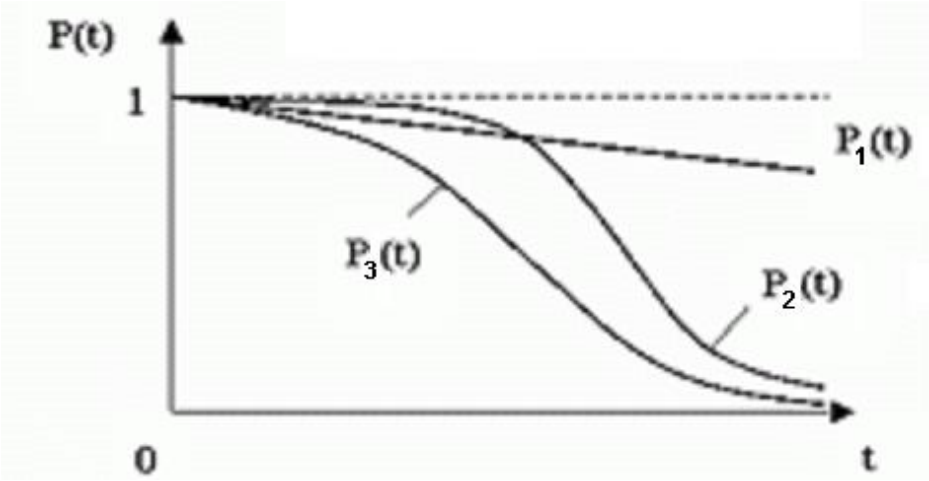


Рис. 2.3 Графік зміни ймовірності забезпечення безпеки при комбінованій кібератаці

Використання вказаних моделей дозволяє на основі знань динаміки ПБ розрахувати основні статистичні показники, пов'язані з кібератаками – $P(t), F(t), f(t)$. Крім того, аналіз цих моделей дозволяє запропонувати підходи до розпізнавання НК та ПК.

Підхід для розпізнавання неочікуваних кібератак. Оскільки НК характеризується деструктивним впливом, результативність якого не залежить від терміну експлуатації, то для її своєчасного виявлення потрібно реалізувати постійний контроль та оцінку ПБ. Виявити НК необхідно до того, коли ПБ вийдуть за межі попереджувального допуску. В якості бази

визначення ПБ доцільно використовувати параметри зовнішніх програмних запитів до ІС.

Рішення про наявність кібератаки приймається, якщо виявлено відповідність параметрів цих запитів ША або не відповідність ШНП:

$$Якщо\ \{X(t_k)\} \subset \left(\{X\}_A \cup \{D\}_A\right) \wedge / \vee \{X(t_k)\} \not\subset \left(\{X\}_N \cup \{D\}_N\right) \rightarrow A, \quad (2.2)$$

де $\{X(t_k)\}$ – множина значень ПБ в *k*-ий момент часу,

$\{X\}_A$ – значення ПБ, що відповідають ША,

$\{X\}_N$ – комбінація значень ПБ, що відповідають ШНП,

$\{D\}_A$ – множина попереджувальних допусків на ПБ для ША,

$\{D\}_N$ – множина попереджувальних допусків на ПБ для ШНП,

A – кібератака.

Наприклад, для виявлення скриптового ШПЗ слід провести аналіз використаних в ньому потенційно небезпечних програмних конструкцій. Вказаний аналіз необхідно провести до виконання скрипта. Аналіз відомих прикладів НК [60, 69] вказує на те, що труднощі їх виявлення в першу чергу пов'язані з багатоваріантним характером комбінацій ПБ, що вказують на наявність кібератаки. Цей факт значно ускладнює визначення граничних меж величин ПБ, які входять до множини $\{X\}_A$.

Підхід для розпізнавання поступових кібератак. Оскільки ПК є тривалим процесом, то для її своєчасне виявлення доцільно скористатись ШП, розрахованими на протязі деякого інтервалу часу. В якості ПБ можливо використовувати параметри зовнішніх запитів та функціональні параметри РІС. Правило класифікації ПК має вигляд:

$$Якщо\ \{X(t_k)\} \subset \left(\{X(t)_{t=t_k}\}_A \cup \{D\}_A\right) \wedge / \vee \{X(t_k)\} \not\subset \left(\{X(t)_{t=t_k}\}_N \cup \{D\}_N\right) \rightarrow A, \quad (2.3)$$

де $\{X(t)_{t=t_k}\}_A$ – значення ПБ для ША в *k*-ий момент часу,

$\{X(t)_{t=t_k}\}_N$ –значення ПБ для ШНП в *k*-ий момент часу.

Слід зазначити, що результати [124, 125] дозволяють представити ШП ПБ у вигляді одноперіодичного або багатоперіодичного динамічного ряду даних. В одноперіодичних ШП ПБ на протязі заданого терміну має характер одноперіодичної часової функції, а в багатоперіодичному ШП – характер багатоперіодичної часової функції.

Відповідно положень теорії динамічних рядів даних, багатоперіодичний ШП можна представити у вигляді суми одноперіодчних ШП [79, 125]. Графік одноперіодичного ШП показано на рис. 2.5, а графік багатоперіодичного ШП показано на рис. 2.6. ШП відповідає функції $X = f(t)$. На рис. 2.5 буквами А та В позначені перехідні (екстремальні) точки функції $X = f(t)$. При цьому букви A_2, A_4,…A_D відповідають максимумам, а букви B_1, B_2,…B_{D-1} – мінімумам цієї функції. Індекси 1,2,…D означають номер перехідної точки. На інтервалах типу $B_d A_{d+1}$ та $A_{d+1}B_{d+2}$ одноперіодичний ШП має стаціонарний характер. На інтервалах типу $B_d A_{d+1}$ функція $X = f(t)$ зростає, а інтервалах типу $A_{d+1}B_{d+2}$ – спадає.

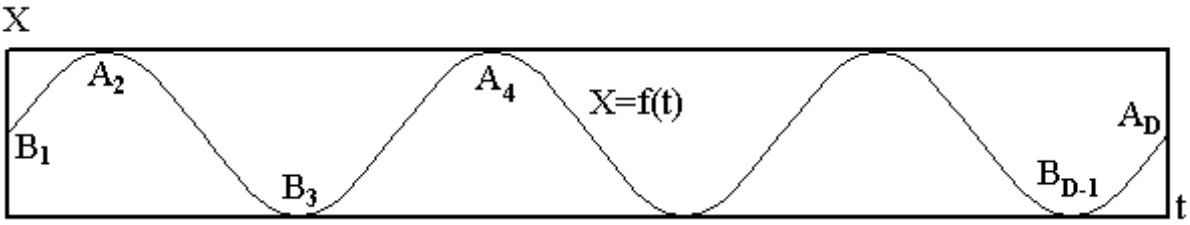


Рис. 2.4. Графік одноперіодичного шаблону поведінки

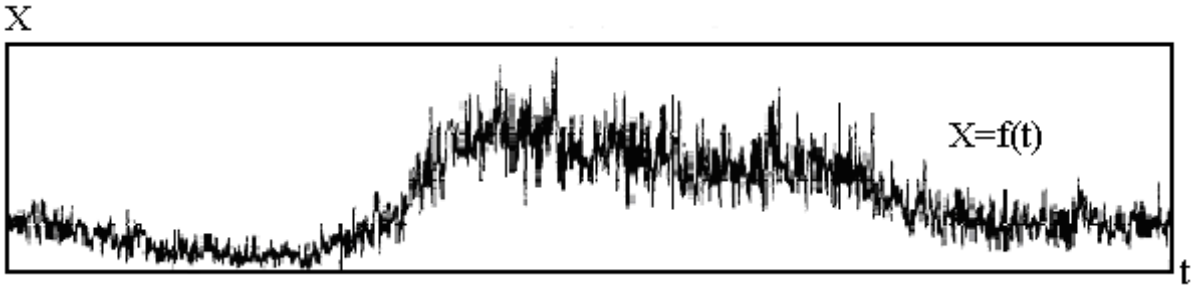


Рис. 2.5. Графік багатоперіодичного шаблону поведінки

Для випадку односторонньої області оцінки рівня захищеності та умові збільшення *i*-го ПБ в разі виникнення кібератаки (див. рис. 2.2) $X_{i,min} = 0$, а $X_i = 0$ відповідає найкращому стану захищеності. За рахунок цього (2.3) змінюється так:

$$Якщо\ X_i(t) \in [0, X_{i,max} - D],\ t = t_k \rightarrow A, \tag{2.4}$$

де D – величина допуску на X_i.

Графічна інтерпретація (2.4) показана на рис. 2.6. Прикладом використання цієї моделі може бути виявлення мережевої кібератаки з метою підбору парольних даних. Для її виявлення можливо встановити допуск на кількість неправильних вводів парольних даних з певної підмережі за встановлений проміжок часу.

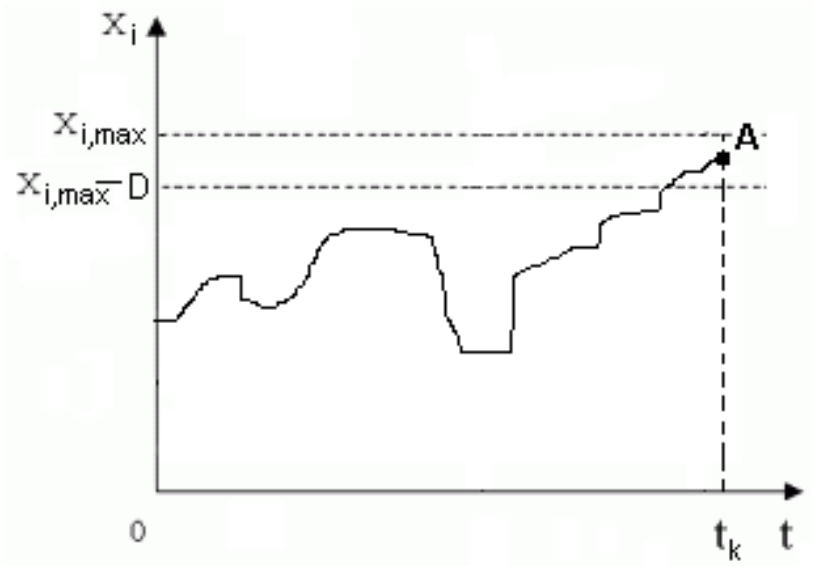


Рис. 2.6 Графічне відображення моделі оцінки ПБ при виявленні ПК

Зазначимо, що відповідно [71], величина D має випадковий характер та повинна враховувати динаміку ПБ:

$$D = F(X_i(t)), \tag{2.5}$$

Занадто велика величина D призведе до зменшення функціоналу ІС, а занадто мала – до збільшення ймовірності успіху кібератаки. Труднощі прийняття рішення про наявність/відсутність ПК пов'язані зі складним характером $X_i(t)$. Тому для розробки ефективних НМЗ виявлення різнотипних кібератак необхідно створення відповідної моделі оцінки ПБ.

Відповідно [71, 89], вказана модель повинна враховувати множину характеристик об'єкту захисту ($\boldsymbol{O}$), аналіз яких дозволяє визначити перелік ПБ. Базовими характеристиками об'єкту захисту є: структура (o_1), призначення (o_2), вразливості (o_3), функціональність (o_4), загрози (o_5). Тобто

$$\boldsymbol{O} = \{o_1, ...o_5\}. \tag{2.6}$$

Підхід до визначення оптимального виду нейромережевої моделі. Проведений в п. 1.3 аналіз дозволяє стверджувати, що розробка ефективних НМЗ йде шляхом пристосування визначених характеристик НММ до значимих умов задачі оцінювання ПБ. Це дозволяє запропонувати наступний підхід – оптимальним є той вид НММ, характеристики якого більш повно відповідають значимим умовам задачі оцінювання ПБ. В базовому варіанті множина значимих умов поділяється на визначені в п. 1.3 категорії, що характеризують навчальні дані, обмеження процесу навчання, обчислювальні потужності, вихідну інформацію, технічну реалізацію та сферу застосування НМЗ. Можливою інтерпретацією підходу є вираз

$$e(a_i) \to max,\ a_i \in A,\ i = 1,2,...,I, \tag{2.7}$$

де e – критерій оптимізації,

a_i – i-ий вид НММ,

A – множина допустимих видів НММ,

I – кількість допустимих видів НММ.

Очевидно, що використаний в (2.6) критерій оптимізації потребує деталізації з позицій врахування визначених характеристик виду НММ. Крім того, слід враховувати можливу близькість величини критерію оптимізації для різних видів НММ. Тому доцільно визначати множину оптимальних видів НММ. До цієї множини будуть входити НММ, в яких величина критерію оптимізації близька до максимальної. Близькість можливо оцінити за допомогою коефіцієнта відхилення – k_E.

Підхід до визначення принципової доцільності застосування нейромережевих засобів оцінювання параметрів безпеки. Аналіз теоретичних робіт [140] вказує на те, що принципова доцільність застосування НМЗ визначається можливістю в прийнятний термін визначити параметри НММ, які забезпечують достатню точність розпізнавання. При заданій архітектурі визначення параметрів НММ реалізується в процесі навчання. Для цього необхідно виконати наступні умови:

1. В піддослідному процесі визначити ПБ, 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.

Потенційно в якості вхідних параметрів НМ можуть бути використані всі зареєстровані ПБ ІС. Перед поданням в НМ вказані параметри потрібно відповідним чином закодувати та нормалізувати [49]. В найпростішому випадку вихід НМ повинен вказувати на наявність або відсутність кібератаки. В складніших випадках вихід НМ повинен вказувати на вид кібератаки. Відповідно, для формування навчальних прикладів необхідні статистичні дані про величини ПБ у випадку реалізації кібератаки та під час нормального функціонування ІС. Також слід враховувати наступні вимоги:

1. НМ навчена на прикладах 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 величин ПБ, 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 [8, 13]:

$$P_{min} \geq (10..20)N_x, \tag{2.8}$$

де P_{min} – мінімальна кількість навчальних прикладів,

N_x – кількість вхідних параметрів НМ.

4. Кількість навчальних прикладів має бути обмеженою.

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

Тому формування достатнього обсягу навчальних прикладів може викликати труднощі, пов'язані як з масштабуванням навчальних даних (вимоги 1, 2), так власне із збором достатньої кількості статистичної інформації (вимоги 3, 4, 5). Таким чином, термін визначення параметрів НММ можливо оцінити за допомогою формули:

$$T_f = T_n + t_n, \tag{2.9}$$

де T_n – термін формування навчальної вибірки,

t_n – термін навчання НМ.

Відповідно, НМЗ доцільно використовувати тільки в тому випадку, якщо розрахований за допомогою (2.8) термін визначення параметрів НММ менший від допустимого терміну розробки НМЗ. Тобто

$$T_f \leq T_a, \tag{2.10}$$

де T_a – допустимий термін створення системи розпізнавання.

Підхід до визначення ефективності розробки НМЗ оцінювання ПБ ІС. Аналіз сучасних НМЗ, що використовуються в СЗІ, дозволив визначити ряд базових критеріїв, кожен з яких дозволяє оцінити ефективність певного аспекту застосування вказаних засобів. При цьому, з точки зору теорії НМ [49, 101], критерії $\phi_{по}, \phi_{ота}, \phi_{бва}, \phi_{опа}, \phi_{бпа}, \phi_{омн}, \phi_{мна}$ вказують на потрібний обсяг обчислювальних ресурсів, які співвідносяться із кількістю обчислювальних операцій, необхідних для досягнення заданої точності класифікації. Критерії $\phi_{ота}, \phi_{бва}, \phi_{мна}$ вказують на можливість адаптації НМЗ до варіативності умов застосування. Критерії $\phi_{веп}$ та $\phi_{мна}$ співвідносяться із забезпеченням адаптації до кібератак, для яких відсутні статистичні дані, а $\phi_{одв}$ вказує на можливість принципової оцінки доцільності застосування НМЗ.

Таким чином, ефективність НМЗ оцінки ПБ можливо оцінити шляхом використання інтегральних критеріїв, що характеризують точність класифікації кібератак ($d_{ткк}$), можливість принципової оцінки доцільності застосування НМЗ ($d_{одв}$), адаптацію до нових видів кібератак ($d_{анв}$), пристосованість до варіативності умов застосування ($d_{вуз}$) та до функціонування при обмежених обчислювальних ресурсах ($d_{оор}$). При цьому

$$d_{ткк} = f\left(\phi_{ота}, \phi_{бва}, \phi_{омн}, \phi_{опа}, \phi_{бпа}, \phi_{мна}\right), \tag{2.11}$$

$$d_{одв} = f\left(\phi_{одв}\right), \tag{2.12}$$

$$d_{анв} = f\left(\phi_{веп}, \phi_{мна}\right), \tag{2.13}$$

$$d_{вуз} = f\left(\phi_{ота}, \phi_{бва}, \phi_{мна}\right), \tag{2.14}$$

$$d_{оор} = f\left(\phi_{по}, \phi_{ота}, \phi_{бва}, \phi_{омн}, \phi_{опа}, \phi_{бпа}, \phi_{мна}\right). \tag{2.15}$$

Інтеграція вказаних критеріїв дозволяє визначити інтегральну ефективність НМЗ. При цьому важливість критерію для конкретної задачі можливо врахувати за допомогою вагових коефіцієнтів, визначених на основі експертних даних. Ефективність НМЗ буде вважатись достатньою, якщо вона перевищує заданий апріорно мінімально допустимий рівень.

Підхід до класифікації подібних кібератак. Підхід полягає в тому, що *i*-та (Ka_i) та *k*-та (Ka_k) кібератаки вважаються подібними, якщо вони мають однаковий характер – неочікуваний (Ks) або поступовий (Kq), а приведена різниця ПБ (R_p), що використовуються для їх розпізнавання, не перевищує максимальну (R_{max}):

$$\left(Ka_i = Ks \wedge Ka_k = Ks\right) \vee \left(Ka_i = Kq \wedge Ka_k = Kq\right) \wedge \left(R_p \le R_{max}\right) \rightarrow Ka_i \sim Ka_k , \quad (2.16)$$

$$R_p = \left|R_i - R_k\right| / R , \quad (2.17)$$

$$R = max\left(R_i, R_k\right), \quad (2.18)$$

де R_i, R_k – кількість ПБ при розпізнаванні *i*-ої та *k*-ої кібератак.

Параметр R_{max} визначається апріорно на основі експертних даних.

Підхід до застосування продукційних правил при поданні експертних знань в нейромережеві засоби оцінки параметрів безпеки. Підхід базується на аналогії між експертними знаннями у вигляді продукційних правил та навчальними прикладами НМ.

В спрощеному випадку елементарне продукційне правило виду (2.19) можна вважати аналогом навчального прикладу НМ виду (2.20).

Якщо умова істина/хибна → (Висновок). (2.19)

$$X = a \rightarrow Y , \quad (2.20)$$

де X – вхідний параметр НМ,

a – значення вхідного параметру,

Y – очікуваний вихідний сигнал НМ.

В більш складному випадку типовий навчальний приклад НМ (2.20) – це комбінація продукційних правил виду (2.21, 2.22).

$$X_1 = a_1, ... X_N = a_N \rightarrow Y\ , \tag{2.21}$$

Якщо умова1 істина/хибна,… умоваN істина/хибна $\rightarrow$ *(Висновок)*, (2.22)

де X_N – n-ий вхідний параметр,

a_N – значення n-го вхідного параметру,

Y – очікуваний вихід НМ.

Стосовно оцінки ПБ для виявлення кібератак вираз (2.22) можливо трансформувати так:

$$\textit{Якщо}\ p_1 \in \left[P_1^{min}, P_1^{max}\right]_l \wedge ... \wedge p_K \in \left[P_K^{min}, P_K^{max}\right]_l \rightarrow Y_l\,, \tag{2.23}$$

де $p_1, ... p_K$ – ПБ,

$\left[P_1^{min}, P_1^{max}\right], ... \left[P_K^{min}, P_K^{max}\right]$ – задані діапазони величин ПБ,

K – кількість ПБ,

l – номер продукційного правила,

Y_l – результат продукційного правила.

При цьому достатньо відомі та апробовані види НММ, що можуть навчатись шляхом безпосереднього запам’ятовування представлених навчальних прикладів, тобто шляхом подання експертних знань у вигляді продукційних правил. Тому в такі види НММ можливо подати експертні знання у вигляді продукційних правил виду (2.23) про значення ПБ, що стосуються розпізнавання кібератак.

2.2. Критерії оптимізації виду нейромережевої моделі

Для формування критеріїв оптимізації виду НММ використано розроблений підхід до визначення оптимального виду НММ та результати дослідження можливостей застосування методів теорії НМ для оцінювання ПБ. Базовий перелік отриманих критеріїв показано в табл. 2.1. В подальшому перелік може бути розширений, наприклад, за рахунок деталізації певних критеріїв або врахування нових сфер застосування НМ.

Таблиця 2.1

Критерії оптимізації

№	Категорія	Пояснення критерію
1	2	3
$E_{1,1}$	Навчальні дані	Обмеженість кількості вхідних параметрів
$E_{1,2}$		Обмеженість навчальної вибірки
$E_{1,3}$		Допустимість шуму
$E_{1,4}$		Допустимість кореляції
$E_{1,5}$		Необхідність відображення всіх аспектів процесу
$E_{1,6}$		Необхідність пропорційного представлення прикладів
$E_{1,7}$		Можливість використання дискретних вхідних параметрів
$E_{1,8}$		Можливість використання неперервних вхідних параметрів
$E_{1,9}$		Можливість використання навчальної вибірки, обсяг якої менший за кількість вхідних параметрів
$E_{2,1}$	Процес навчання	Короткий термін навчання
$E_{2,2}$		Необхідність представлення в навчальних прикладах очікуваного виходу

Продовж. табл. 2.1

1	2	3
$E_{2,3}$	Процес навчання	Автоматизація навчання
$E_{2,4}$		Можливість донавчання
$E_{2,5}$		Якість навчання
$E_{2,6}$		Можливість навчання на експертних даних
$E_{2,7}$		Незмінність результатів
$E_{3,1}$	Обчислювальна потужність	Обсяг пам'яті
$E_{3,2}$		Екстраполяції результатів навчання
$E_{4,1}$	Вихідна інформація	Інтерпретованість виходу у вигляді ймовірності
$E_{4,2}$		Інтерпретованість виходу у графічному вигляді
$E_{4,3}$		Можливість вербалізації
$E_{5,1}$	Технічна реалізація	Швидкість прийняття рішення
$E_{5,2}$		Обсяг програмної реалізації
$E_{6,1}$	Сфера застосування	Розпізнавання образів
$E_{6,2}$		Аналіз тексту
$E_{6,3}$		Управління параметрами захисту
$E_{6,4}$		Пристосованість до автономного функціонування
$E_{6,5}$		Моделювання часових рядів
$E_{6,6}$		Аналіз зображень
$E_{6,7}$		Аналіз звуку
$E_{6,8}$		Розвідувальний аналіз даних

Результати п. 1.3 дозволили в першому наближені виставити оцінки відповідності основних видів НММ запропонованим критеріям. Вказані оцінки виставлені по трибальній шкалі та наведені в табл. 2.2 та табл. 2.3. Критерій E_i=1, якщо i-та характеристика задачі оцінювання ПБ повністю забезпечується у виді НММ, E_i=0 – якщо забезпечується частково і E_i=-1 – якщо не забезпечується.

Таблиця 2.2

Величини критеріїв оптимізації для НМ з прямим поширенням сигналу та АРТ

№	Вид нейромережевої моделі				
	БШП	Згорткові	РБФ	АРТ	Ймовірнісні
1	2	3	4	5	6
$E_{1,1}$	-1	-1	-1	-1	-1
$E_{1,2}$	-1	-1	-1	0	-1
$E_{1,3}$	1	1	0	-1	0
$E_{1,4}$	1	1	1	1	1
$E_{1,5}$	-1	-1	1	-1	1
$E_{1,6}$	-1	1	-1	-1	-1
$E_{1,7}$	1	1	1	1	1
$E_{1,8}$	1	1	1	1	1
$E_{1,9}$	-1	-1	1	1	1
$E_{2,1}$	-1	-1	0	1	1
$E_{2,2}$	1	1	1	-1	1
$E_{2,3}$	1	1	-1	1	1
$E_{2,4}$	0	0	1	1	1
$E_{2,5}$	1	1	0	1	1
$E_{2,6}$	-1	-1	-1	-1	1
$E_{2,7}$	1	1	1	1	1
$E_{3,1}$	1	1	-1	-1	-1
$E_{3,2}$	1	0	-1	-1	-1
$E_{4,1}$	0	-1	0	-1	1
$E_{4,2}$	-1	0	-1	-1	-1
$E_{4,3}$	1	-1	0	-1	0
$E_{5,1}$	1	1	1	1	1
$E_{5,2}$	-1	-1	1	0	-1

Продовж. табл. 2.2

1	2	3	4	5	6
$E_{6,1}$	1	0	1	1	1
$E_{6,2}$	-1	-1	-1	0	0
$E_{6,3}$	-1	-1	-1	-1	-1
$E_{6,4}$	0	-1	1	1	-1
$E_{6,5}$	1	-1	0	0	0
$E_{6,6}$	1	1	-1	-1	-1
$E_{6,7}$	1	0	-1	-1	-1
$E_{6,8}$	-1	-1	-1	-1	-1

Таблиця 2.3

Величини критеріїв оптимізації для рекурентних НМ, СНМ та ТК

№	Вид нейромережевої моделі				
	Елмена	Джордана	СНМ	АНМ	ТК
1	2	3	4	5	6
$E_{1,1}$	-1	-1	1	-1	-1
$E_{1,2}$	-1	-1	1	-1	-1
$E_{1,3}$	1	1	1	-1	1
$E_{1,4}$	1	1	1	-1	1
$E_{1,5}$	-1	-1	-1	0	1
$E_{1,6}$	1	1	-1	0	1
$E_{1,7}$	1	1	1	1	1
$E_{1,8}$	1	1	-1	0	1
$E_{1,9}$	-1	-1	1	1	1
$E_{2,1}$	-1	-1	0	1	1
$E_{2,2}$	1	1	-1	1	-1
$E_{2,3}$	-1	-1	1	0	0
$E_{2,4}$	0	0	1	0	1

Продовж. табл. 2.3

1	2	3	4	5	6
$E_{2,5}$	0	0	1	1	0
$E_{2,6}$	-1	-1	-1	-1	-1
$E_{2,7}$	1	1	1	0	0
$E_{3,1}$	1	1	0	0	-1
$E_{3,2}$	1	1	0	1	0
$E_{4,1}$	0	0	-1	0	0
$E_{4,2}$	-1	-1	-1	-1	1
$E_{4,3}$	1	1	-1	-1	-1
$E_{5,1}$	1	1	0	-1	1
$E_{5,2}$	-1	-1	-1	0	-1
$E_{6,1}$	0	0	0	1	1
$E_{6,2}$	-1	-1	1	-1	1
$E_{6,3}$	-1	-1	-1	1	1
$E_{6,4}$	-1	-1	1	-1	-1
$E_{6,5}$	1	1	-1	-1	-1
$E_{6,6}$	0	-1	-1	-1	-1
$E_{6,7}$	0	0	-1	-1	0
$E_{6,8}$	-1	-1	-1	1	1

З урахуванням трибальної числової оцінки (2.6) модифікується так:

$$E_{\Sigma} = \sum_{k=1}^{K} E_k(a_i) \to max,\ a_i \in \boldsymbol{A},\ i = 1,2,...,7\ , \qquad 2.24)$$

де E_{Σ} – інтегральний критерій оптимізації виду НММ,

$\boldsymbol{A}$ – множина допустимих видів НММ.

Відповідно результатів п.1.3., компоненти $\boldsymbol{A}$ визначаються так:

$$\boldsymbol{A} = \{БШП, РБФ, PNN, ТК, АРТ, АНМ, СНМ\}. \quad (2.25)$$

Для конкретної задачі оцінювання ПБ вагомість критеріїв оптимізації можливо врахувати, ввівши в (2.22) відповідні вагові коефіцієнти:

$$E_{\Sigma} = \sum_{k=1}^{K} \left(r_k \times E_k(a_i)\right) \to max,\ a_i \in \boldsymbol{A},\ i = 1,2,...,7, \quad (2.26)$$

де r_k – ваговий коефіцієнт k-ого критерію оптимізації.

2.3. Вдосконалення математичного забезпечення процесу навчання багатошарового персептрону

Проведені дослідження вказують на те, що можливою причиною великої відносної помилки навчання в області мінімальних значень вихідних параметрів БШП є запис критерію оцінки якості навчання у вигляді квадратичного функціонала виду:

$$\varepsilon(W) \to min,\ \varepsilon^2(W) = \left(y_i - y_i^r\right)^2, \quad (2.27)$$

де y_i, y_i^r – очікуваний та реальний вихідний сигнал i-го нейрону БШП,

W – матриця вагових коефіцієнтів синаптичних зв'язків.

Можливим шляхом зменшення відносної помилки є використання функціоналу відносної квадратичної помилки виду:

$$\bar{\varepsilon}_i^2(W) = \left(\varepsilon_i(W)/y_i\right)^2 = \left(y_i - y_i^r/y_i\right)^2, y_i \neq 0. \quad (2.28)$$

де $\bar{\varepsilon}_i$ – приведена помилка навчання i-го вихідного нейрону,

y_i – очікуваний сигнал i-го вихідного нейрону,

y_i^r – реальний сигнал *i*-го вихідного нейрону.

Зазначимо, що умова $y_i \neq 0$ визначається, виходячи із використання логістичної сигмоїдальної функції активації штучного нейрону:

$$y^r(z) = 1/1 + e^{-\alpha \cdot z} . \tag{2.29}$$

де α – деякий коефіцієнт,

z – зважена сума вхідних сигналів для нейрону.

Модифікація (2.28) призводить до певних змін у математичній моделі класичного алгоритму зворотнього поширення помилки. Визначимо ці зміни для БШП з довільною кількістю схованих шарів. Будемо базуватись на описі алгоритму навчання, наведеному в [4]. При цьому

$$\frac{d}{dz} y^r(z) = \alpha y^r(z)\left(1 - y^r(z)\right). \tag{2.30}$$

Розглянемо обчислення вагових коефіцієнтів зв'язків *i*-го нейрону вихідного шару. Використання приведеного квадратичного критерію якості навчання (2.28) та використаний в [210] постулат про можливість розгляду складових (2.28-2.30) у вигляді неперервних величин дозволяє записати математичне забезпечення градієнтного алгоритму корекції вагових коефіцієнтів в режимі навчання "on-line" в наступному вигляді:

$$\Delta w_{s,i}^{(j+1)} = -\gamma_{s,i}^{(j+1)} \frac{\partial \bar{\varepsilon}_i^2}{\partial w_{s,i}^{(j+1)}} . \tag{2.31}$$

де $\gamma_{s,i}^{(j+1)}$ – коефіцієнт швидкості навчання зв'язку між *i*-им нейроном вихідного шару та *s*-им нейроном попереднього шару,

$\Delta w_{s,i}^{(j+1)}$ – величина корекції зв'язку між *i*-им нейроном вихідного

шару та *s*-им нейроном попереднього шару,

$\bar{\varepsilon}_i$ – приведена помилка навчання *i*-ого нейрону вихідного шару.

Розрахуємо частинну похідну

$$\frac{\partial \bar{\varepsilon}_i^2}{\partial w_{s,i}^{(j+1)}} = \frac{\partial \bar{\varepsilon}_i^2}{\partial y_i^r} \frac{\partial y_i^r}{\partial z_i} \frac{\partial z_i}{\partial w_{s,i}^{(j+1)}}. \qquad 2.32)$$

де z_i – зважена сума вхідних сигналів для *i*-го вихідного нейрону.

Зазначимо, що z_i розраховується так:

$$z_i = \sum_{s=1}^{S} (w_{s,i}^{(j+1)} y_s^{(j)}). \qquad 2.33)$$

де S – кількість нейронів в передостанньому схованому шарі нейронів,

$y_s^{(j)}$ – вихідний сигнал *s*-го нейрону останнього схованого шару.

Враховуючи (2.29-2.31, 2.33), визначимо множники добутку (2.32)

$$\frac{\partial \bar{\varepsilon}_i^2}{\partial y_i^r} = \frac{\partial \left(\frac{y_i - y_i^r}{y_i} \right)^2}{\partial y_i^r} = \frac{1}{y_i^2} \frac{\partial \left(y_i - y_i^r \right)^2}{\partial y_i^r} = -\frac{2\left(y_i - y_i^r \right)}{y_i^2} = -\frac{2\varepsilon_i}{y_i^2}. \qquad (2.34)$$

$$\frac{\partial y_i^r}{\partial z_i} = \alpha y_i^r \left(1 - y_i^r \right). \qquad (2.35)$$

$$\frac{\partial z_i}{\partial w_{s,i}^{(j+1)}} = y_s^{(j)}. \qquad (2.36)$$

де $y_s^{(j)}$ – вихід *s*-го нейрону останнього схованого шару, пов’язаного з *i*-им нейроном вихідного шару.

Підставивши (2.34-2.36) в (2.32) отримаємо

$$\frac{\partial \bar{\varepsilon}_i^2}{\partial w_{s,i}^{(j+1)}} = -\frac{2\left(y_i - y_i^r\right)}{y_i^2} \times \alpha y_i^r \left(1 - y_i^r\right) \times y_s^{(j)}. \qquad 2.37)$$

Кінцевий вираз для корекції зв'язку між *i*-им нейроном вихідного (*j*+1)-го шару та *s*-им нейроном попереднього шару виглядає так:

$$\Delta w_{s,i}^{(j+1)} = \frac{2\alpha\gamma_{s,i}^{(j+1)} y_i^r \left(1 - y_i^r\right)\left(y_i - y_i^r\right) y_s^{(j)}}{y_i^2}. \qquad (2.38)$$

По аналогії з [140], використаємо позначення локальної помилки синаптичного зв'язку між *i*-им нейроном вихідного (*j*+1)-го шару та *s*-им нейроном попереднього шару у вигляді $\delta_{s,i}^{(j+1)}$. З урахуванням логістичної функції активації та функціоналу приведеної квадратичної помилки для зв'язку *i*-го нейрону вихідного шару з *s*-им нейроном попереднього шару вказана локальна помилка розраховується за допомогою наступного виразу:

$$\delta_{s,i}^{(j+1)} = \frac{y_i^r \left(1 - y_i^r\right)\left(y_i - y_i^r\right)}{y_i^2}. \qquad (2.39)$$

Використання $\delta_{s,i}^{(j+1)}$ дозволяє записати (2.31) у наступному вигляді:

$$\Delta w_{s,i}^{(j+1)} = 2\alpha\gamma_{s,i}^{(j+1)} \delta_{s,i}^{(j+1)} y_s^{(j)}. \qquad (2.40)$$

Розглянемо математичний апарат визначення величини корекції зв'язку між *s*-им нейроном останнього *j*-го схованого шару та *k*-им нейроном попереднього шару ($\Delta w_{k,s}^{(j)}$). Використовуючи приведений функціонал

помилки, величину корекції вказаного зв'язку запишемо так:

$$\Delta w_{k,s}^{(j)} = -\gamma_{k,s}^{(j)} \frac{\partial \bar{\varepsilon}_s}{\partial w_{k,s}^{(j)}}. \tag{2.41}$$

де $\bar{\varepsilon}_s$ – відносна помилка вихідного сигналу s-ого нейрону j-го схованого шару,

$\gamma_{k,s}^{(j)}$ – параметр швидкості навчання зв'язку між s-им нейроном j-го схованого шару та k-им нейроном попереднього (j-1)-го шару.

Обчислимо частинну похідну:

$$\frac{\partial \bar{\varepsilon}_s}{\partial w_{k,s}^{(j)}} = \sum_{i=1}^{M} \frac{\partial \bar{\varepsilon}_i^2}{\partial w_{k,s}^{(j)}} = \sum_{i=1}^{M} \frac{\partial \left(\frac{y_i - y_i^r}{y_i} \right)^2}{\partial w_{k,s}^{(j)}} = \sum_{i=1}^{M} \frac{1}{y_i^2} \frac{\partial \varepsilon_i^2}{\partial y_i^r} \frac{\partial y_i^r}{\partial z_i} \frac{\partial z_i}{\partial y_s^{(j)}} \frac{\partial y_s^{(j)}}{\partial z_s^{(j)}} \frac{\partial z_s^{(j)}}{\partial w_{k,s}^{(j)}}. \tag{2.42}$$

де M – кількість нейронів у вихідному шарі,

$\bar{\varepsilon}_i$ – приведена помилка на i-му виході БШП,

$z_s^{(j)}$ – зважена сума вхідних сигналів для s-го нейрону в j-му шарі,

Запишемо вирази для визначення множників добутку (2.40)

$$\frac{\partial \varepsilon_i^2}{\partial y_i^r} = \frac{-2\left(y_i - y_i^r\right)}{y_i^2}. \tag{2.43}$$

$$\frac{\partial y_i^r}{\partial z_i} = \alpha y_i^r \left(1 - y_i^r\right). \tag{2.44}$$

$$\frac{\partial z_i}{\partial y_s^{(j)}} = w_{s,i}^{(j+1)}. \tag{2.45}$$

$$\frac{\partial y_s^{(j)}}{\partial z_s^{(j)}} = \alpha y_s^{(j)}\left(1 - y_s^{(j)}\right). \tag{2.46}$$

$$\frac{\partial z_s^{(j)}}{\partial w_{k,s}^{(j)}} = y_k^{(j-1)}. \tag{2.47}$$

де $y_k^{(j-1)}$ – вихідний сигнал k-го нейрону (j-1)-го шару.

Підставивши (2.43-2.47) в (2.42) отримаємо:

$$\Delta w_{k,s}^{(j)} = 2\gamma_{k,s}^{(j)}\alpha^2 \sum_{i=1}^{M}\left(\frac{y_i - y_i^r}{y_i^2} y_i^r\left(1 - y_i^r\right) w_{s,i}^{(j+1)} y_s^{(j)}\left(1 - y_s^{(j)}\right) y_k^{(j-1)}\right). \tag{2.48}$$

Для розрахунку локальної помилки зв'язків (2.48) підставлено в (2.39)

$$\Delta w_{k,s}^{(j)} = 2\gamma_{k,s}^{(j)}\alpha^2 y_s^{(j)}\left(1 - y_s^{(j)}\right) y_k^{(j-1)} \sum_{i=1}^{M}\left(\delta_s^{(j+1)} w_{s,i}^{(j+1)}\right). \tag{2.49}$$

Узагальнивши (2.39, 2.49), по аналогії з [140] , отримаємо вирази (2.50, 2.51) для визначення локальної помилки та величини корекції вагових коефіцієнтів для вхідних зв'язків довільного g-го шару нейронів:

$$\delta_{k,s}^{g} = \left(1 - y_s^{(g)}\right) y_s^{(g)} \sum_{i=1}^{M} \delta_s^{(g+1)} w_{s,i}^{(g+1)}. \tag{2.50}$$

$$\Delta w_{k,s}^{(g)} = 2\gamma_{k,s}^{(g)}\alpha^{N-g+1} y_s^{(g)}\left(1 - y_s^{(g)}\right) y_k^{(g-1)} \sum_{i=1}^{M}\left(\delta_s^{(g+1)} w_{s,i}^{(g+1)}\right). \tag{2.51}$$

Також розглянемо модифікацію алгоритму зворотнього розповсюдження помилок при використанні в якості функції активації гіперболічного тангенсу. В даному випадку вихідний сигнал нейрону

розраховується відповідно виразу:

,

$$y^r(z) = tanh(\alpha z) = \frac{e^{\alpha z} - e^{-\alpha z}}{e^{\alpha \cdot z} + e^{-\alpha \cdot z}}. \quad (2.52)$$

де α – деякий коефіцієнт,

z – зважена сума вхідних сигналів для нейрону.

При цьому

$$\frac{d}{dz} y^r(z) = 1 - tanh(\alpha z)^2 = 1 - \left(y^r(z)\right)^2. \quad (2.53)$$

Розглянемо особливості обчислення вагових коефіцієнтів зв'язків *i*-го нейрону вихідного шару. На відміну від (2.30) частинна похідна ∂y_i^r розраховується так:

$$\frac{\partial y_i^r}{\partial z_i} = 1 - \left(y_i^r\right)^2. \quad (2.54)$$

При формуванні (2.28) знову використано постулат про можливість розгляду складових (2.34-2.36) як неперервних величин. Підставивши (2.34, 2.36, 2.52) в (2.32), отримаємо:

$$\frac{\partial \bar{\varepsilon}_i^2}{\partial w_{s,i}^{(j+1)}} = -\frac{2\left(y_i - y_i^r\right)}{y_i^2} \times \left(1 - \left(y_i^r\right)^2\right) \times y_s^{(j)}. \quad (2.55)$$

Вираз для розрахунку величини корекції зв'язку між *i*-им нейроном вихідного (*j*+1)-го шару та *s*-им нейроном попереднього шару отримаємо, підставивши (2.53) в (2.38)

$$\Delta w_{s,i}^{(j+1)} = -\gamma_{s,i}^{(j+1)} \times \frac{-2\left(y_i - y_i^r\right)}{y_i^2} \times \left(1 - \left(y_i^r\right)^2\right) \times y_s^{(j)}. \quad (2.56)$$

Після тривіальних спрощень отримаємо:

$$\Delta w_{s,i}^{(j+1)} = \frac{2\gamma_{s,i}^{(j+1)}\left(y_i - y_i^r\right)\left(1 - \left(y_i^r\right)^2\right) \times y_s^{(j)}}{y_i^2}. \quad (2.57)$$

По аналогії з [140], використаємо поняття локальної помилки синаптичного зв'язку. З урахуванням функції активації типу гіперболічного тангенсу та функціоналу приведеної квадратичної помилки для зв'язку *i*-го нейрону вихідного шару з *s*-им нейроном попереднього шару вказана локальна помилка розраховується так:

$$\delta_{s,i}^{(j+1)} = \frac{\left(y_i - y_i^r\right)\left(1 - \left(y_i^r\right)^2\right)}{y_i^2}. \quad (2.58)$$

Використання (2.58) дозволяє записати (2.57) у наступному вигляді:

$$\Delta w_{s,i}^{(j+1)} = 2\gamma_{s,i}^{(j+1)} \delta_{s,i}^{(j+1)} y_s^{(j)}. \quad (2.59)$$

По аналогії з випадком використання логістичної функції активації, розглянемо математичний апарат визначення величини корекції зв'язку між *s*-им нейроном останнього *j*-го схованого шару та *k*-им нейроном попереднього шару ($\Delta w_{k,s}^{(j)}$).

Зазначимо, що вирази (2.31-2.34) залишаються без змін, адже до їх складу в явному вигляді ні функція активації, ні її похідна не входять. Однак при визначенні множників добутку (2.34) вирази (2.35, 2.36) відповідно змінюються на (2.60, 2.61)

$$\frac{\partial y_i^r}{\partial z_i} = 1 - \left(y_i^r\right)^2. \tag{2.60}$$

$$\frac{\partial y_s^{(j)}}{\partial z_s^{(j)}} = 1 - \left(y_s^{(j)}\right)^2. \tag{2.61}$$

де $y_s^{(j)}$ – вихідний сигнал s-го нейрону j-го шару.

Підставивши (2.31-2.34, 2.60, 2.61) в (2.41), отримаємо:

$$\Delta w_{k,s}^{(j)} = -\gamma_{k,s}^{(j)} \sum_{i=1}^{M} \left(-\frac{2\left(y_i - y_i^r\right)}{y_i^2} \left(1 - \left(y_i^r\right)^2\right) w_{s,i}^{j+1} y_s^{(j)} \left(1 - \left(y_s^{(j)}\right)^2\right) y_k^{(j-1)} \right). \tag{2.62}$$

Після тривіальних спрощень отримаємо кінцевий вираз для розрахунку величини корекції зв'язку між s-им нейроном останнього j-го схованого шару та k-им нейроном попереднього шару:

$$\Delta w_{k,s}^{(j)} = 2\gamma_{k,s}^{(j)} \sum_{i=1}^{M} \left(\frac{2\left(y_i - y_i^r\right)}{y_i^2} \left(1 - \left(y_i^r\right)^2\right) w_{s,i}^{(j+1)} y_s^{(j)} \left(1 - \left(y_s^{(j)}\right)^2\right) y_k^{(j-1)} \right). \tag{2.63}$$

Використавши (2.39), вираз (2.63) для розрахунку локальної помилки зв'язку схованих нейронів перепишемо так:

$$\Delta w_{k,s}^{(j)} = 2\gamma_{k,s}^{(j)} \left(1 - \left(y_s^{(j)}\right)^2\right) y_k^{(j-1)} \sum_{i=1}^{M} \left(\delta_s^{(j+1)} w_{s,i}^{(j+1)}\right). \tag{2.64}$$

Узагальнивши (2.39, 2.49), отримаємо розрахункові вирази для визначення локальної помилки та величини корекції вагових коефіцієнтів для вхідних зв'язків довільного g-го шару нейронів при використанні гіперболічного тангенсу:

$$\delta_{k,s}^{g} = \left(1 - \left(y_s^{(g)}\right)^2\right) \sum_{i=1}^{M} \delta_s^{(g+1)} w_{s,i}^{(g+1)} \,. \tag{2.65}$$

$$\Delta w_{k,s}^{(g)} = 2\gamma_{k,s}^{(g)} \left(1 - \left(y_s^{(g)}\right)^2\right) y_k^{(g-1)} \sum_{i=1}^{M} \left(\delta_s^{(g+1)} w_{s,i}^{(g+1)}\right). \tag{2.66}$$

З метою верифікації отриманих результатів проведені експерименти, в яких БШП застосовано для моделювання поліноміальних функцій, в яких мінімальні та максимальні значення аргументів відрізнялись між собою в 100 разів. Вказана різниця між мінімальними та максимальними значеннями області визначення функції є характерною для ПБ багатьох об'єктів захисту Інтернет-орієнтованих ІС. Наприклад, таким ПБ може бути завантаження каналу зв'язку веб-сервера.

В якості ілюстрації на рис. 2.7-2.12 показано графіки абсолютної та відносної помилок навчання та екстраполяції функції $y = x^3 + x^2 + x$, апроксимованої ДШП з класичним та модифікованим алгоритмом навчання. Визначено, що оптимальна кількість схованих нейронів в ДШП дорівнює 12.

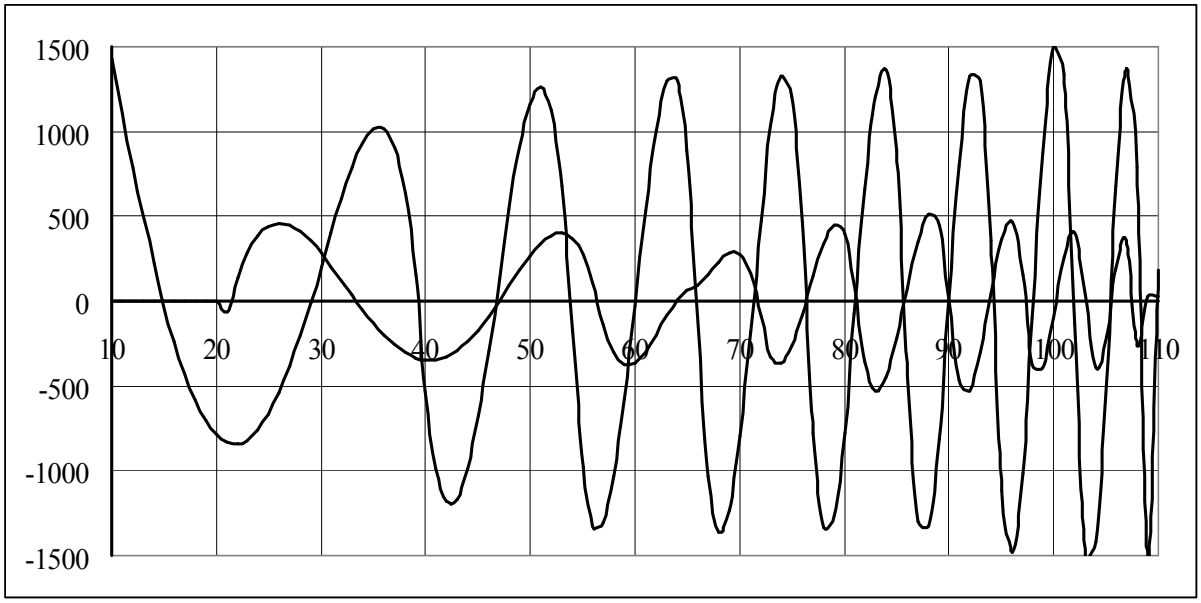


Рис. 2.7 Графіки абсолютної похибки навчання

На рис. 2.7-2.12 цифрою *1* позначено графіки, отримані при використанні класичного алгоритму навчання, а цифрою *2* позначено графіки, які відповідають модифікованому алгоритму. Вісь ординат на графіках рис. 2.7, 2.9, 2.11 відповідає абсолютній помилці навчання (Δ), а на графіках рис. 2.8, 2.10, 20.12 – приведеній помилці навчання (δ). На всіх графіках вісь абцис відповідає значенню піддослідної функції (x).

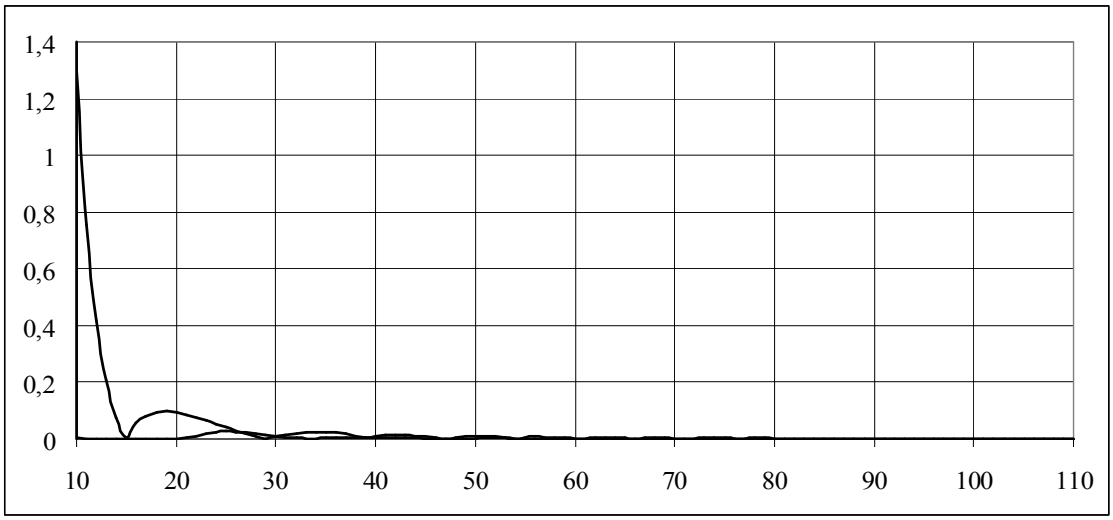


Рис. 2.8 Графіки відносної похибки навчання

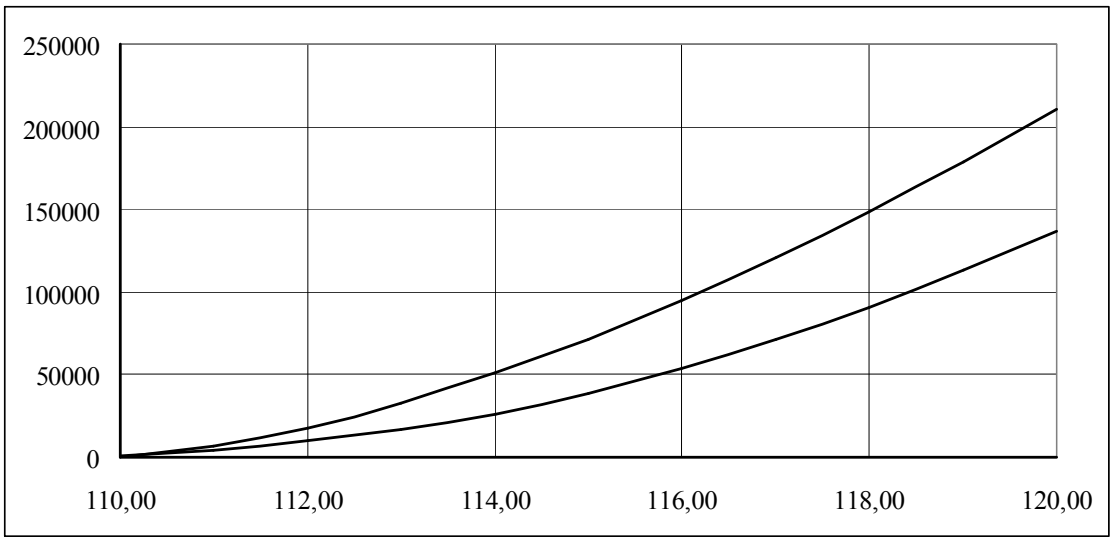


Рис. 2.9 Графіки абсолютної похибки екстраполяції за верхню межу навчальних даних

Як показує аналіз рис. 2.7, 2.8, використання запропонованої модифікації алгоритму зменшує відносну помилку навчання в середньому в 2 рази. При цьому в області невеликих значень модельованої функції відносна помилка навчання зменшилась приблизно в 10 разів.

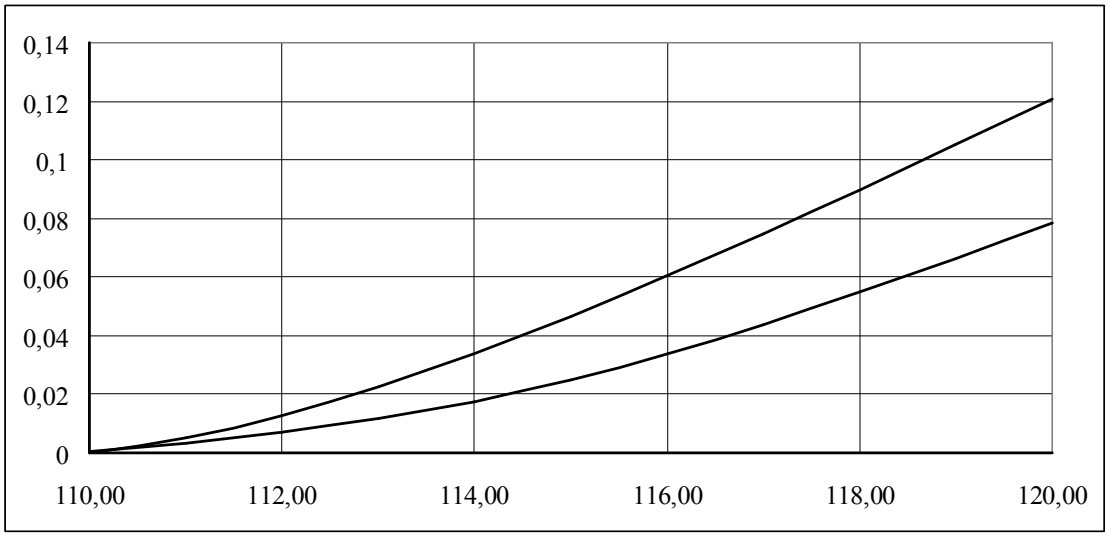


Рис. 2.10 Графіки відносної похибки екстраполяції за верхню межу навчальних даних

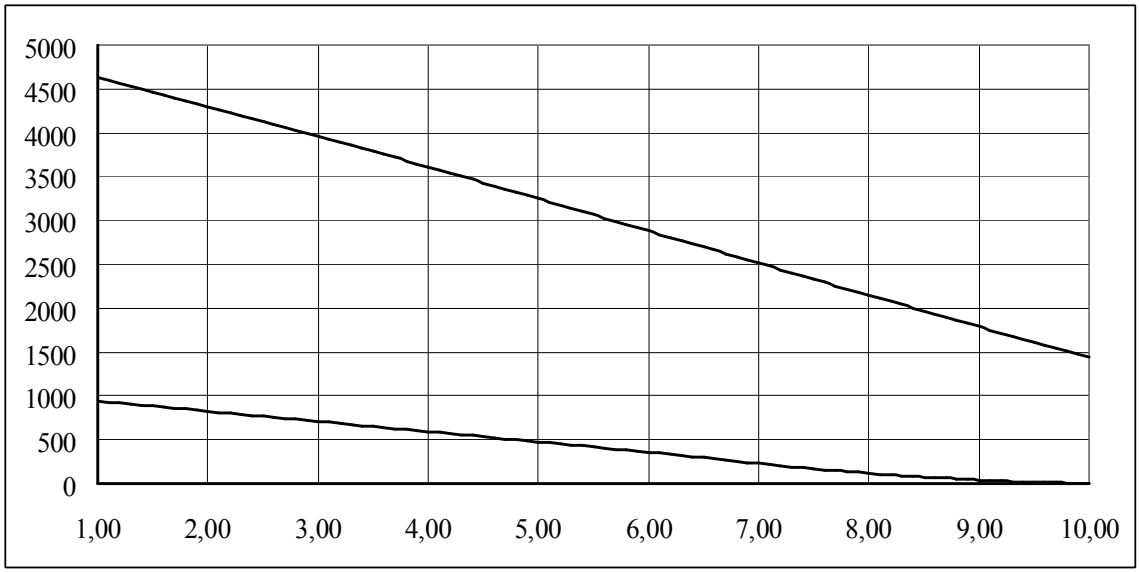


Рис. 2.11 Графіки абсолютної похибки екстраполяції за нижню межу навчальних даних

Аналіз показаних на рис. 2.9, 2.10 графіків вказує на те, що використання модифікованого алгоритму навчання приблизно в 1,5 рази зменшує похибку екстраполяції за верхню межу навчальних даних.

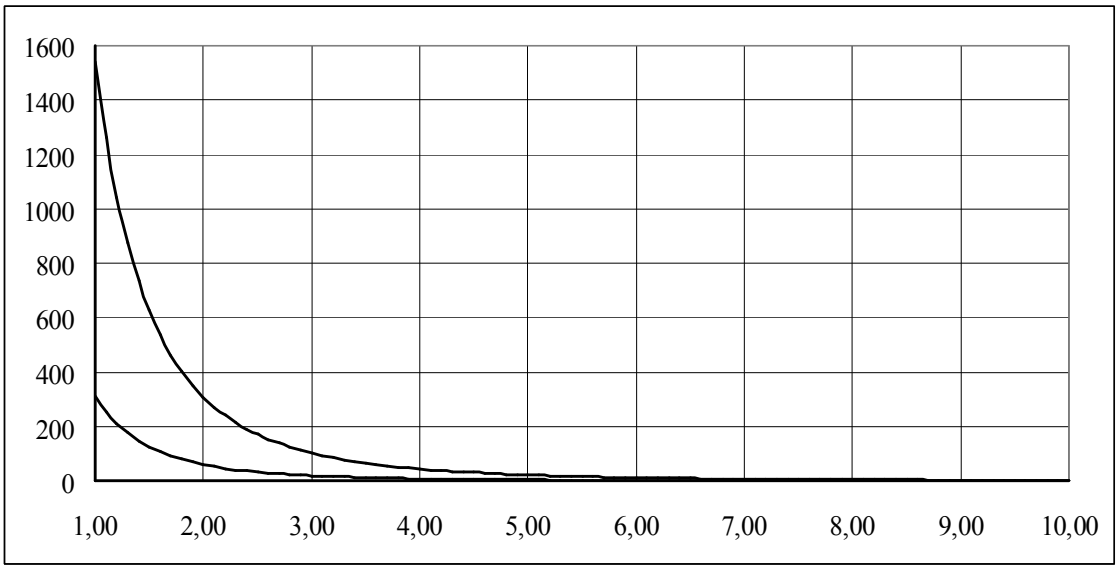


Рис. 2.12 Графіки відносної похибки екстраполяції за нижню межу навчальних даних

Аналіз показаних на рис. 2.11, 2.12 графіків вказує на те, що використання модифікованого алгоритму навчання приблизно в 7 разів зменшує відносну похибку екстраполяції за нижню межу навчальних даних. Таким чином, результати чисельних експериментів підтверджують доцільність запропонованої модифікації алгоритму навчання БШП.

2.4. Верифікація нейромережевих моделей оцінювання параметрів безпеки

В якості відправного пункту дослідження використано результати [33, 49, 140], в яких показана можливість використання НМ для апроксимації з заданою точністю довільної функцій. Так, в [33] показано рішення теореми Хехт-Нільсена, в котрій показана принципова можливість представлення

неперервної довільної функції багатьох змінних за допомогою НМ з прямим поширенням сигналу, що містить як мінімум один СШН. Структурно така НМ складається з N вхідних нейронів, як мінімум з $2N+1$ схованих нейронів з сигмоїдальними функціями активації виду (2.21, 2.44) і M вихідних нейронів з невідомими функціями активації. В [210] результати цієї теореми дещо розширені. Доведено, що параметри сигмоїдальної функції активації можуть бути задані апріорно, а у вихідному шарі нейронів може бути використана лінійна функція активації виду:

$$g(x) = ax + b, \tag{2.67}$$

де g – лінійна функція активації,

x – сумарний вхідний сигнал нейрону,

a, b – коефіцієнти.

Зазначимо, що описаний тип НММ з одним шаром схованих нейронів отримав назву ДШП, а з декількома шарами схованих нейронів – назву БШП [49]. Схожий результат, але вже для НММ типу РБФ, отримано в роботах Д. Парка та І. Сандберга, наведених в [33] . Доведено, що при виконанні певних структурних правил (достатня кількість схованих нейронів), за допомогою РБФ можливо апроксимувати довільну гладку функцію. Таким чином, теоретичній верифікації підлягають НММ типу БШП або РБФ.

Оскільки стан захищеності ІС залежить від подій, які в ньому відбуваються, та характеризуються набором певних підконтрольних ПБ, то модель виявлення кібератак можливо записати так:

$$\exists\, s(t) \in S_a(t) \wedge p(t) \in P_a(t) \Rightarrow A, \tag{2.68}$$

$$\exists\, s(t) \notin S_n(t) \vee p(t) \notin P_n(t) \Rightarrow A, \tag{2.69}$$

де $s(t)$ – множина подій, що відбулись в ІС,

$p(t)$ – множина значень ПБ ІС на момент часу t,

$S_a(t), P_a(t)$ – множина подій в ІС та множина значень ПБ, характерних при реалізації атаки,

$S_n(t), P_n(t)$ – множина подій та множина значень ПБ, характерних для нормального стану ІС на момент часу t,

A – реалізація кібератаки.

Доповненням до виразів (2.68, 2.69) можуть бути вирази (2.70, 2.71), за допомогою яких можливо виявити нормальний стан захищеності ІС:

$$\exists\, s(t) \notin S_a(t) \wedge p(t) \notin P_a(t) \Rightarrow N, \tag{2.70}$$

$$\exists\, s(t) \in S_n(t) \wedge p(t) \in P_n(t) \Rightarrow N, \tag{2.71}$$

де N – нормальний стан захищеності РІС.

Метод виявлення кібератак на основі (2.68, 2.69) отримав назву "виявлення зловживань", а метод виявлення кібератак на основі (2.70, 2.71) – "виявлення аномалій" [37]. Використавши (2.68-2.71), узагальнюючу модель виявлення кібератаки можливо записати у вигляді неперервної функції багатьох змінних:

$$\begin{cases} U = F(s(t), p(t), S_a(t), S_n(t), P_a(t), P_n(t)) \\ U \in (A, N) \end{cases}. \tag{2.72}$$

Зазначимо, що подібна модель виявлення кібератак на ІС використана в [48]. При цьому модель (2.72), як і моделі (2.68-2.71), носить узагальнений характер. В багатьох випадках для виявлення атак використовуються тільки окремі компоненти.

В підсумку, застосування до функціоналу (2.70) теореми Хехт-Нільсена та результатів Д. Парка та І. Сандберга дозволяє стверджувати, що за допомогою БШП та РБФ можна з заданою точністю розпізнати кібератаки на

ІС. При цьому необхідною умовою для верифікації НММ є можливість представлення ПБ у вигляді неперервних функцій.

Розглянемо використання отриманого результату на конкретному прикладі.

Дано: база даних KDD-99, яка містить приклади нормального функціонування ІС та сигнатури мережевих кібератак.

Довести: гарантованість розпізнавання представлених мережевих кібератак за допомогою нейромережевих моделей виду БШП та РБФ.

Рішення. База даних KDD-99 містить близько 5000000 записів – образів мережевих з'єднань, зареєстрованих через певні проміжки часу [18]. 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.ін. В 42 полі записана інформація, що характеризує стан захищеності ІС – або відсутність атаки (normal), або її тип. В 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). Тому для виявлення кібератак можливо використати тільки величини 41 ПБ (параметри мережевого трафіку), множина виявлених атак складається із 22 елементів (представлених видів атак), множина нормальних станів із одного елементу. Це дозволяє переписати (2.68) у вигляді функції

$$\begin{cases} U = F(p_1, p_2, ...p_{41}) \\ U \in (A_1, A_2, ...A_{22}, N_1\} \end{cases}, \tag{2.73}$$

де $p_1, p_2, ...p_{41}$ – ПБ,

$A_1, A_2, ...A_{22}$ – види мережевих кібератак,

N_1 – нормальний стан ІС.

Застосування до функції (2.73) теореми Хехт-Нільсена та результатів Д. Парка та І. Сандберга вказує на можливість використання БШП та РБФ для виявлення кібератак.

2.5. Висновки до другого розділу

В даному розділі вирішувалась наукова задача подальшого розвитку теоретичних положень побудови нейромережевих засобів оцінювання параметрів безпеки Інтернет-орієнтованих інформаційних систем для виявлення кібератак. В процесі вирішення отримано наступні результати:

– Підхід до розпізнавання неочікуваних кібератак, які характеризуються стрибкоподібним та неочікуваним, з точки зору системи захисту інформації, виходом параметрів безпеки за безпечні межі. Підхід передбачає оцінювання відхилення поточних значень параметрів безпеки від нестаціонарних шаблонів поведінки. В якості параметрів безпеки використано параметри зовнішніх запитів до ресурсів інформаційної системи. Застосування підходу дозволяє адаптувати нейромережеві засоби оцінювання до неочікуваних кібератак.

– Підхід до розпізнавання поступових кібератак, які характеризуються тривалим та очікуваним, з точки зору системи захисту інформації, процесом зміни параметрів безпеки до значень, які перевищують безпечні межі. Підхід передбачає оцінювання відхилення поточних значень параметрів безпеки від шаблонів поведінки, розрахованих на протязі заданого інтервалу функціонування. В якості параметрів безпеки використано параметри зовнішніх запитів та функціональні параметри інформаційних систем. Застосування підходу дозволяє адаптувати нейромережеві засоби оцінювання параметрів безпеки до поступових кібератак.

– Підхід до визначення оптимального виду нейромережевої моделі. Оптимальним є той вид нейромережевої моделі, характеристики якого більш повно відповідають множині значимих умов задачі оцінювання параметрів

безпеки. Визначено базовий варіант вказаної множини значимих умов. Застосування підходу дозволило визначити критерії оптимізації виду нейромережевої моделі.

– Підхід до визначення принципової доцільності застосування нейромережевих засобів. Доцільність застосування визначається можливістю в прийнятний термін підготувати достатній обсяг навчальних даних та провести навчання нейромережевої моделі при визначеному обсязі обчислювальних ресурсів. Даний підхід є підґрунтям методу визначення часових характеристик використання нейромережевих засобів, який дозволяє визначити принципову доцільність їх застосування при заданих умовах оцінювання параметрів безпеки.

– Підхід до визначення ефективності розробки нейромережевих засобів, який передбачає використання інтегральних критеріїв оцінки ефективності, що характеризують точність класифікації кібератак, можливість принципової оцінки доцільності застосування НМЗ, адаптацію до нових видів кібератак, пристосованість до варіативності умов застосування та до функціонування при обмежених обчислювальних ресурсах. Даний підхід являється підґрунтям розробки моделі створення ефективних нейромережевих засобів оцінювання параметрів безпеки.

– Підхід до класифікації подібних кібератак. Кібератаки вважаються подібними, якщо вони мають однаковий характер, а приведена різниця параметрів безпеки, що використовуються для їх розпізнавання, не перевищує максимальну. Підхід дозволяє визначити множину кібератак, для розпізнавання якої доцільно сформувати одне продукційне правило.

– Підхід до застосування продукційних правил для подання експертних знань. Підхід базується на аналогії між експертними знаннями у вигляді продукційних правил та навчальними прикладами нейромережевої моделі. Даний підхід являється основою для розробки нейромережевих моделей, пристосованих для навчання за допомогою експертних даних.

– Критерії оптимізації виду нейромережевої моделі, які, відповідно

розробленого підходу, дозволяють співвіднести можливості сучасних видів нейромережевих моделей із значимими умовами задачі оцінювання параметрів безпеки. Також сформовано вираз для розрахунку інтегрального критерію оптимізації. Застосування розроблених критеріїв оптимізації дозволяє підвищити ефективність нейромережевих засобів шляхом їх адаптації до умов задачі оцінювання параметрів безпеки.

– Вдосконалено математичне забезпечення процесу навчання багатошарового персептрону, що дозволяє вирівняти приведену помилку навчання для прикладів з мінімальними та максимальним величинами вхідних параметрів, що характерно при оцінюванні мережевих параметрів безпеки і дозволяє зменшити похибку класифікації мережевих кібератак. Вдосконалення базується на оцінці якості його навчання за допомогою функціоналу приведеної помилки навчання. Показано, що використання вдосконаленого математичного забезпечення дозволяє до 2 разів зменшити помилку навчання багатошарового персептрону.

– Вперше теоретично верифіковано нейромережеві моделі оцінювання параметрів безпеки. Рішення базуються на можливості моделювати кібератаку неперервною багатопараметричною функцією та використанні теорем Колмогорова-Арнольда і Хехт-Нільсена, в яких доведено можливість застосування НМ з прямим розповсюдженням сигналу для представлення довільної неперервної багатопараметричної функції з заданою точністю. Наведено приклад верифікації нейромережевих моделей для виявлення мережевих кібератак, сигнатури яких представлені в базі даних KDD-99.

РОЗДІЛ 3
МОДЕЛІ ОЦІНЮВАННЯПАРАМЕТРІВ БЕЗПЕКИ ІНТЕРНЕТ-ОРІЄНТОВАНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

3.1. Модель процесів інтеграції параметрів безпеки, що використовуються нейромережевими засобами розпізнавання кібератак

Для створення моделі оцінки параметрів, що використовуються в нейромережевих засобах розпізнавання кібератак, використано три множини – множину можливих кібератак на РІС ($\boldsymbol{Ka}$), множину можливих ПБ ($\boldsymbol{X}$) та множину вихідних параметрів НМ ($\boldsymbol{Y}$). В загальному випадку

$$\boldsymbol{Ka} = \bigcup_{j=1}^{J} Ka_j \,, \tag{3.1}$$

$$\boldsymbol{X} = \bigcup_{i=1}^{I} X_i \,, \tag{3.2}$$

$$\boldsymbol{Y} = \bigcup_{g=1}^{G} Y_g \,, \tag{3.3}$$

де Ka_j – j-а кібератака,

J – кількість можливих кібератак,

G – кількість вихідних параметрів НМ,

Y_g – g-ий вихідний параметр,

X_i – i-ий ПБ,

I – кількість ПБ.

Кожному елементу (типу кібератаки) множини $\boldsymbol{Ka}$ поставлена у

відповідність підмножина вихідних параметрів $\boldsymbol{Y}_N$ (значення яких свідчать про наявність кібератаки), що складається із елементів множини $\boldsymbol{Y}$. В свою чергу, кожній $\boldsymbol{Y}_N$ поставлена у відповідність підмножина вхідних параметрів $\boldsymbol{X}_V$ (необхідних для виявлення кібератаки), яка складається із елементів множини $\boldsymbol{X}$. Таким чином, для виявлення кібератаки слід сформувати множини трійок «вхідні параметри→вихідні параметри→ кібератака»:

$$\begin{aligned}
&\boldsymbol{X}_V \to \boldsymbol{Y}_N \to \boldsymbol{Ka} = \bigcup_{i=1}^{V_j} X_i \to \bigcup_{g=1}^{N_j} Y_g \to \bigcup_{j=1}^{J} Ka_j = \\
&(\{X_{1,1}, X_{1,2}, \ldots X_{1,V_1}\} \to \{Y_{1,1}, Y_{1,2}, \ldots Y_{1,N_1}\} \to Ka_1), \\
&(\{X_{1,1}, X_{1,2}, \ldots X_{1,V_1}\} \to \{Y_{1,1}, Y_{1,2}, \ldots Y_{1,N_1}\} \to Ka_1), \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
&(\{X_{J,1}, X_{J,2}, \ldots X_{J,V_J}\} \to \{Y_{J,1}, Y_{J,2}, \ldots Y_{J,N_J}\} \to Ka_J).
\end{aligned} \tag{3.4}$$

Використання НМ передбачає, що область множин $\boldsymbol{X}$ та $\boldsymbol{Y}$ визначені на $[0..1]$.

При відомих трійках (3.4) виявлення кібератаки *k*-го типу зводиться до встановлення відповідності між поточними величинами вхідних параметрів $\{x_{k,1}, x_{k,2}, \ldots x_{k,V_k}\}$ та величинами множини вхідних параметрів $\{X_{k,1}, X_{k,2}, \ldots X_{k,V_k}\}$, які відповідають величинам вихідних параметрів $\{Y_{k,1}, Y_{k,2}, \ldots Y_{k,N_k}\}$, котрі свідчать про Ka_k:

$$\{x_{k,1}, x_{k,2} \ldots x_{k,V_k}\} \cong \{X_{k,1}, X_{k,2} \ldots X_{k,V_k}\} \Rightarrow \{Y_{k,1}, Y_{k,2} \ldots Y_{k,N_k}\} \Rightarrow Ka_k, \tag{3.5}$$

Модель (3.4) деталізовано з врахуванням розроблених підходів до розпізнавання НК та ПК. Для цього множину можливих кібератак представлено у наступному вигляді:

$$\boldsymbol{Ka} = (\boldsymbol{Ks}, \boldsymbol{Kq}), \tag{3.6}$$

де $\boldsymbol{Ks}, \boldsymbol{Kq}$ – відповідно множина поступових та несподіваних кібератак.

Множину вхідних параметрів $\boldsymbol{X}$ також розділено на дві частини:

$$\boldsymbol{X} = (\boldsymbol{Xs}, \boldsymbol{Xq}), \tag{3.7}$$

де $\boldsymbol{Xs}$ – множина ПБ, що використовуються для розпізнавання ПК,

$\boldsymbol{Xq}$ – множина ПБ, що використовуються для розпізнавання НК.

Підставивши (3.6, 3.7) в (3.5), отримаємо:

$$\{xs_{k,1}, xs_{k,2} \ldots xs_{k,V_k}\} \cong \{Xs_{k,1}, Xs_{k,2} \ldots Xs_{k,V_k}\} \to \{Ys_{k,1}, Ys_{k,2} \ldots Ys_{k,N_k}\} \to Ks_k, \tag{3.8}$$

$$\{xq_{k,1}, xq_{k,2} \ldots xq_{k,V_k}\} \cong \{Xq_{k,1}, Xq_{k,2} \ldots Xq_{k,V_k}\} \to \{Yq_{k,1}, Yq_{k,2} \ldots Yq_{k,N_k}\} \to Kq_k. \tag{3.9}$$

Використання (3.8, 3.9) дозволяє визначити модель нейромережевої оцінки вхідних параметрів для виявлення *k*-ої ПК чи НК у наступному вигляді:

$$\{xs_{k,1}, xs_{k,2} \ldots xs_{k,V_k}\}[nnet]\{Xs_{k,1}, Xs_{k,2} \ldots Xs_{k,V_k}\} \to \{Ys_{k,1}, Ys_{k,2} \ldots Ys_{k,N_k}\}, \tag{3.10}$$

$$\{xq_{k,1}, xq_{k,2} \ldots xq_{k,V_k}\}[nnet]\{Xq_{k,1}, Xq_{k,2} \ldots Xq_{k,V_k}\} \to \{Yq_{k,1}, Yq_{k,2} \ldots Yq_{k,N_k}\}, \tag{3.11}$$

де $[nnet]$ – оператор нейромережевого порівняння.

Враховуючи (3.10, 3.11), узагальнені вирази для нейромережевої оцінки поточних вхідних параметрів можливо представити так:

$$xs_i [nnet] Xs_i \to Ys_i, \tag{3.12}$$

$$xq_i [nnet] Xq_i \to Yq_i. \tag{3.13}$$

Зазначимо, що, відповідно розробленого підходу до розпізнавання ПК та результатів [40], в (3.12) необхідно врахувати залежності xs_i та Xs_i від терміну експлуатації. Тому:

$$xs_i(t)[nnet]\,Xs_i(t) \to Ys_i\,. \tag{3.14}$$

Таким чином, при виявленні ПК НМ повинна застосовуватись для класифікації часових рядів даних, що, відповідно [124], може викликати значні труднощі. Для подолання цих труднощів запропоновано проводити попередню обробку поступових ПБ з метою видалення із них часової залежності.

Відповідно результатів [21-23], визначити часову залежність пропонується за допомогою додаткової марківської моделі поступових ПБ. Це дозволяє видозмінити (3.14) так:

$$(xs_i(t) - \overline{Xs}_i(t))[nnet](Xs_i(t) - \overline{Xs}_i(t)) \to Ys_i\,, \tag{3.15}$$

де $\overline{Xs}_i(t)$ – розрахована за допомогою марківської моделі величина ПБ в момент часу t.

Ще однією задачею розробки ефективних НМЗ оцінювання ПБ ІС для виявлення кібератак є визначення номенклатури вхідних параметрів НММ. Необхідність розв'язання вказаної задачі пояснюється наступними чинниками:

– використання в якості вхідних параметрів НММ великої кількості ПБ ІС значно збільшує обсяг обчислювальних ресурсів та ускладнює процес накопичення навчальних прикладів;

– використання малоінформативних ПБ призводить до навчання НММ на зашумлених даних, що негативно впливає на правильність класифікації невідомих прикладів та збільшує обсяг обчислювальних ресурсів;

– вилучення із вхідних параметрів НММ інформативних ПБ може призвести до повної втрати її класифікаційних властивостей.

Разом з тим, результати [59, 86] вказують на те, що остаточне рішення про номенклатуру вхідних параметрів НМ приймається в результаті досить тривалих порівняльних експериментів. Для зменшення кількості цих експериментів доцільно визначити важливість кожного із можливих ПБ. Оскільки сучасні формалізовані методи оцінки важливості ПБ не відповідають вимогам точності [88,], прийнято рішення про застосування експертного оцінювання. Пропонується використати метод парних порівнянь, що пояснюється його доведеною ефективністю у випадках великої кількості піддослідних об'єктів, з якими асоціюються ПБ [90].

В цьому випадку вхідними даними моделі являється вектор, елементами якого є матриці експертних оцінок вагомості ПБ:

$$\overline{T} = \{E_1, E_2, ...E_M\}, \tag{3.16}$$

де E_m – матриця оцінок m-го експерта,

M – кількість експертів.

В свою чергу матриця оцінок має наступний вигляд:

$$E_m = \begin{vmatrix} e_{1,1} & \dots & e_{1,j} & \dots & e_{1,N} \\ \dots & \dots & \dots & \dots & \dots \\ e_{i,1} & \dots & e_{i,j} & \dots & e_{j,N} \\ \dots & \dots & \dots & \dots & \dots \\ e_{N,N} & \dots & e_{N,j} & \dots & e_{N,N} \end{vmatrix}, \tag{3.17}$$

де $e_{i,j}$ – оцінка i-го ПБ відносно j-го параметра,

N – кількість ПБ.

Для заповнення матриці (3.17) експерт повинен попарно порівняти значимості ПБ. Якщо експерт вважає, що i-ий ПБ важливіший, ніж j-ий

параметр, то $e_{i,j}=1$. В протилежному випадку $e_{i,j}=0$. Оскільки порівняння значимості параметра з самим собою беззмістовне, то діагональ матриці (3.17) не заповнюється.

Приклад заповнення M експертами відповідних матриць оцінок ПБ показано в табл. 3.1.

Таблиця 3.1

Оцінка ПБ

Експерт	ПБ	x_1	x_n	x_N
1	x_1	–	1	1
	x_n	0	–	1
	x_N	0	0	–
m	x_1	–	0	1
	x_n	1	–	1
	x_N	0	0	–
M	x_1	–	1	1
	x_n	0	–	1
	x_N	0	0	–

В підсумку, в узагальненому вигляді вхідні дані моделі представляють собою трьохвимірний масив виду:

$$\boldsymbol{C}=\left\{c_{1,1,1},\ldots c_{i,j,k},\ldots c_{N,N,M}\right\}, \tag{3.18}$$

де $c_{i,j,k}$ – виставлена *k*-им експертом оцінка порівняння *i*-го ПБ з *j*-им ПБ.

Результатом оцінювання важливості ПБ є вектор коефіцієнтів вагомості:

$$\overline{\boldsymbol{B}} = \{\beta_1, \beta_2, ...\beta_N\}, \tag{3.19}$$

де β_i – коефіцієнт вагомості *i*-го ПБ.

Узагальнено процес оцінювання важливості ПБ можливо записати у вигляді функції:

$$f : \boldsymbol{C} \to \overline{\boldsymbol{B}}, \tag{3.20}$$

Основу математичного забезпечення перетворення (3.20) складають наступні вирази:

$$\beta_i = \frac{s_i}{\sum_{i=1}^{N} s_i}, \tag{3.21}$$

$$s_j = \sum_{j=1}^{M} c_{i,j}^{\Sigma}, \tag{3.22}$$

де $c_{i,j}^{\Sigma}$ – елемент матриці переваг C^{Σ},

s_i – значимість *i*-го ПБ.

Елементи матриці переваг розраховуються так:

$$c_{i,j}^{\Sigma} = \sum_{m=1}^{M} c_{i,j,m}. \tag{3.23}$$

Приклад матриці переваг, заповненої по даним табл. 3.1 в передумові, що $M = 3$ і $N = 3$, показано у вигляді табл. 3.2.

Для перевірки правильності визначення коефіцієнтів вагомості значимості можливо використовувати умову нормування:

$$\sum_{i=1}^{N} \beta_i = 1. \qquad (3.24)$$

Крім розрахунку коефіцієнтів вагомості, математичне забезпечення оцінювання важливості ПБ включає в себе вираз для визначення ступеню узгодженості експертних даних:

$$\begin{cases} \textit{Якщо } W > 0{,}5 \Rightarrow \textit{експертні дані узгоджені}, \\ \textit{Якщо } W \le 0{,}5 \Rightarrow \textit{експертні дані не узгоджені}. \end{cases}, \qquad (3.25)$$

де W – коефіцієнт конкордації, $W \in [0,1]$

Таблиця 3.2

Приклад матриці переваг

Експерт	ПБ		
	x_1	x_2	x_3
1	1	2	3
2	2	1	3
3	1	2	3

У випадку неузгодженості експертних даних процедуру експертного оцінювання слід повторити.

При цьому розрахунок коефіцієнту конкордації реалізується за допомогою наступних виразів:

$$W = \frac{12L}{M^2(N^3 - N)}, \qquad (3.26)$$

$$L = \sum_{n=1}^{N} (r_n - r_{cp})^2, \qquad (3.27)$$

$$r_{cp} = 0{,}5M(N+1), \tag{3.28}$$

$$r_n = \sum_{m=1}^{M}\sum_{i=1}^{N} c_{n,i,m}, n = 1{,}2...N, \tag{3.29}$$

де W – коефіцієнт конкордації,

r_n – сумарний ранг оцінок n-го ПБ по всім експертам,

r_{cp} – середній ранг експертних оцінок важливості ПБ,

L – коефіцієнт відхилення сумарних рангів від середнього.

В якості вхідних параметрів НМЗ слід використовувати тільки ті ПБ, для яких коефіцієнт вагомості більший від заданого мінімального значення (β_{min}). Тобто:

$$\textit{Якщо } \beta_i > \beta_{min} \Rightarrow i - \textit{ий параметр використовувати доцільно}. \tag{3.30}$$

В результаті розроблено модель процесів інтеграції ПБ ІС, що використовуються для розпізнавання ПК та НК (рис.3.1). Вхідною інформацією моделі являється введена множина характеристик об'єкту захисту (***O***), що визначається виразом (2.6), а виходом моделі є множина ПБ, оцінювання яких дозволяє визначити ПК та НК, характерні для об'єкту захисту ІС. Модель складається із п'яти базових процесів, котрі співвідносяться з процесами інтеграції.

Процес 1 – визначення початкових параметрів, що відбувається в результаті аналізу множини ***O***. Початковими параметрами являються множина кібератак, характерних для об'єкту захисту ІС $\{Ka\}_K$ та множина підконтрольних ПБ $\{x\}_N$, де K – кількість кібератак, N – кількість ПБ.

Процес 2 – класифікація ПБ. Результатом даного процесу є визначення із $\{x\}_N$ множини ПБ, придатних для розпізнавання НК $\{xq\}$, та множини ПБ, придатних для розпізнавання ПК $\{xs\}$.

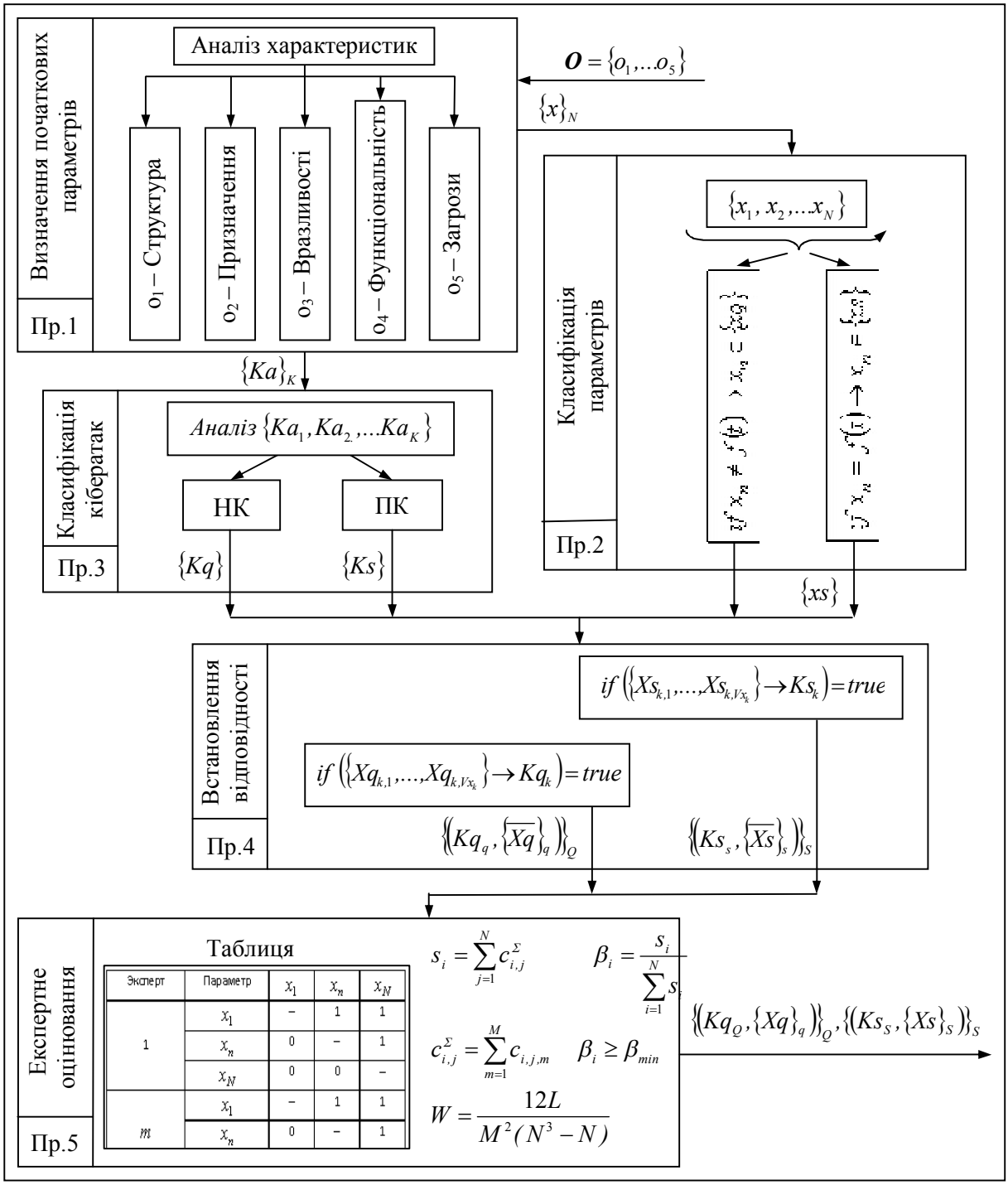


Рис. 3.1. Відображення моделі процесів інтеграції параметрів безпеки

Відповідно розроблених підходів, придатність *n*-го ПБ до розпізнавання ПК та НК визначається за допомогою виразів:

$$if\ x_n \neq f(t) \rightarrow x_n \in \{xq\}. \qquad (3.31)$$

$$if\ x_n = f(t) \to x_n \in \{xs\}. \quad (3.32)$$

Процес 3 – класифікація кібератак. Процес орієнтований на аналіз $\{Ka\}_K$ з метою виділення множини НК $\{Kq\}$ та множини ПК $\{Ks\}$. В процесі аналізу використовуються підходи до розпізнавання НК та ПК.

Процес 4 – встановлення відповідності. В даному процесі в першому наближенні для кожної можливої кібератаки встановлюється відповідність з множиною ПБ, котрі можуть використовуватись в якості вхідних параметрів НММ. Відповідність встановлюється, якщо для *k*-го виду ПК та/або НК справедливими є вирази:

$$if\left(\left\{Xs_{k,1},...,Xs_{k,Vx_k}\right\} \to Ks_k\right) = true, \quad (3.33)$$

$$if\left(\left\{Xq_{k,1},...,Xq_{k,Vx_k}\right\} \to Kq_k\right) = true, \quad (3.34)$$

де $\left\{Xs_{k,1},...,Xs_{k,Vx_k}\right\}$, $\left\{Xq_{k,1},...,Xq_{k,Vx_k}\right\}$ – множини ПБ, що використовуються для розпізнавання ПК та НК,

Vx_k – кількість ПБ, що використовуються для розпізнавання *k*-го виду ПК та/або НК.

Виходом процесу є визначені в першому наближенні $\left\{\left\langle Kq_q, \left\{\overline{Xq}\right\}_q\right\rangle\right\}_Q$ та $\left\{\left\langle Ks_s, \left\{\overline{Xs}\right\}_s\right\rangle\right\}_S$, де Q, S – кількість можливих НК та ПК.

Процес 5 – експертне оцінювання. В результаті реалізації даного процесу остаточно визначається множина ПБ, що використовується в якості вхідних параметрів НМЗ. Для цього методом парних порівнянь оцінки експертних даних за допомогою виразів (3.16-3.30) розраховуються ПБ, які доцільно використовувати для розпізнавання. Для *k*-ої кібератаки вихід процесу задається виразом $\langle K_k, \{X\}_k\rangle$, де $\{X\}_k$ – множина важливих ПБ.

Слід зазначити, що процеси 1-4 можуть виконуватись за допомогою наведеного методу парних порівнянь, відповідно (3.16-3.30). Крім того, можуть використовуватись інші методи оцінки експертних даних, наведені в [88, 90].

Розроблена модель використана для визначення ПБ, які можуть використовуватись в НМЗ антивірусних сканерів для виявлення веб-орієнтованих скриптових вірусів, написаних на мові програмування JavaScript. Зазначимо, що, відповідно специфіки антивірусних сканерів, в якості ПБ використовуються назви операторів мови програмування JavaScript, отримані в результаті аналізу програмного коду Веб-сторінки [169]. Також результати [46] вказують на можливість використання в НМ одного вихідного параметру, величина якого вказує на наявність певного скриптового вірусу. Тому вирази (3.1-3.3) трансформуються так:

$$Ka = \{ \textit{скриптовий вірус 1}, \ldots, \textit{скриптовий вірус } J \}, \tag{3.35}$$

$$X = \{ \textit{оператор 1}, \ldots, \textit{оператор } I \}, \tag{3.36}$$

$$Y = \{ \textit{вихідний параметр 1} \}, \tag{3.37}$$

де J – кількість скриптових вірусів, які можуть бути розпізнані,

I – кількість операторів мови програмування JavaScript.

Значення ПБ не залежать від часу, а про реалізацію кібератаки (наявність вірусу) свідчить певна комбінація їх значень, котра, з точки зору апроксимації статистичних даних, носить неочікуваний характер. Тому виявлення Веб-орієнтованих вірусів класифіковано як виявлення НК, а при оцінюванні ПБ враховано, що $\boldsymbol{Ka} = \boldsymbol{Kq}$, $\boldsymbol{X} = \boldsymbol{Xq}$ та використано вирази (3.11, 3.13, 3.15). Для зменшення обсягу аналізуємих НМ вхідних параметрів використана задана виразами (3.16-3.30) процедура експертного оцінювання

вагомості параметрів. В (3.30) прийнято, $\beta_{min} = 0{,}5$. В результаті визначена множина вхідних параметрів, які відповідають назвам потенційно небезпечних операторів JavaScript.

3.2. Марківська модель одноперіодичного шаблону поведінки

Відповідно розробленого одноперіодичного ШП, марківська модель повинна описувати нестаціонарний процес $X = f(t)$, який послідовно зростає на стаціонарних інтервалах типу $B_d A_{d+1}$ і спадає на стаціонарних інтервалах типу $A_{d+1} B_{d+2}$, де d – номер перехідної точки (див. рис. 2.4). Тому розроблена марківська модель одноперіодичного ШП M_{BAB} складається із двох однорідних ЛМ M_{BA} та M_{AB}, призначених для моделювання ПБ на стаціонарних інтервалах типу $B_d A_{d+1}$ і $A_{d+1} B_{d+2}$ відповідно. Структура розробленої моделі M_{BAB} показана на рис. 3.2.

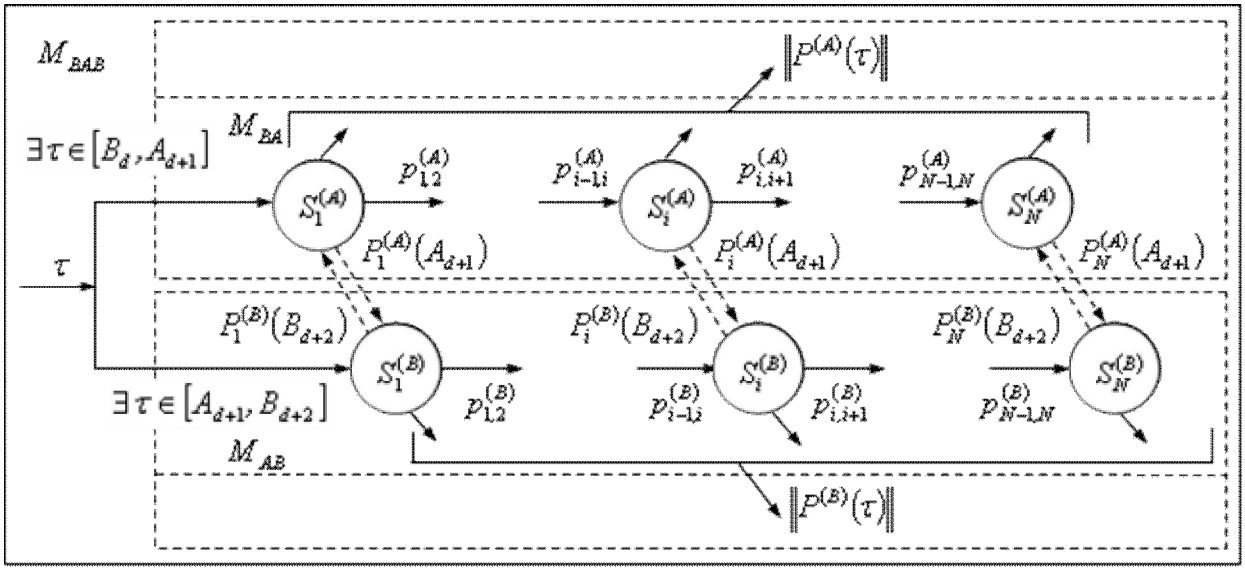


Рис. 3.2. Марківська модель одноперіодичного шаблону поведінки

На рис. 3.2. позначено: τ – крок розрахунку, $S_i^{(A)}$ та $S_i^{(A)}$ – i-ий стан

M_{BA} та M_{AB}, $p_{i,i+1}^{(A)}$ та $p_{i,i+1}^{(B)}$ – ймовірність переходу із *i*-го в *(i+1)*-ий стан для M_{BA} та M_{AB} , $P_i^{(A)}(A_{d+1})$ – ймовірність перебування процесу в *i*-му стані для M_{BA} в перехідній точці (*d*+1), $P_i^{(B)}(B_{d+2})$ – ймовірність перебування процесу в *i*-му стані для M_{AB} в перехідній точці (*d*+2), $\left\|P^{(A)}(\tau)\right\|$ – вектор розподілу ймовірностей на τ-му кроці розрахунку для M_{BA}, $\left\|P^{(B)}(\tau)\right\|$ – вектор розподілу ймовірностей на τ-му кроці розрахунку для M_{AB}, *N* – кількість станів M_{BA} та M_{AB}.

Для визначення параметрів M_{BA} та M_{AB} використовуються апріорно розраховані інтервали $B_d A_{d+1}$ та $A_{d+1}B_{d+2}$, кількість та межі станів ЛМ, матриці перехідних ймовірностей $\pi^{(A)} = \left\|p_{i,i+1}^{(A)}\right\|$ і $\pi^{(B)} = \left\|p_{i,i+1}^{(B)}\right\|$ та вектор початкового розподілу ймовірностей $\left\|P^{(A)}(0)\right\| = \left\langle P_1^{(A)}(0), P_2^{(A)}(0), ..., P_N^{(A)}(0)\right\rangle$.

Поточний час моделювання розраховується так:

$$t = \tau \times \Delta t\,, \tag{3.38}$$

де Δt – тривалість кроку моделювання.

Для обчислення ймовірностей станів ЛМ M_{BA} (M_{AB}) використовується система рівнянь Колмогорова-Чепмена:

$$\left\|P^{(A)}(\tau)\right\| = \left\|P^{(A)}(\tau - 1)\right\| \times \pi^{(A)}, \tag{3.39}$$

$$\left\|P^{(B)}(\tau)\right\| = P\left\|^{(B)}(\tau - 1)\right\| \times \pi^{(B)}, \tag{3.40}$$

де $\left\|P^{(A)}(\tau)\right\|$ ($\left\|P^{(B)}(\tau)\right\|$) – вектор ймовірностей станів M_{BA} (M_{AB}) на τ-му кроці розрахунку.

Також використовуються умови нормування:

$$\sum_{i=1}^{N} P_i^{(A)} = 1, \tag{3.41}$$

$$\sum_{i=1}^{N} P_i^{(B)} = 1. \tag{3.42}$$

При переході τ із інтервалу $B_d A_{d+1}$ в $A_{d+1} B_{d+2}$ початковий вектор розподілу M_{AB} дорівнює кінцевому вектору розподілу M_{BA}:

$$\left\| P^{(B)}\left(A_{d+1}\right) \right\| = \left\| P^{(A)}\left(A_{d+1}\right) \right\|. \tag{3.43}$$

При переході τ із інтервалу $A_{d+1} B_{d+2}$ в $B_{d+2} A_{d+3}$ навпаки, початковий вектор розподілу M_{BA} дорівнює кінцевому вектору розподілу M_{AB}:

$$\left\| P^{(A)}\left(B_{d+2}\right) \right\| = \left\| P^{(B)}\left(B_{d+2}\right) \right\|. \tag{3.44}$$

Таким чином, марківська модель M_{BAB} дозволяє моделювати одноперіодичний ШП ПБ ІС.

3.3. Марківська модель багатоперіодичного шаблону поведінки

Розробка марківської моделі багатоперіодичного ШП базується на визначеній в [125] можливості представлення багатоперіодичного ряду динаміки у вигляді суперпозиції декількох одноперіодичних рядів. Тому розроблена марківська модель багатоперіодичного ШП M_{BAB}^{Σ}, структура якої показана на рис. 3.3, складається із модулів $M_{BAB}^{(1)}, M_{BAB}^{(2)}, \ldots, M_{BAB}^{(K)}$,

призначених для моделювання K значимих періодів ШП. При цьому довільний k-ий модуль $M_{BAB}^{(k)}$ представляє собою розроблену в п. 3.2 марківську модель одноперіодичного ШП, призначену для моделювання k-ої періодичної складової. В першому наближенні для всіх модулів кількість та межі станів ЛМ однакові. В свою чергу, $M_{BAB}^{(k)}$ складається із двох ЛМ – $M_{BA}^{(k)}$ та $M_{AB}^{(k)}$, призначених для моделювання k-ої періодичної складової ШП на стаціонарних інтервалах цього періоду $B_d A_{d+1}{}^{(k)}$ і $A_{d+1} B_{d+2}{}^{(k)}$.

Виходом k-го модулю $M_{BAB}^{(k)}$ на τ-му кроці розрахунку є $\left\| P^{(k)}(\tau) \right\|$ – вектор розподілу ймовірностей для k-ої періодичної складової ШП.

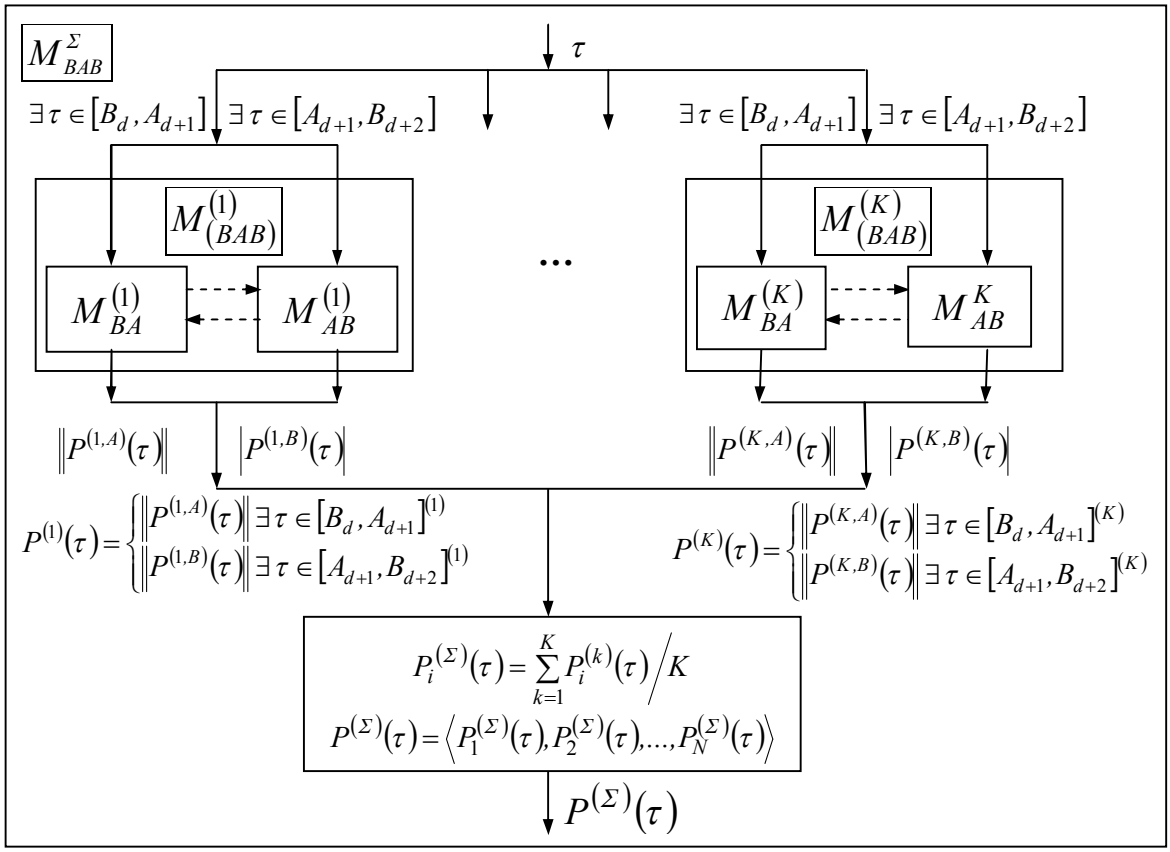


Рис. 3.3. Марківська модель багатоперіодичного ШП ПБ

Виходом моделі M_{BAB}^{Σ} на τ-му кроці розрахунку є інтегральний вектор розподілу ймовірностей виду:

$$\left\| P^{(\Sigma)}(\tau) \right\| = \left\langle P_1^{(\Sigma)}(\tau), P_2^{(\Sigma)}(\tau), ..., P_N^{(\Sigma)}(\tau) \right\rangle, \quad (3.45)$$

де $P_i^{(\Sigma)}(\tau)$ – інтегральна ймовірність перебування ПБ в *i*-му стані ЛМ на *τ*-му кроці розрахунку.

В свою чергу, $P_i^{(\Sigma)}(\tau)$ розраховується так:

$$P_i^{(\Sigma)}(\tau) = K^{-1} \sum_{k=1}^{K} P_i^{(k)}(\tau), \quad (3.46)$$

де $P_i^{(k)}(\tau)$ – ймовірність перебування ПБ в *i*-му стані *k*-го ЛМ на *τ*-му кроці розрахунку.

Розроблені марківські моделі M_{BAB} та M_{BAB}^{Σ} застосовано для створення ШП Веб-серверу. В якості ПБ *X* використано кількість звернень до Веб-серверу. Графіки динаміки математичного сподівання даного ПБ, побудовані на основі статистичних даних та за допомогою запропонованих марківських моделей, показано на рис. 3.4.

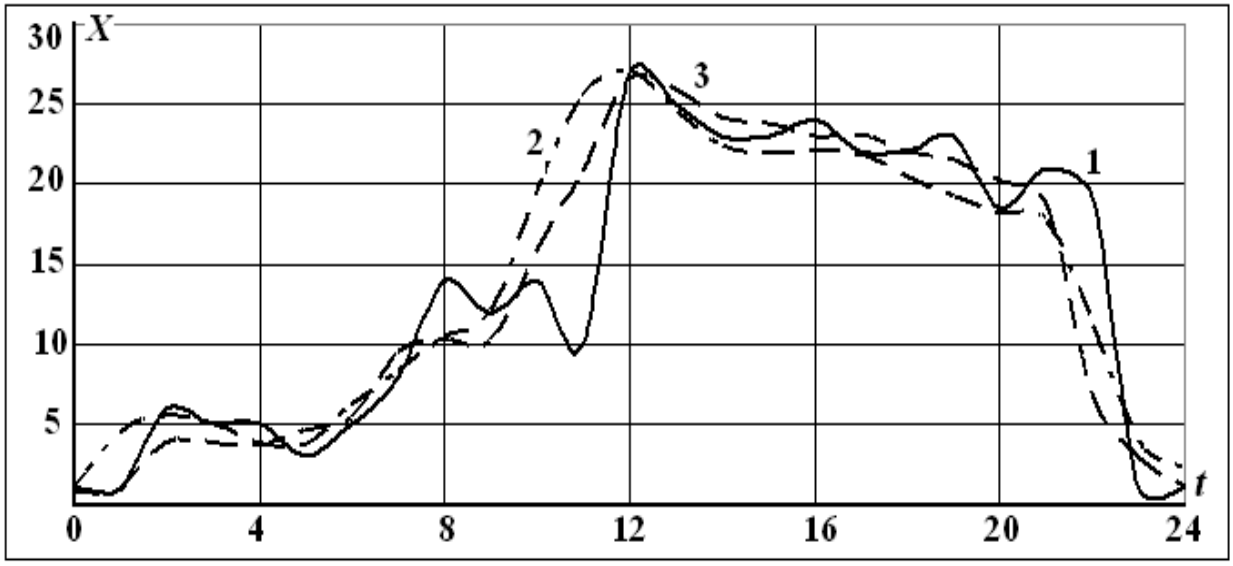


Рис. 3.4. Графіки динаміки математичного сподівання кількості звернень

На рис. 3.4. цифрою 1 позначено графік на основі статистичних даних, цифрою 2 – на основі одноперіодичної моделі M_{BAB}, а цифрою 3 – на основі двохперіодичної моделі M_{BAB}^{Σ}.

Для одноперіодичної моделі M_{BAB} середня похибка моделювання становить 0,09, а для двохперіодичної моделі M_{BAB}^{Σ} – 0,07. При цьому середня похибка моделювання, розрахована на основі поліноміальних моделей [115], становить 0,14÷0,18.

Таким чином, застосування моделей M_{BAB} та M_{BAB}^{Σ} дозволяє зменшити похибку моделювання в 1,5-2 рази, що підтверджує доцільність їх застосування.

3.4. Модель на основі багатошарового персептрону

Відповідно [49], розробка структурної моделі БШП базується на визначенні загальної кількості синаптичних зв'язків. По аналогії з [140], в основу такого визначення покладемо критерій мінімізації відносної помилки БШП при дотриманні обмежень для актуальних задач оцінювання ПБ:

$$\begin{cases} G_{MLP} \to min, \\ O \le O_d. \end{cases}, \tag{3.47}$$

де G_{MLP} – відносна помилка БШП,

O_d –обмеження.

Відносна помилка БШП дорівнює відношенню помилки узагальнення (ε) до кількості синаптичних зв'язків (L_w)

$$G_{MLP} = \frac{\varepsilon}{L_w}. \tag{3.48}$$

Перепишемо (3.47) з урахуванням (3.48)

$$\begin{cases} \dfrac{\varepsilon}{L_w} \to min, \\ O \le O_d . \end{cases} \tag{3.49}$$

Для визначення точки мінімуму, яка відповідає оптимальній кількості синаптичних зв'язків (L_w^{opt}), слід розв'язати наступне рівняння:

$$\frac{\partial L_w}{\partial \varepsilon} = 0 . \tag{3.50}$$

Відповідно [49], помилка узагальнення БШП розраховується як сума помилок апроксимації (ε_a) та опису моделі (ε_o)

$$\varepsilon = \varepsilon_a + \varepsilon_o . \tag{3.51}$$

Помилка апроксимації (навчання) співвідноситься із запам'ятовуванням БШП навчальних даних, а помилка опису моделі співвідноситься з узагальненням (стисненням) цих даних. Зазначимо, що як запам'ятовування, так і стиснення навчальних даних відбувається за рахунок зміни вагових коефіцієнтів синаптичних зв'язків.

Вважають [40, 210] , що помилка апроксимації БШП (ε_a) пропорційна відношенню кількості синаптичних зв'язків до кількості компонент вхідного вектора (N_X)

$$\varepsilon_a \sim \frac{N_X}{L_w}, \tag{3.52}$$

Помилка опису моделі БШП пропорційна відношенню кількості

синаптичних зв’язків до кількості навчальних прикладів (P)

$$\varepsilon_o \sim \frac{L_w}{P}, \tag{3.53}$$

Узагальнений вираз для оцінки загальної помилки отримаємо, підставивши (3.51) та (3.52) в (3.53):

$$\varepsilon \sim \left(\frac{N_X}{L_w} + \frac{L_w}{P} \right). \tag{3.54}$$

Після тривіальних перетворень отримаємо точку максимуму:

$$L_w^{opt} \sim \sqrt{P \times N_X}\ . \tag{3.55}$$

Вираз (3.55) для розрахунку оптимальної кількості синаптичних зв’язків дозволяє перейти до визначення оптимальної кількості схованих нейронів. Зазначимо, що співвідношення між кількістю синаптичних зв’язків та кількістю схованих нейронів БШП задається виразом

$$L_w = N_X \times N_1 + \sum_{s=1}^{S-1} \left(N_s \times N_{s+1} \right) + N_S \times N_Y\ . \tag{3.56}$$

де N_X – кількість вхідних нейронів,

N_1 – кількість нейронів в першому СШН,

N_s – кількість нейронів в s–ому СШН,

N_Y – кількість нейронів у ШВ,

S – кількість СШН.

Врахуємо теорему Хехта-Нільсена [33, 34], в якій доведено, що для представлення довільної функції достатньо двохшарової НМ прямого

розповсюдження сигналу з повнозв'язною структурою, що складається з *n* вхідних нейронів, (2*n*+1) схованих нейронів та *m* вихідних нейронів. Це дозволяє спростити модель БШП до ДШП.

Вказане спрощення, хоча і суперечить [32, 34] в контексті зменшення обчислювальних можливостей, однак відповідає таким вимогам до НМ в задачах оцінювання ПБ, як максимальна простота та надійність. Адаптований до ДШП вираз (3.56) виглядає так:

$$L_w = \left(N_X + N_Y\right) \times N_1 . \tag{3.57}$$

Для багатьох задач оцінювання ПБ для розпізнавання кібератак вихід НМ може вказувати тільки на ймовірність (впевненість) виникнення очікуваної події, наприклад, реалізації мережевої кібератаки на ІС. В цьому випадку НМ може мати один вихідний елемент ($N_Y = 1$), що зумовлює зміну (3.57) виразом

$$L_w = \left(N_X + 1\right) \times N_1 . \tag{3.58}$$

Прирівняємо (3.57) до (3.55). Отримаємо

$$\sqrt{P \times N_X} \sim \left(N_X + N_Y\right) \times N_1^{opt}, \tag{3.59}$$

$$N_1^{opt} \sim \frac{\sqrt{P \times N_X}}{N_X + N_Y}, \tag{3.60}$$

де N_1^{opt} – оптимальна кількість схованих нейронів в ДШП з довільною кількістю вихідних нейронів.

Для ДШП з одним вихідним зв’язком (3.60) можна спростити так:

$$Ns_1^{opt} \sim \frac{\sqrt{P \times N_X}}{N_X + 1}, \tag{3.61}$$

де Ns_1^{opt} – оптимальна кількість схованих нейронів в ДШП з одним вихідним нейроном.

Вирази (3.60, 3.61) представляють собою пропорції, а значить, не дозволяють безпосередньо розрахувати оптимальну кількість нейронів в СШН. Для переходу до рівняння введемо в (3.60) коефіцієнт пропорційності:

$$N_1^{opt} = k \times \frac{\sqrt{P \times N_X}}{N_X + N_Y}, \tag{3.62}$$

де k – коефіцієнт пропорційності.

Проведемо оцінку означеного коефіцієнту. В загальному випадку[34], мінімально допустима кількість схованих нейронів визначається теоремою Хехта-Нільсена, а максимально допустима кількість обмежується кількістю навчальних прикладів. Тобто

$$N_1^{min} \geq 2N_X + 1, \tag{3.63}$$

$$N_2^{max} \leq P, \tag{3.64}$$

де N_1^{min}, N_2^{max} – мінімальна та максимальна кількість нейронів у СШН.

Порівнявши (3.62) з (3.63) та (3.63) з (3.64), отримаємо

$$\begin{cases} k \times \dfrac{\sqrt{P \times N_X}}{N_X + N_Y} \geq 2N_X + 1, \\ k \times \dfrac{\sqrt{P \times N_X}}{N_X + N_Y} \leq P. \end{cases} \tag{3.65}$$

Як наслідок,

$$\begin{cases} k \geq \dfrac{(2N_X+1)\times(N_X+N_Y)}{\sqrt{P\times N_X}}, \\ k \leq \dfrac{P\times(N_X+N_Y)}{\sqrt{P\times N_X}}. \end{cases} \quad (3.66)$$

В теорії НМ [140] вважається доведеним, що кількість навчальних прикладів повинна перевищувати кількість вхідних параметрів як мінімум в 10 разів. Тобто

$$N_X \times P \geq 10N_X^2. \quad (3.67)$$

Підставивши (3.67) в (3.66) отримаємо

$$\begin{cases} k \geq \dfrac{(2N_X+1)\cdot(N_X+N_Y)}{10N_X}, \\ k \leq \dfrac{P\times(N_X+N_Y)}{10N_X}. \end{cases} \quad (3.68)$$

$$\begin{cases} k \geq \dfrac{(2N_X+1)\cdot(N_X+N_Y)}{10N_X}, \\ k \leq \dfrac{P\times N_X+P\times N_Y}{10N_X}. \end{cases} \quad (3.69)$$

$$\begin{cases} k \geq \dfrac{(2N_X+1)\cdot(N_X+N_Y)}{10N_X}, \\ k \leq \dfrac{10N_X^2+P\times N_Y}{10N_X}. \end{cases} \quad (3.70)$$

Проведемо уточнення меж діапазону величин коефіцієнту

пропорційності з урахуванням специфіки актуальних задач оцінювання ПБ. Як правило, в таких задачах кількість вихідних параметрів не перевищує кількості вхідних параметрів, а кількість навчальних прикладів повинна перевищувати кількість розрізняємих класів (вихідних параметрів) як мінімум в 10 разів. Тому, не порушуючи нерівності (3.70), можна вважати

$$N_Y \approx N_X. \tag{3.71}$$

$$N_Y \times P \approx 10N_X^2. \tag{3.72}$$

Після підстановки (3.71, 3.72) в (3.70) отримаємо

$$\begin{cases} k \geq \dfrac{(2N_X + 1)\cdot(N_X + N_X)}{10N_X}, \\ k \leq \dfrac{10N_X^2 + 10N_X^2}{10N_X}. \end{cases} \tag{3.73}$$

$$\begin{cases} k \geq 0{,}4N_X + 0{,}2, \\ k \leq 2N_X. \end{cases} \tag{3.74}$$

Підстановка (3.74) в (3.62) дозволяє записати вирази для оцінки діапазону оптимальної кількості схованих нейронів ДШП у вигляді:

$$N_1^{opt} \geq (0{,}4N_X + 0{,}2) \times \frac{\sqrt{P \times N_X}}{N_X + N_Y}, \tag{3.75}$$

$$N_1^{opt} \leq \frac{2\sqrt{P \times N_X}}{N_Y}, \tag{3.76}$$

Врахувавши в (3.75, 3.76) те, що кількість схованих нейронів має бути

цілим числом, отримаємо остаточні розрахункові вирази:

$$N_1^{opt}(min) = Round\left((0{,}4N_X + 0{,}2) \times \frac{\sqrt{P \times N_X}}{N_X + N_Y} \right), \tag{3.77}$$

$$N_1^{opt}(max) = Round\left(\frac{2\sqrt{P \times N_X}}{N_Y} \right), \tag{3.78}$$

де $N_1^{opt}(max)$, $N_1^{opt}(min)$ – максимальна та мінімальна межа діапазону оптимальної кількості схованих нейронів,

Round(X) – операція визначення найближчого цілого числа від аргументу *X*.

Схема відображення оптимізації структури БШП показана на рис. 3.5.

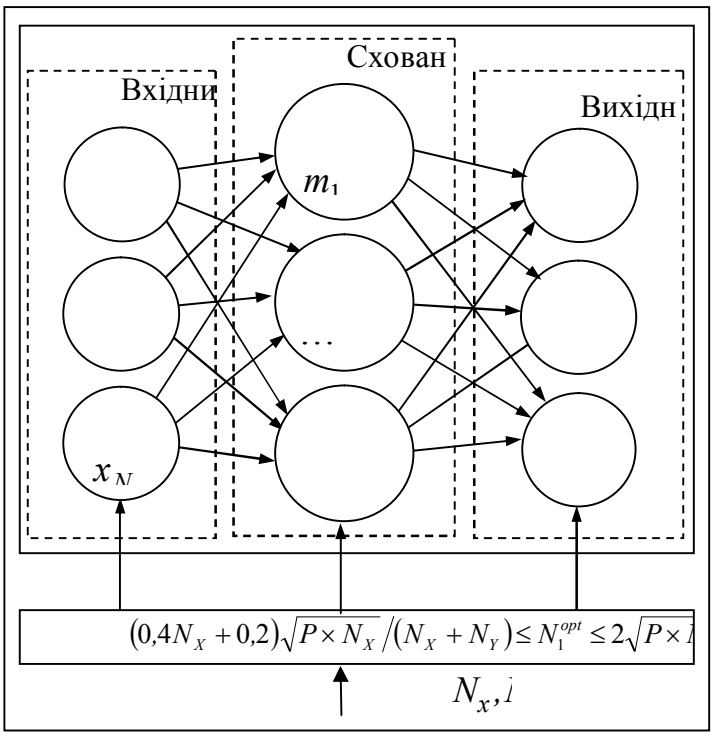


Рис.3.5. Схема оптимізації структури БШП

Базуючись на (3.77, 3.78), для ДШП з одним вихідним нейроном

оптимальну кількість схованих нейронів можна оцінити так:

$$Ns_1^{opt}(min) = Round\left((0{,}4N_X + 0{,}2) \times \frac{\sqrt{P \times N_X}}{N_X + 1}\right), \tag{3.79}$$

$$Ns_1^{opt}(max) = Round\left(2\sqrt{P \times N_X}\right), \tag{3.80}$$

Для ряду задач оцінювання ПБ кількість вхідних параметрів значно перевищує кількість вихідних параметрів

$$N_X >> N_Y, N_X >> 1. \tag{3.81}$$

Використання (3.81) дозволяє спростити (3.77-3.80) так:

$$N_1^{opt}(min) = Round\left((0{,}4N_X + 0{,}2) \times \sqrt{P/N_x}\right), \tag{3.82}$$

$$N_1^{opt}(max) = Round\left(2\sqrt{P \times N_X}\right), \tag{3.83}$$

З метою оцінки інформативності отриманих результатів розглянемо зміну діапазону пошуку оптимальної кількості схованих нейронів при використанні розробленої моделі. Зміну діапазону будемо оцінювати за допомогою виразу

$$\delta = \frac{N_1^{opt}(max) - N_1^{opt}(min)}{N_1^{max} - N_1^{min}}. \tag{3.84}$$

Врахуємо, що для більшості практичних задач оцінювання ПБ для розпізнавання кібератак на ІС кількість вхідних параметрів менша від 100 ($N_x \approx 100$), а кількість навчальних прикладів, необхідних для ефективного

навчання НМ, повинна перевищувати 10000 ($P \approx 10000$). Вважатимемо, що застосовується ДШП з одним виходом ($N_Y=1$). Використавши вказані параметри в розробленій моделі (3.82, 3.83), отримаємо $Ns_1^{opt}(min) \approx 400$, $Ns_1^{opt}(max) \approx 2000$. Для загальної моделі (3.77, 3.78) відповідні результати інші – $N_1^{min} \approx 200$, $N_1^{max} \approx 10000$. Використавши ці результати в (3.84), отримаємо $\delta \approx 0{,}16$. Таким чином, діапазон пошуку оптимальної кількості схованих нейронів звузився приблизно в 6 раз.

Розглянемо загальніший варіант – $N_x \approx 100$, а $P/N_x \approx 10$. В цьому випадку $Ns_1^{opt}(min) \approx 120$, $Ns_1^{opt}(max) \approx 620$, $N_1^{min} \approx 200$, $N_1^{max} \approx 1000$. Відповідно, $\delta \approx 0{,}63$. Таким чином, діапазон пошуку оптимальної кількості схованих нейронів звузився приблизно в 1,6 раз. Для елементарного випадку – $N_x \approx 1$, а $P \approx 100$. В цьому випадку отримаємо $Ns_1^{opt}(min) = 6$, $Ns_1^{opt}(max) = 20$, $N_1^{min} = 3$, $N_1^{max} = 100$. Підставивши ці величини в (3.84), отримаємо $\delta \approx 0{,}18$. Таким чином, діапазон пошуку оптимальної кількості схованих нейронів звузився приблизно в 5,7 раз. Враховуючи [32], можна очікувати, що застосування розробленої моделі дозволить зменшити обчислювальні витрати на загальну розробку ДШП в 1,1-2 рази.

З метою верифікації отриманих результатів проведені числові експерименти по апроксимації одно-, дво- та багатопараметричної поліноміальної функції за допомогою ДШП, кількість схованих нейронів якого розраховувалась за допомогою (3.77, 3.78, 3.82, 3.83) та за допомогою виразів, отриманих в результаті аналізу [101, 140, 210]:

$$\frac{0{,}5P}{1+\ln P} \le N_2^{min} \tag{3.85}$$

$$N_2^{max} \le \frac{1{,}5P+1{,}5}{N_1} + 0{,}5N_1 \tag{3.86}$$

$$\frac{P}{20} - 0{,}5N_1 - 0{,}5N_0 \le N_3^{min}, \tag{3.87}$$

$$N_3^{max} \le \frac{P}{4} - 0{,}5N_1 - 0{,}5N_0, \tag{3.88}$$

де N_2^{min}, N_2^{max} – мінімальна та максимальна межа діапазону оптимальної кількості схованих нейронів, відповідно[101],

N_2^{min}, N_2^{max} – мінімальна та максимальна межа, відповідно [210] .

Результати експериментів підтвердили звуження діапазону пошуку оптимальної кількості схованих нейронів в середньому в 1,5-2 рази.

В якості ілюстрації проведених експериментів на рис. 3.6-3.10 показано зміну відносних помилок навчання та узагальнення ДШП, який застосовано для інтерполяції та екстраполяції функції $y = 2x + 1$. На рис. 3.6-3.10 символом Δ позначено відносну абсолютну помилку результатів, отриманих за допомогою ДШП, X – аргумент функції, N – кількість схованих нейронів.

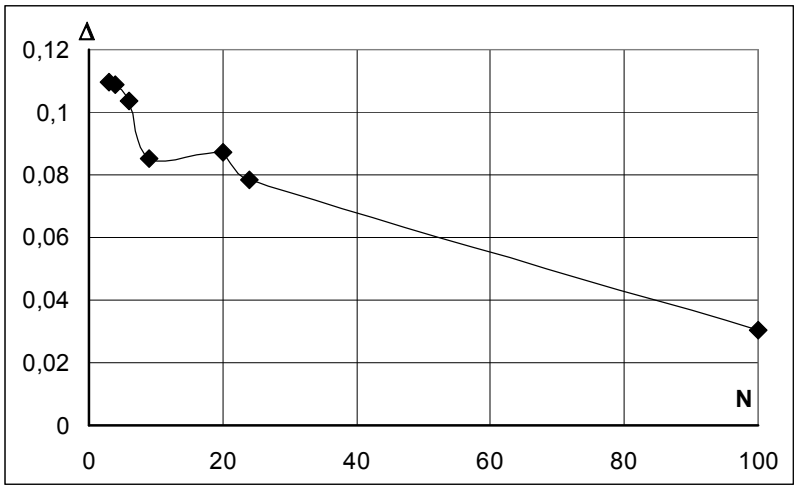


Рис. 3.6 Середня відносна помилка навчання ДШП

Кількість схованих нейронів ДШП розраховувались за допомогою виразів (3.77, 3.78, 3.82, 3.83, 3.85-3.88). При проведені експериментів

прийнято, що кількість навчальних прикладів P=100, а кількість вхідних та вихідних нейронів N_X=N_Y=1. Відповідно, в (3.77, 3.78, 3.82, 3.83, 3.85-3.88) $N_1^{min}=3$, $N_2^{max}=100$, $N_1^{opt}(min)=6$, $N_1^{opt}(max)=20$, $N_2^{min}=9$, $N_2^{max}=152$, $N_3^{min}=4$, $N_3^{max}=24$. Як показує аналіз рис. 3.6, помилка ДШП на навчальних прикладах пропорційно зменшується із збільшенням кількості схованих нейронів, що в цілому відповідає висновкам [46]. Помилка узагальнення ДШП оцінювалась на основі величини відносної помилки інтерполяції даних за допомогою ДШП (див. рис. 3.7).

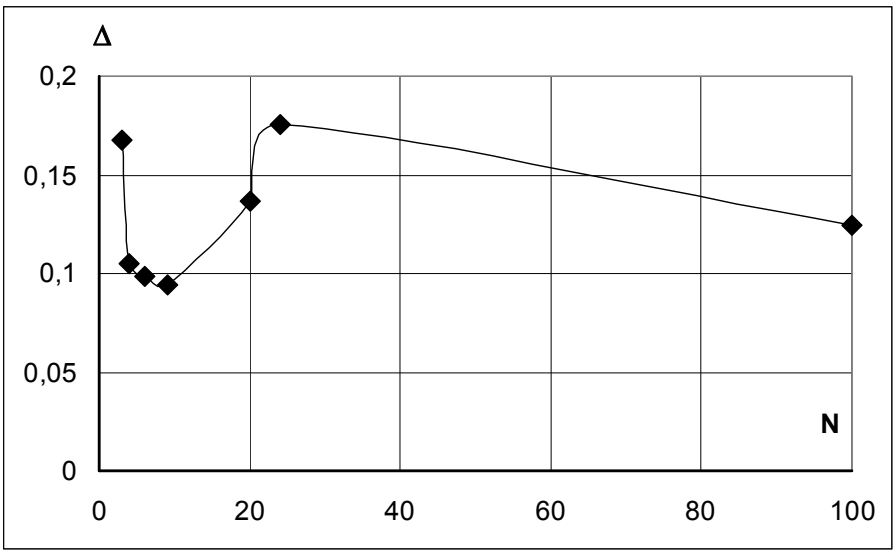


Рис. 3.7 Середня відносна абсолютна помилка інтерполяції ДШП

Аналіз рис. 3.7 вказує на мінімізацію помилки інтерполяції ДШП при кількості схованих нейронів в межах від 6 до 20, що відповідає виразам (3.55, 3.56) та підтверджує достовірність теоретичних викладок. Для повноти аналізу придатності використання ДШП в задачах оцінювання ПБ проведено аналіз обчислювальної складності його навчання. Оскільки в більшості випадків оптимальний розподіл вагових коефіцієнтів шукається за допомогою градієнтних методів [140], то кількість операцій (ξ_1), потрібних для розрахунку градієнта функції помилки $\partial\varepsilon/\partial L_w$, визначається пропорцією виду:

$$\xi_1 \sim P \times L_w, \tag{3.89}$$

де L_w – кількість синаптичних зв'язків,

P – кількість навчальних прикладів.

Враховуючи, що швидкість сходження найкращих методів навчання пропорційна кількості синоптичних зв'язків [140], загальну кількість обчислювальних операцій (ξ), потрібних для досягнення нульової помилки, розрахуємо так:

$$\xi = \mu \times P \times L_w^2, \tag{3.90}$$

де μ – коефіцієнт пропорційності, що в першому наближені дорівнює 1.

Вираз (3.90) дозволяє оцінити мінімальну кількість навчальних операцій для БШП з довільною структурою.

Зазначимо, що для ДШП кількість синаптичних зв’язків дорівнює

$$L_w = (N_X + N_Y)N_1. \tag{3.91}$$

Підставивши (3.90) в (3.91), отримаємо

$$\xi = \mu \times P \times (N_X + N_Y)^2 N_1{}^2. \tag{3.92}$$

Для розрахунку мінімальної кількості навчальних операцій (ξ_{opt}) в ДШП з оптимальною структурою підставимо вирази (3.82, 3.83) в (3.92):

$$\xi_{min}^{opt} = \mu \times P \times (N_X + 1)^2 \left(Round\left((0{,}4N_X + 0{,}2) \times \frac{\sqrt{P \times N_X}}{N_X + N_Y} \right) \right)^2, \tag{3.93}$$

$$\xi_{max}^{opt} = \mu \times P \times \left(N_X + 1_Y\right)^2 \left(Round \frac{\left(2\sqrt{P \times N_X}\right)}{N_Y} \right)^2, \quad (3.94)$$

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

Після перетворень та спрощень отримаємо:

$$\xi_{min}^{opt} \approx 0{,}16 \mu P^2 N_X^3, \quad (3.95)$$

$$\xi_{max}^{opt} \approx 4 \mu P^2 N_X^3. \quad (3.96)$$

Вирази (3.95, 3.96) визначають залежність між кількістю обчислювальних операцій оптимізованого ДШП та максимальною кількістю статистично подібних прикладів, яку він може безпомилково запам’ятати.

Також співставлення (3.95) та (3.96) показує, що

$$\xi_{max}^{opt} \approx 25 \xi_{min}^{opt}. \quad (3.97)$$

В більшості задач оцінювання ПБ кількість обчислювальних навчальних операцій обмежена максимально допустимим терміном навчання.

Позначимо максимально допустиму кількість навчальних операцій як ξ_d. Підставивши ξ_d в (3.95, 3.96) та провівши відповідні перетворення, отримаємо вираз для оцінки максимальної кількості навчальних прикладів ДШП з оптимальною структурою при умові безпомилкового навчання:

$$P_{1,max} \approx 2{,}5 \frac{\sqrt{\xi_d}}{\sqrt{\mu} N_X^{1,5}}, \quad (3.98)$$

$$P_{2,max} \approx 0{,}5\frac{\sqrt{\xi_d}}{\sqrt{\mu}N_X^{1,5}}, \tag{3.99}$$

де $P_{1,max}$, $P_{2,max}$ – максимальна кількість навчальних образів для ДШП з кількістю схованих нейронів, яка дорівнює нижній та верхній межі оптимального діапазону.

При цьому

$$P_{1,max} = 5P_{2,max}. \tag{3.100}$$

Зазначимо, що вирази (3.95, 3.96, 3.99, 3.100) отримані при умові нульової помилки апроксимації (навчання) $\varepsilon_a = 0$. В той же час результати [140] вказують на можливість навчання з деякою допустимою помилкою:

$$\varepsilon_a \le \varepsilon_{ad} \tag{3.101}$$

де ε_{ad} – допустима помилка навчання (апроксимації).

Також аналіз [140, 210] вказує на експоненційний характер залежності між кількістю обчислювальних навчальних операцій та помилкою навчання. Це дозволяє модифікувати (3.95, 3.96, 3.99, 3.100) для врахування очікуваної помилки навчання

$$\xi_{min}^{opt} \approx 0{,}16e^{-\chi\varepsilon_a}\mu P^2 N_X^3, \tag{3.103}$$

$$\xi_{max}^{opt} \approx 4e^{-\chi\varepsilon_a}\mu P^2 N_X^3, \tag{3.104}$$

$$P_{1,max} \approx \frac{\sqrt{\xi_d}}{0{,}4\sqrt{e^{-\chi\varepsilon_a}\mu}N_X^{1,5}}, \tag{3.105}$$

$$P_{2,max} \approx \frac{\sqrt{\xi_d}}{2\sqrt{e^{-\chi\varepsilon_a}\mu N_X^{1,5}}}, \qquad (3.106)$$

де χ – деяка константа.

В першому наближенні можна вважати, що $\chi \approx 1$. При цьому в багатьох задачах оцінювання ПБ для виявлення кібератак [88, 94] очікувана розмірність вхідного сигналу, а відповідно, і кількість вхідних нейронів не буде перевищувати 1000 ($N_X \leq 10^3$). Разом з тим, очікувана розмірність вихідного сигналу, яка відповідає кількості вихідних нейронів, дорівнює 1, тобто $N_Y=1$. Також можна вважати, що загальновживаний термін навчання НМ повинен знаходитись в межах однієї доби ($\approx 10^5$ с).

Для прикладу розглянемо задачу побудови БШП для розпізнавання поштових скриптових вірусів. Відповідно [158, 164], кількість параметрів ПБ, за допомогою яких можливо розпізнати найбільш поширені поштові скриптові віруси, написані на мові VBS, менша ніж 100 ($N_X \leq 100$). Можливо застосувати БШП з одним вихідним нейроном, вихідний сигнал якого вказує: вірусу немає ($0 \leq N_Y < 0{,}33$), підозра на вірус ($0{,}33 \leq N_Y < 0{,}66$) та вірус є ($0{,}66 < N_Y \leq 1$). При використанні персонального комп'ютера з потужністю приблизно 10^{10} операцій в секунду, вказаному терміну навчання 10^5 с відповідатиме $\xi \approx 8{,}64 \times 10^{14}$ обчислювальних операцій. На основі (3.105, 3.106) визначено, що при умові безпомилкового навчання, обсяг навчальної бази даних ДШП становитиме $P \approx 1{,}5*10^4 \ldots 7{,}5*10^4$ прикладів, що відповідає обсягу баз даних сучасних антивірусних засобів. При цьому очікувана помилка узагальнення знаходиться в діапазоні [0,03...0,14], що вважається прийнятним для евристичних аналізаторів сучасних антивірусних систем.

Розраховані величини обсягу навчальної вибірки та очікуваної помилки узагальнення підтверджують високий потенціал використання БШП з оптимізованою структурою при вирішенні задач оцінювання ПБ ІС для виявлення кібератак.

3.5. Модель мережі MPNN

Для створення мережі MPNN використано розроблений підхід до застосування продукційних правил для подання експертних знань в НММ, придатні для навчання шляхом безпосереднього запам'ятовування начальних прикладів. Зазначимо, що в результаті досліджень, проведених в п. 1.3, визначено, що, з точки зору оцінки ПБ для виявлення кібератак, серед таких НММ високий потенціал має PNN. Структуру мережі PNN, призначеної для розпізнавання мережевих кібератак за рахунок класифікації одного із двох можливих станів ІС (*A* – безпечний стан, *B* – реалізація мережевої кібератаки), показано на рис. 3.8. В цій мережі нейрони ШО з номерами від *1* до *L* відповідають навчальним прикладам, які співвідносяться з безпечним станом, а нейрони з номерами від *L+1* до *N* – співвідносяться з реалізацією кібератак.

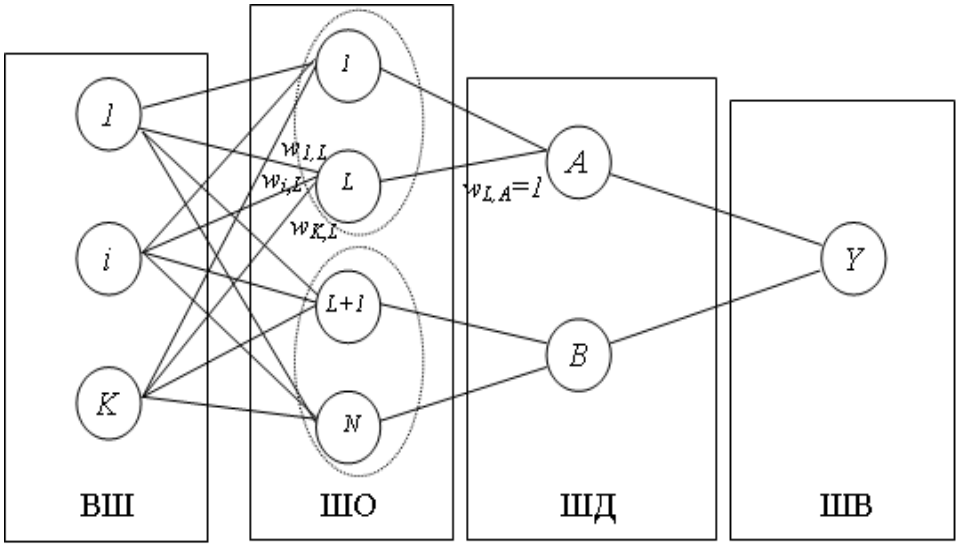


Рис. 3.8. Структура мережі PNN

Безпечний стан співвідноситься з нейроном ШД *A*, а стан реалізації мережевої атаки – з нейроном ШД *B*. На нейрони ВШ подають інформацію, яка відповідає нормалізованим величинам контрольованих ПБ ІС, значення

яких можуть сигналізувати про наявність/відсутність кібератак. Наприклад, для мережевої кібератаки ПБ можуть бути 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 кібератаки виду (2.18, 2.19) достатньо:

- визначити в ШД два нейрони А та В, котрі співвідносяться з безпечним та небезпечним станом ІС;
- 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 *A* або *B*.

Для прикладу на рис. 3.8 показано вагові коефіцієнти $w_{L,1}$, $w_{i,L}$, $w_{K,L}$ та $w_{L,A}$, за рахунок яких в мережу PNN внесено приклад *i*, який відповідає безпечному стану ІС.

Зазначимо, що, відповідно [140], для підвищення ефективності процесу розрахунку вихідного сигналу мережу PNN доцільно представити в 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.

Аналіз [184] відомих прикладів правил визначення безпечного/небезпечного стану ІС, що застосовуються в СВА, виявив дві властивості, які недостатньо враховуються в структурі та математичному забезпеченні класичної мережі PNN:

1. Кожному окремому типу кібератаки може відповідати одна комбінація ПБ. 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. 3.9.

2. В багатьох випадках продукційні правила матимуть вигляд (2.20).

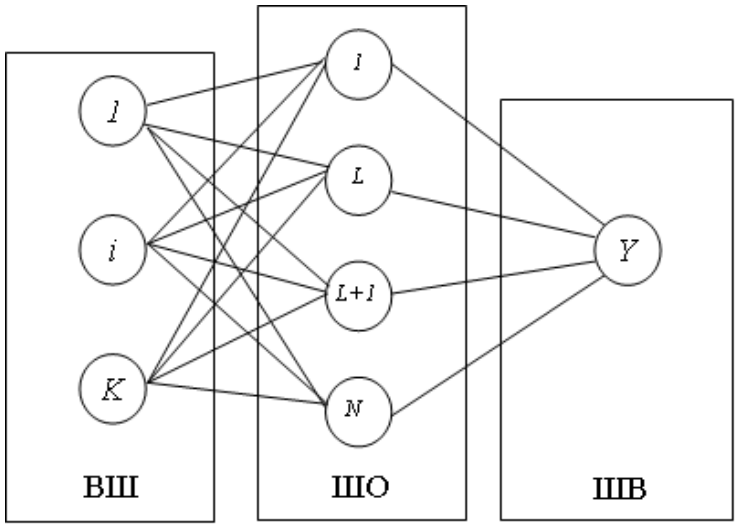


Рис. 3.9. Структура мережі PNN без ШД

Безпосереднє визначення такого правила в PNN неможливе, оскільки лінійна активаційна функція ШО не в змозі відобразити складову

$$p_i \in \left\lfloor P_i^{min}, P_i^{max} \right\rfloor. \tag{3.101}$$

Разом з тим, умову (2.20) можна представити за допомогою системи рівнянь вигляду

$$\begin{cases} p_1 = P_1^{min} \wedge p_2 = P_2^{min} \wedge \ldots \\ p_1 = P_1^{min} + \Delta_1 \wedge p_2 = P_2^{min} + \Delta_2 \wedge \ldots, \\ p_1 = P_1^{max} \wedge p_2 = P_2^{max} \wedge \ldots \end{cases} \tag{3.102}$$

де $\Delta_1, \Delta_2, ...$ – задані коефіцієнти.

Однак використання виразу (3.102) призводить до вагомого ускладнення PNN за рахунок значного збільшення кількості нейронів ШО. Можливим шляхом адаптації моделі PNN до умови (3.102) є введення до її складу проміжного (фільтруючого) шару нейронів, завданням якого буде фільтрація вхідного сигналу, відповідно виразу (3.102). Вказаний фільтруючий шар (ШФ) має знаходитись між ВШ та ШО. Структура модифікованої мережі PNN, що отримала назву MPNN, показана на рис. 3.10.

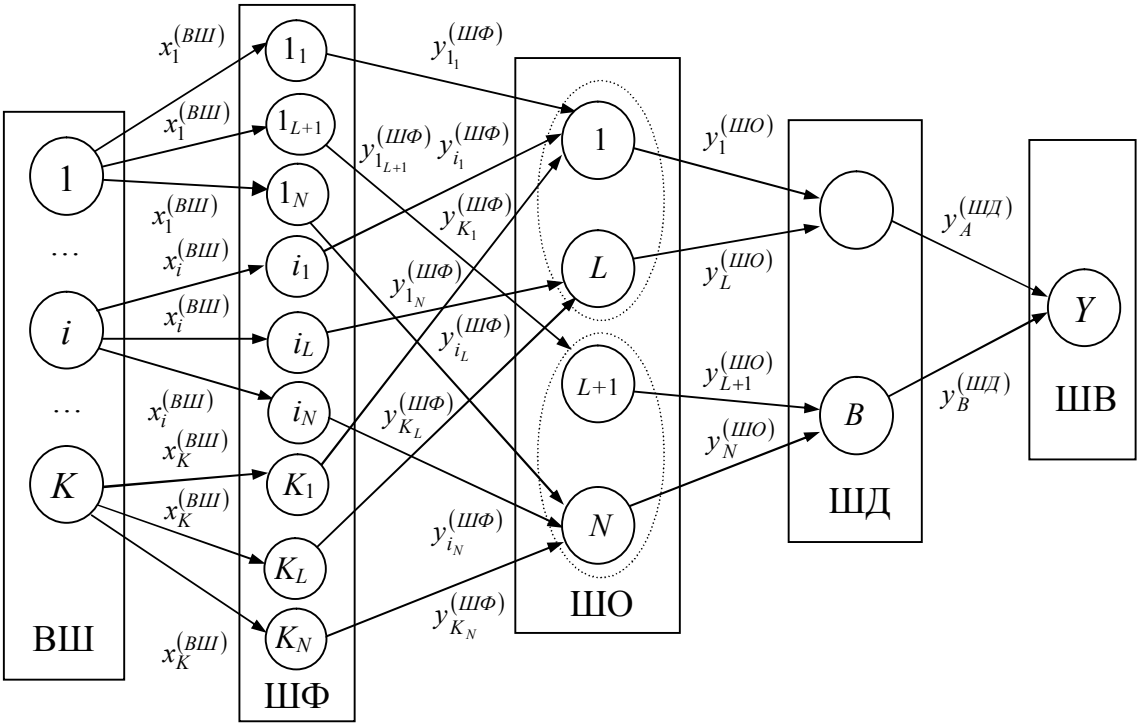


Рис. 3.10. Структура моделі MPNN

Завданням i_l нейрону ШФ є фільтрація i-го ПБ, відповідно l–го продукційного правила. Для цього застосовується функція активації виду:

$$\exists x_i^{(ВШ)} \in \left[P^{min}, P^{max}\right]_l \rightarrow y_{j_l}^{(ШФ)} = x_i^{(ВШ)}, \exists x_i^{(ВШ)} \notin \left[P^{min}, P^{max}\right] \rightarrow y_{j_l}^{(ШФ)} = 0, \quad (3.103)$$

де $x_i^{(ВШ)}$ – значення і-го ПБ,

$y_{j_l}^{(ШФ)}$ – вихідний сигнал j_l нейрону ШФ.

Вихідний сигнал l-го нейрону ШО розраховується так:

$$y_l^{(ШО)} = \sum_{k=1}^{K} exp\left(-\left(w_{k_l,l} - y_{k_l}^{(ШФ)}\right)^2 \Big/ 2\sigma^2\right), \tag{3.104}$$

де $w_{k_l,l}$ – ваговий коефіцієнт зв'язку між k_l-им нейроном ШФ та l-им нейроном ШО,

K – кількість компонент вхідного вектора-образу,

σ – радіус функції Гауса.

В нейронах ШД використовується лінійна функція активації. Вихідний сигнал j-го нейрону ШД ($y_j^{(ШД)}$) розраховується так:

$$y_j^{(ШД)} = \sum_{i=1}^{N} y_i^{(ШО)}, \tag{3.105}$$

де N – кількість нейронів ШО, пов'язаних з j-им нейроном ШД,

$y_i^{(ШО)}$ – активність i-ого нейрону ШО, пов'язаного з j-им нейроном ШД.

Завданням єдиного нейрону ШВ є визначення максимального вихідного сигналу нейронів ШД. Даний нейрон вказує на розпізнаний клас.

3.6. Модель створення ефективних нейромережевих засобів оцінювання параметрів безпеки

Відповідно результатів першого розділу, ефективність НМЗ оцінювання ПБ ІС для виявлення кібератак багато в чому залежить від відповідності типу та параметрів НММ до умов поставленої задачі. При цьому загальну ефективність застосування НМЗ можливо оцінити за

допомогою розроблених критеріїв, визначених виразами (2.10-2.13). Крім того, процес створення НМЗ повинен починатись із визначення принципової можливості їх ефективного використання для розв'язку конкретної задачі розпізнавання кібератак. Також варто звернути увагу на те, що верифікацію розроблених НМЗ слід реалізувати не тільки шляхом окремих чисельних експериментів, які в багатьох випадках не дозволяють гарантувати ефективність виявлення, але і шляхом формалізованих викладок. Базуючись на означених викладках та розроблених підходах до визначення оптимального виду НММ, принципової доцільності та ефективності застосування НМЗ розроблена показана на рис. 3.11 модель створення ефективних НМЗ.

Вхідними даними моделі являються:

$$\boldsymbol{M} = \{m_1,..,m_K\}, \tag{3.106}$$

$$\boldsymbol{O} = \{o_1,...,o_5\}, \tag{3.107}$$

$$\boldsymbol{У} = \{y_1,...,y_6\}, \tag{3.108}$$

де $\boldsymbol{M}$ – множина доступних видів НМЗ,

$\boldsymbol{O}$ – множина характеристик об'єкту захисту,

$\boldsymbol{У}$ – множина умов задачі оцінювання ПБ.

В першому наближенні множина $\boldsymbol{M}$ формується на основі досліджених НММ виду: БШП, РБФ, ТК, АНМ, СНМ, PNN, MPNN, АРТ. Множина $\boldsymbol{O}$ введена в підходах до розпізнавання НК та ПК, а множина $\boldsymbol{У}$ визначена в процесі аналізу можливостей НММ і складається із елементів, що характеризують навчальні дані (y_1), обмеження процесу навчання (y_2), обчислювальні потужності (y_3), вихідну інформацію (y_4), технічну реалізацію (y_5) та сферу застосування НМЗ (y_6). Першочерговий аналіз вхідних параметрів, який відбувається у модулях "Аналіз нейромережевих засобів",

"Допустимі засоби" та "Аналіз умов задачі оцінки ПБ та об'єкту захисту" реалізується на основі обробки експертних даних.

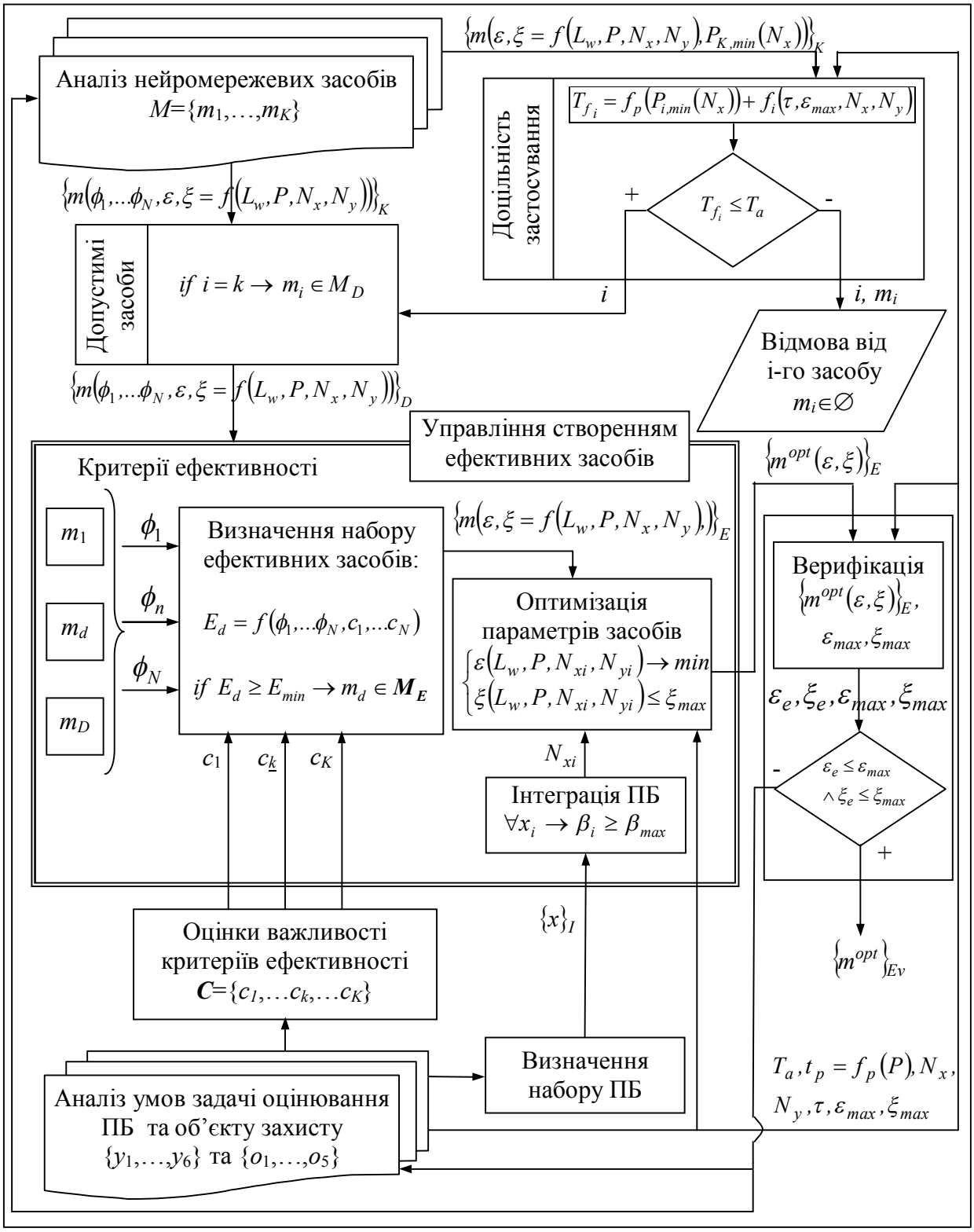


Рис. 3.11. Модель процесу створення ефективних НМЗ виявлення кібератак

В базовому випадку можливо застосовувати метод парних порівнянь, використаний в п.3.1. В подальшому можливо застосовувати інші експертні методи, характеристики яких наведено в [90].

В результаті аналізу елементів $\boldsymbol{M}$ для кожного з них визначаються множина величин критеріїв ефективності ($\boldsymbol{\Phi} = \{\phi_1, \phi_2, ..., \phi_N\}$), залежність мінімальної кількості навчальних прикладів (P_{min}) від кількості вхідних параметрів (N_x) та залежності похибки навчання (ε) і кількості обчислювальних операцій (ξ) від кількості синаптичних зв'язків (L_w), навчальних прикладів (P), кількості вхідних (N_x) та вихідних параметрів (N_y). Таким чином, результатом аналізу кожного $m_k \in \boldsymbol{M}$ є визначення функціоналів виду:

$$P_{min} = f(N_x), \tag{3.109}$$

$$\varepsilon = f(L_w, P, N_x, N_y), \tag{3.110}$$

$$\xi = f(L_w, P, N_x, N_y), \tag{3.111}$$

Крім того, для кожного $m_k \in \boldsymbol{M}$ визначається залежність терміну навчання (T) від тривалості навчальної ітерації (τ), допустимої похибки навчання (ε_{max}), кількості навчальних прикладів, кількості вхідних параметрів і вихідних параметрів:

$$T_k = f_k(\tau, \varepsilon_{max}, P, N_x, N_y), \tag{3.112}$$

В результаті аналізу множин $\boldsymbol{Y}$ та $\boldsymbol{O}$, що відбувається в модулі "Аналіз умов задачі оцінки ПБ та об'єкту захисту" визначається набір ПБ ($\{x_i\}$),

допустимий термін створення НМЗ (T_a), допустима помилка навчання, допустима кількість навчальних ітерацій (ξ_{max}), тривалість однієї навчальної ітерації (τ), кількість вхідних та вихідних параметрів. Також розраховується залежність терміну створення навчальної вибірки (t_p) від кількості навчальних прикладів та визначається множина оцінок важливості критеріїв ефективності ($\boldsymbol{C}$):

$$t_p = f_p(P), \tag{3.113}$$

$$\boldsymbol{C} = \{c_1, c_2, ..., c_K\}, \tag{3.114}$$

де c_k – коефіцієнт важливості *k*-го критерію оцінки ефективності.

Подання величин τ, P, ε_{max}, N_x, N_y, визначених в модулях аналізу, в вираз (3.115) дозволяє розрахувати тривалість терміну розробки параметрів кожного *i*-го НМЗ:

$$T_{f_i} = f_p(P_i) + f_i(\tau, \varepsilon_{max}, N_x, N_y), \tag{3.115}$$

де T_{f_i} – тривалість розробки *i*-го НМЗ.

Рішення про принципову доцільність застосування *i*-го НМЗ приймається у випадку істинності виразу

$$T_{f_i} \le T_a \tag{3.116}$$

де T_a – максимально допустима тривалість розробки НМЗ.

Остаточне формування множини допустимих НМЗ $\{m\}_D$ реалізується в модулі «Допустимі засоби». Вихідні дані даного модулю, що визначаються виразами (3.109-3.113), поступають в модуль «Управління створенням

ефективних засобів», в якому на основі підходу до визначення ефективності НЗМ за допомогою розробленої моделі інтеграції ПБ формується множина ефективних НМЗ з оптимізованими параметрами ($\{m\}_E^{opt}$). Для цього використовується розроблена модель інтеграції ПБ та проводиться оптимізація параметрів визначених ефективних НМЗ.

Для того, щоб визначити приналежність деякого *d*-го НМЗ, що являється елементом $\{m\}_D$, до множини ефективних НМЗ використовуються наступне правило:

$$Якщо\ E_d \geq E_{min} \rightarrow m_d \in \{m\}_E, \tag{3.117}$$

де E_d – ефективність *d*-го НМЗ,

E_{min} – мінімально допустима ефективність НМЗ,

$\{m\}_E$ – множина ефективних НМЗ.

В свою чергу, ефективність *d*-го НМЗ являється функціоналом від критеріїв ефективності та їх вагових коефіцієнтів:

$$E_d = f(\phi_1,...\phi_N, c_1,...c_N). \tag{3.118}$$

Параметри НМЗ, що входять до складу множини ефективних НМЗ, оптимізуються по критерію мінімізації помилки розпізнавання з врахуванням максимально допустимої кількості обчислювальних операцій:

$$\begin{cases} \varepsilon(L_w, P, N_{xi}, N_{yi}) \rightarrow min \\ \xi(L_w, P, N_{xi}, N_{yi}) \leq \xi_{max} \end{cases}, \tag{3.119}$$

де ε – помилка розпізнавання НМЗ,

ξ – кількість обчислювальних операцій для досягнення мінімальної помилки розпізнавання,

ξ_{max} – максимально допустима кількість обчислювальних операцій,

N_{xi} – кількість вхідних параметрів НММ,

N_{yi} – кількість вихідних параметрів НММ.

Множина ефективних НМЗ з оптимізованими параметрами передається до модулю «Верифікації», де на основі порівняння кожного із її компонентів з $\varepsilon_{max}, \xi_{max}$ приймається рішення про можливість використання:

$$Якщо\ (\varepsilon_e \le \varepsilon_{max} \wedge \xi_e \le \xi_{max}) \to m_e \in \{m^{opt}\}_{Ev} \qquad (3.120)$$

де ε_e – помилка розпізнавання для $m_e \in \{m\}_E$,

ξ_e – кількість обчислювальних операцій для $m_e \in \{m\}_E$,

$\{m^{opt}\}_{Ev}$ – множина верифікованих НМЗ.

Множина $\{m^{opt}\}_{Ev}$ являється результуючим виходом моделі.

Зазначимо, що при негативній верифікації слід додатково проаналізувати доступні НМЗ, умови задачі оцінювання ПБ та об’єкт захисту з метою визначення шляхів підвищення ефективності НМЗ.

На відміну від відомих, в описаній моделі передбачено процес інтеграції ПБ на основі експертних даних, визначення принципової доцільності застосування НМЗ, оптимізацію виду та параметрів НМЗ.

3.7. Висновки до третього розділу

В даному розділі вирішувалась науково-практична задача подальшого розвитку моделей нейромережевих засобів оцінювання параметрів безпеки Інтернет-орієнтованих інформаційних систем. В процесі вирішення отримано наступні результати:

– Розроблена модель процесів інтеграції параметрів безпеки, що використовуються нейромережевими засобами розпізнавання кібератак.

Модель базується на створеному підході до розпізнавання поступових та неочікуваних кібератак. В ній, на відміну від відомих, враховано особливості оцінки параметрів безпеки, що використовуються для виявлення поступових та несподіваних кібератак, передбачено використання ланцюгів Маркова для формування шаблонів нормальної поведінки параметрів захищеності, котрі залежать від часу і використовуються для виявлення поступових кібератак, передбачено за рахунок експертного визначення вагомості параметрів безпеки вилучення малоінформативних параметрів. Застосування моделі дозволяє адаптувати множину вхідних параметрів нейромережевих засобів до виду кібератаки та умов використання. Апробація моделі дозволила визначити множину параметрів безпеки для розпізнавання скриптового шкідливого програмного забезпечення, написаного на мові JavaScript.

– Розроблена марківська модель одноперіодичного шаблону поведінки, яка дозволяє проводити апроксимацію параметрів безпеки у випадку їх одноперіодичної нестаціонарності та являється базою для створення марківської моделі багатоперіодичного шаблону поведінки. Основними відмінностями даної моделі є представлення шаблону поведінки у вигляді одноперіодичного ряду динаміки, використання розроблених аналітичних залежностей для розрахунку стаціонарних інтервалів одноперіодичного ряду динаміки та параметрів ланцюгів Маркова, призначених для моделювання вказаних інтервалів.

– Розроблена марківська модель багатоперіодичного шаблону поведінки, що базується на марківській моделі одноперіодичного шаблону та дозволяє створювати шаблони поведінки, адаптовані до типового нестаціонарного характеру параметрів безпеки. Основними відмінностями даної моделі є представлення шалону поведінки у вигляді багатоперіодичного ряду динаміки та використання розроблених аналітичних залежностей для розрахунку стаціонарних інтервалів багатоперіодичного ряду динаміки та параметрів ланцюгів Маркова, призначених для моделювання вказаних інтервалів. Експериментально показано, що

використання розроблених маркіських моделей дозволяє зменшити похибку моделювання шаблону поведінки параметрів безпеки веб-серверу в 1,5-2 рази, що підтверджує ефективність запропонованих рішень.

– – Розроблена модель на основі багатошарового персептрону, яка за допомогою розроблених аналітичних залежностей для оцінки оптимальної кількості схованих нейронів, кількості обчислювальних навчальних операцій, обсягу пам’яті і помилки навчання дозволяє в 1,5-6 рази зменшити ресурсоємність процесу визначення оптимальної структури багатошарового персептрону та апріорно оцінювати його обчислювальні можливості, що є підґрунтям для підвищення ефективності нейромережевих засобів, створених на його основі. Експериментальні дослідження, спрямовані на визначення оптимальної структури багатошарового персептрону, призначеного для апроксимації поліноміальних функцій, підтвердили правильність теоретичних розрахунків. Показано можливість застосування розробленої моделі в нейромережевих засобах, призначених для розпізнавання скриптового шкідливого програмного забезпечення.

– Розроблена модель мережі MPNN, яка, на відміну від класичної мережі PNN, за рахунок введення нейронного шару фільтрації з лінійною біполярною з насиченням функцією активації дозволяє подавати в нейронну мережу навчальні дані у вигляді продукційних правил, що являється базою для забезпечення оперативного розпізнавання нейромережевими засобами нових видів кібератак з недостатнім обсягом статистичних даних.

– Розроблена модель створення ефективних нейромережевих засобів оцінювання параметрів безпеки, в якій, на відміну від відомих, передбачено: процес інтеграції параметрів безпеки на основі експертних даних, визначення принципової доцільності застосування нейромережевих засобів, визначення множини ефективних нейромережевих засобів, оптимізацію їх виду та параметрів. Застосування моделі є підґрунтям для формалізації процесу створення методів розробки ефективних нейромережевих засобів.

РОЗДІЛ 4
МЕТОДИ ПОБУДОВИ НЕЙРОМЕРЕЖЕВИХ ЗАСОБІВ ОЦІНЮВАННЯ ПАРАМЕТРІВ БЕЗПЕКИ ІНТЕРНЕТ-ОРІЄНТОВАНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

4.1. Метод застосування продукційних правил для подання експертних знань

Відправною точкою розробки став описаний в п. 3.5. узагальнений метод подачі навчальних прикладів-продукційних правил для навчання PNN (див. рис. 3.8). Відзначимо, що в процесі навчання PNN визначаються вагові коефіцієнти синаптичних зв'язків та формується структура мережі (ШО та ШД). Оскільки метод застосування продукційних правил для подання експертних знань орієнтовано на розроблену MPNN, що відрізняється від PNN наявністю ШФ (див. рис. 3.10), то узагальнений метод дещо модифікується. Для подання в MPNN нового прикладу-продукційного правила необхідно:

- 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 (3.103), перетворювати сигнали, які передаються від вхідних нейронів до нового нейрону ШО;
- встановити зв'язки між новими нейронами ШО та ШФ;
- встановити зв'язки між новими нейронами ШФ та відповідними вхідними нейронами;
- встановити для нових нейронів ШФ вагові коефіцієнти вхідних

зв´язків рівними величинам параметрів нового навчального прикладу.

Для підвищення універсальності запропонованих рішень передбачено використання методу подання продукційних правил не тільки в складі комплексної методології розробки НМЗ, але й самостійно. Тому процес створення методу базувався на розробленій модель створення ефективних НМЗ оцінки ПБ. Крім того, для визначення множин ПБ та кібератак, на основі яких формуються вхідні параметри НММ та продукційні правила, використано розроблену модель процесу інтеграції ПБ ІС. Це дозволило використати в якості вхідних даних методу множину характеристик об'єкту захисту ІС ($\boldsymbol{O}$), складові якої задані виразом (2.6). Процес формування продукційних правил щодо розпізнавання кібератак реалізується за допомогою розробленого підходу до визначення статистично подібних кібератак. В підсумку метод застосування продукційних правил для подання експертних знань в MPNN, призначену для розпізнавання кібератак на ІС, складається з наступних етапів:

Етап 1 – формування множини можливих кібератак. Етап передбачає аналіз множини характеристик об'єкту захисту ІС, в результаті якого визначається множина кібератак, котру повинна розпізнавати НМ:

$$\boldsymbol{Ka} = \{Ka_1, Ka_2, ..., Ka_J\}, \tag{4.1}$$

де J– кількість кібератак,

Ka_j – j-та кібератака.

Етап 2 – визначення ПБ для розпізнавання довільної кібератаки. На цьому етапі, використовуючи розроблену модель інтеграції ПБ, для кожної Ka_j визначається множина ПБ, котрі будуть використані як вхідні параметри НМЗ оцінювання:

$$Ka_j \to \{Xj\}_{Nj}, \tag{4.2}$$

де Nj – кількість ПБ для розпізнавання j-ої кібератаки.

Етап 3 – <u>визначення подібних кібератак.</u> Етап орієнтований на виділення із $\boldsymbol{Ka}$ статистично подібних між собою кібератак:

$$\left\{\boldsymbol{Ka}_1^{(p)},...,\boldsymbol{Ka}_J^{(p)}\right\}, \forall \boldsymbol{Ka}_1^{(p)} \subset \boldsymbol{Ka},...,\boldsymbol{Ka}_J^{(p)} \subset \boldsymbol{Ka}\ ,\ \boldsymbol{Ka}_1^{(p)} \cup \boldsymbol{Ka}_2^{(p)} ... \cup \boldsymbol{Ka}_J^{(p)} = \boldsymbol{Ka}\ , \quad (4.3)$$

де $\boldsymbol{Ka}_i^{(p)}$ – i-та множина подібних кібератак.

Відповідно розробленого підходу, подібність довільної k-ої та j-ої кібератак визначається кортежем:

$$\left\langle T_{(k,j)}\left(Ka_k, Ka_j\right), R_{(k,j)}\left(\left\{X_k\right\}_{N_k}, \left\{X_j\right\}_{N_j}\right)\right\rangle, \quad (4.4)$$

де $T_{(k,j)}\left(Ka_k, Ka_j\right)$ – функція подібності типу k-ої та j-ої кібератак,

$R_{(k,j)}\left(\left\{X_k\right\}_{N_k}, \left\{X_j\right\}_{N_j}\right)$ – функція подібності множин ПБ, що використовуються для розпізнавання k-ої та j-ої кібератак.

Розрахунок подібності k-ої та j-ої кібератак виконується за п’ять кроків.

Крок 1 – <u>визначення типу кібератак</u>. На цьому кроці визначається тип k-ої та j-ої кібератак:

$$Ka_k \to Ks_k \vee Kq_k, Ka_j \to Ks_j \vee Kq_j\ . \quad (4.5)$$

Крок 2 – <u>розрахунок функції подібності типу кібератак</u>. Для розрахунку порівнюються типи k-ої та j-ої кібератак. Використовується функціонал:

$$Якщо\left(Ks_k \wedge Ks_j\right) \vee \left(Kq_k \wedge Kq_j\right) \to T_{(k,j)} = 0 \neg T_{(k,j)} \neq 0\ . \quad (4.6)$$

Крок 3 – визначення множини спільних ПБ. Даний крок спрямований на визначення множини ПБ, які використовуються для розпізнавання і *k*-ої, і *j*-ої кібератак:

$$\left\{X_{(k,j)}\right\}_{N_{(k,j)}} = \left\{X_k\right\}_{N_k} \cap \left\{X_j\right\}_{N_j}, \tag{4.7}$$

де $N_{(k,j)}$ – кількість спільних ПБ.

Крок 4 – розрахунок коефіцієнту подібності множин ПБ. Для розрахунку коефіцієнту подібності використовується вираз:

$$R_{(k,j)} = N_{(k,j)} \Big/ N_{(k,j)}^{max}, \text{ де } N_{(k,j)}^{max} = N_k \exists N_k > N_j \neg N_{(k,j)}^{max} = N_j. \tag{4.8}$$

Крок 5 – визначення подібності кібератак. На даному кроці приймається остаточне рішення про подібність *k*-ої та *j*-ої кібератак. Кібератаки вражаються подібними, якщо є справедливим вираз:

$$T_{(k,j)} = 0 \wedge R_{(k,j)} \le R_{max}, \tag{4.9}$$

де R_{max} – апріорно заданий коефіцієнт.

Етап 4 – визначення ПБ для розпізнавання подібних кібератак. Етап орієнтований на визначення множини ПБ, що використовуються в якості вхідних параметрів НМ для розпізнавання подібних кібератак:

$$\left\{\boldsymbol{X}_1^{(p)},...,\boldsymbol{X}_J^{(p)}\right\} \rightarrow \boldsymbol{Ka}_j^{(p)}, \tag{4.10}$$

де $\boldsymbol{X}_j^{(p)}$ – множина ПБ, відповідних *j*-ій множині подібних кібератак

$$\boldsymbol{Ka}_j^{(p)} = \left\{Ka_{1,j},...,Ka_{M_j,j}\right\},$$

M_j – кількість елементів $\boldsymbol{Ka}_j^{(p)}$.

Для визначення $\boldsymbol{X}_j^{(p)}$ використовується вираз:

$$\boldsymbol{X}_j^{(p)} = \boldsymbol{X}_{1,j} \cup ... \boldsymbol{X}_{M,j}, \tag{4.11}$$

де $\boldsymbol{X}_{m,j}$ – множина ПБ для розпізнавання $Ka_{m,j}$.

Етап 5 – <u>отримання експертних даних.</u> Даний етап спрямований на формування множин подібних кібератак $\overline{\boldsymbol{Ka}}^{(p)}$, для яких можливо розробити продукційні правила розпізнавання. Якщо для деякої *j*-ої множини $\boldsymbol{Ka}_j^{(p)} \in \boldsymbol{Ka}$ отримати представницькі експертні дані для розробки продукційних правил на основі аналізу $\boldsymbol{X}_j^{(p)}$ достатньо складно, то

$$\boldsymbol{Ka}_j^{(p)} \not\subset \overline{\boldsymbol{Ka}}^{(p)}. \tag{4.12}$$

В протилежному випадку

$$\boldsymbol{Ka}_j^{(p)} \subset \overline{\boldsymbol{Ka}}^{(p)}. \tag{4.13}$$

Етап 6 – <u>розробка множини нейромережевих моделей</u>. Даний етап орієнтований на формування множини НМ типу MPNN, кожна з яких призначена для розпізнавання окремої множини подібних кібератак:

$$\boldsymbol{Net} = \{net_1, net_2, ... net_M\}, \tag{4.14}$$

де net_j – *j*-та НМ, призначена для розпізнавання *j*-ої множини $\boldsymbol{Ka}_j^{(p)}$.

Етап 7 – <u>розробка структури вхідного шару</u>. В результаті виконання

даного етапу для кожної $net_j \in \boldsymbol{Net}$ визначається кількість нейронів у ВШ MPNN. Використовується вираз

$$N_{x,j} = N_j^{max}. \tag{4.15}$$

Також встановлюється відповідність між *i*-им входом НМ та *i*-им ПБ із множини $\boldsymbol{X}_j^{(p)}$.

Етап 8 – розробка продукційних правил. На цьому етапі для кожної множини $\boldsymbol{Ka}_j^{(p)} \in \boldsymbol{Ka}$ на основі експертних даних розроблюється множина продукційних правил їх розпізнавання:

$$\boldsymbol{Pr}_j = \left\{pr_{1,j},...pr_{L_j,j}\right\}, \tag{4.16}$$

де $pr_{i,j}$ – *i*-те продукційне правило для розпізнавання $Ka_{i,j} \in \boldsymbol{Ka}_j^{(p)}$,

L_j – кількість продукційних правил для розпізнавання $\boldsymbol{Ka}_j^{(p)}$.

Продукційні правила задаються виразами виду:

$$x_1 \in \left[x_1^{min}, x_1^{max}\right] \wedge x_2 \in \left[x_2^{min}, x_2^{max}\right] ... \wedge x_{N^{max}} \in \left[x_{N^{max}}^{min}, x_{N^{max}}^{max}\right] \rightarrow Ka_{i,j}, \tag{4.17}$$

де $x_1, x_2 ...$ – інтегровані ПБ,

$\left[x_1^{min}, x_1^{max}\right], \left[x_2^{min}, x_2^{max}\right] ...$ – задані діапазони величин інтегрованих ПБ.

Етап 9 – розробка ШД. На цьому етапі для кожної $net_j \in \boldsymbol{Net}$ в ШД визначається стільки нейронів, скільки подібних кібератак повинна розпізнавати НМ:

$$N_{ШД,j} = M_j. \tag{4.18}$$

Також встановлюється відповідність між кожним *n*-им нейроном ШД та *n*-ою кібератакою:

$$n_{ШД,j} \to Ka_{n,j}. \tag{4.19}$$

Етап 10 – визначення структури ШО та ШФ. Для кожної $net_j \in \boldsymbol{Net}$ виконання етапу являється пристосуванням структури ШО та ШФ MPNN до заданих продукційних правил:

$$\left\langle \boldsymbol{N}_{ШФ}, \boldsymbol{N}_{ШО}, \boldsymbol{L}_{ШФ}, \boldsymbol{L}_{ШО}, \boldsymbol{L}_{ШД} \right\rangle = f\left(Pr_j\right), \tag{4.20}$$

де $\boldsymbol{N}_{ШФ}$ – множина нейронів ШФ,

$\boldsymbol{N}_{ШО}$ – множина нейронів ШО,

$\boldsymbol{L}_{ШФ}, \boldsymbol{L}_{ШО}, \boldsymbol{L}_{ШД}$ – множина вхідних зв'язків ШФ, ШО та ШД.

Визначення довільного *n*-го продукційного правила виконується за п'ять кроків:

Крок 1 – визначення нейрону ШО. На цьому кроці в ШО додається *n*-ий нейрон, який буде відповідати *n*-му продукційному правилу:

$$n_{ШО,j} \to Pr_{n,j}. \tag{4.21}$$

Крок 2 – модифікація $L_{ШД}$. Модифікація полягає у встановленні для *n*-го нейрону ШО зв'язку з нейроном ШД, що відповідає *n*-ій кібератаці.

Крок 3 – визначення нейронів ШФ. На даному кроці в ШФ додається множина нейронів $N_{n,ШФ} \in N_{ШФ}$, що, відповідно *n*-му продукційному правилу, перетворюють сигнали від вхідних нейронів до *n*-го нейрону ШО. Перетворення задається виразом

$$\exists x_i^{(ВШ)} \in \left[P^{min}, P^{max}\right]_l \rightarrow y_{j_l}^{(ШФ)} = x_i^{(ВШ)}, \exists x_i^{(ВШ)} \notin \left[P^{min}, P^{max}\right]_l \rightarrow y_{j_l}^{(ШФ)} = 0, \quad (4.22)$$

де $x_i^{(ВШ)}$ – значення *i*-го ПБ,

$y_{j_l}^{(ШФ)}$ – вихідний сигнал j_l нейрону ШФ.

Крок 4 – модифікація зв'язків $L_{ШО}$. Реалізація кроку полягає у встановленні зв'язків між $n_{ШО,j}$ та $N_{n,ШФ}$.

Крок 5 – модифікація зв'язків $L_{ШФ}$. На цьому кроці встановлюються зв'язки між N_x та $N_{n,ШФ}$.

Кроки 1-5 виконуються для всіх заданих продукційних правил.

Етап 11 – верифікація розроблених MPNN. Верифікація кожної $net_j \in \boldsymbol{Net}$ полягає у порівнянні її похибки розпізнавання та обчислювальної складності з максимально допустимими значеннями цих параметрів. Для кожної net_j похибка розпізнавання та обчислювальна складність розраховуються на прикладах тестової вибірки. Мережа net_j вважається придатною для практичного використання, якщо для всіх прикладів тестової вибірки є справедливим вираз

$$\varepsilon_j \le \varepsilon_{max} \wedge \xi_j \le \xi_{max}. \quad (4.23)$$

де ε_j – похибка розпізнавання net_j,

ξ_j – обчислювальна складність,

ε_{max} – максимально допустима похибка розпізнавання,

ξ_{max} – максимально допустима обчислювальна складність.

Для перевірки ефективності запропонованого методу проведено експериментальні дослідження, в яких MPNN застосовувалась для виявлення мережевих атак. В якості джерела статистичних даних для формування навчальної та тестової множини НМ використана база даних

KDD-99, котра містить близько 5000000 записів – образів мережевих з'єднань [1]. 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.ін. В 42 полі записана інформація, що характеризує стан захищеності ІС – або відсутність атаки (normal), або її тип. В 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). Кількість записів, що відповідають відсутності кібератак, дорівнює 972781.

Оскільки основною передумовою застосування експертних знань в НМ є недостатня повнота навчальних даних, то основну увагу було зосереджено на розпізнаванні кібератак U2R, для яких кількість записів в KDD-99 найменша. З залученням експертів в галузі захисту інформації розроблено 4 продукційних правила для розпізнавання кібератак buffer_overflow, що відносяться до U2R. Приклад продукційного правила для розпізнавання buffer_overflow має наступний вигляд:

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 ∧ 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 ∧ dst_host_same_srv_rate = 0.00 ∧ dst_host_diff_srv_rate = 0 ∧ dst_host_same_src_port_rate = 1 ∧ dst_host_srv_diff_host_rate = від 0 до 0.02 ∧ dst_host_serror_rate = 0 ∧ dst_host_srv_serror_rate = 0 ∧ dst_host_rerror_rate = 0 ∧ dst_host_srv_rerror_rate = 0.

Також розроблено 14 продуційних правил для визначення нормального стану ІС за відсутності кібератаки. Використавши вказані продукційні правила та наведений вище метод, побудовано MPNN, призначену для виявлення кібератак 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 18, а кількість нейронів ШФ дорівнює 738, основу математичного забезпечення складають вирази (3.101-3.105). Структура розробленої MPNN відповідає рис. 3.10, з урахуванням наведених величин.

Апробація розробленої моделі на даних KDD-99 показала абсолютну точність розпізнавання всіх видів кібератак buffer_overflow. Для порівняння отриманих результатів використано робота [3], в якій наведено результати розпізнавання цього ж типу кібератак з використанням сигнатур представлені в базі KDD-99. Для розпізнавання використано БШП і ТК. Точність розпізнавання атак типу buffer_overflow ТК становить 0.0458. При цьому БШП, по причині малого обсягу навчальних даних, взагалі не вдалось навчити розпізнавати кібератак типу buffer_overflow. В роботі [4] для розпізнавання buffer_overflow застосовано спеціальну адаптивну модель, точність розпізнавання якої не перевищує 0,5.

Тому можна вважати, що запропонований метод дозволить підвищити точність розпізнавання кібератак, сигнатури яких не достатньо представлені в базах даних. Також проведене порівняння вказує на те, що застосування запропонованого методу дозволяє підвищити точність розпізнавання кібератак класу U2R, сигнатури яких представлені в базі даних KDD-99, як

мінімум в 2 рази.

Не зважаючи на можливості подання експертних знань, широкому застосуванню модифікованої мережі PNN в галузі захисту інформації заважає недолік – низька здатність узагальнювати навчальну інформацію. Зазначимо, що здатність НМ до узагальнення загальноприйнято оцінювати відношенням кількості синаптичних зв'язків до кількості навчальних прикладів, яку вона може безпомилково або з певною похибкою запам'ятати.

Для мережі PNN одному навчальному прикладу відповідає один нейрон ШО з кількістю синаптичних зв'язків, яка на одиницю перевищує кількість вхідних параметрів. В той же час в БШП з одним вихідним нейроном і такою ж кількістю синаптичних зв'язків співвідноситься 10-100 навчальних прикладів [5, 7, 8]. Тому при розпізнаванні кібератак узагальнюючі можливості БШП в 10-100 вищі, ніж у MPNN. Разом з тим MPNN та БШП мають досить схожі структурні схеми та відносяться до одного класу НМ з прямими розповсюдженням сигналу. Крім того, аналіз [5, 7] вказує на те, що за допомогою конструктивних алгоритмів можливо створити БШП, базою якого являється мережа PNN. Тому перспективною є розробка методу закладення експертних знань в БШП, призначений для розпізнавання мережевих кібератак.

Ще одним важливим напрямком вдосконалення запропонованого методу повинна бути його адаптація до використання експертних знань про мережеві кібератаки, поданих за допомогою апарату нечіткої логіки [4].

4.2. Метод визначення часових характеристик використання нейромережевих засобів

Основою розробки методу являється підхід до визначення принципової доцільності застосування НМЗ оцінювання ПБ, метод застосування продукційних правил для подання експертних знань в НМЗ оцінювання ПБ та модель створення ефективних НМЗ оцінювання ПБ. Крім того,

передбачається можливість застосування методу визначення доцільності як в складі комплексної методології, так і самостійно. Тому в методі застосовано модель процесів інтеграції ПБ, що використовуються НМЗ розпізнавання кібератак. Реалізація методу полягає у виконанні наступних етапів:

Етап 1 – формування множини вхідних та вихідних параметрів НМ. Етап передбачає використання розробленої моделі інтеграції ПБ, що дозволяє на основі аналізу характеристик об'єкту захисту $\boldsymbol{O} = \{o_1, ...o_5\}$ та характеристик кібератаки $\boldsymbol{\Phi} = \{\phi_1, ...\phi_\Phi\}$ визначити множину ПБ, що будуть використані в якості вхідних параметрів НМ:

$$\boldsymbol{X} = \{x_1, ...x_{N_x}\}, \tag{4.24}$$

де N_x – кількість вхідних параметрів НММ.

Також визначається множина вихідних параметрів НММ, що будуть свідчити про наявність/відсутність кібератак певного типу:

$$\boldsymbol{Y} = \{y_1, ...y_{N_y}\}, \tag{4.25}$$

де N_y – кількість вихідних параметрів.

Етап 2 – отримання статистичних даних. В результаті аналізу характеристик об'єкту захисту, умов задачі оцінювання ПБ та множини доступних НМЗ $\boldsymbol{M}=\{m_1,\ldots,m_K\}$ оцінюється можливість реєстрації ПБ, що використовуються для визначення $\boldsymbol{X}$ та $\boldsymbol{Y}$. Якщо оцінка негативна, то НМЗ, крім MPNN, використовувати недоцільно. Тобто

$$\boldsymbol{M} = \{m_{MPNN}\}. \tag{4.26}$$

Етап 3 – подання експертних знань. Використовуючи розроблений метод застосування продукційних правил для подання експертних знань в

НМЗ, оцінюється можливість розробки моделі MPNN. Якщо оцінка негативна, то модель MPNN виключається із подальшого розгляду:

$$m_{MPNN} \notin \boldsymbol{M} . \tag{4.27}$$

В протилежному випадку:

$$m_{MPNN} \in \boldsymbol{M} . \tag{4.28}$$

Етап 4 – перевірка множини допустимих видів НМ. Етап орієнтовано на перевірку не порожності множини допустимих НМЗ. Якщо справедливим є вираз (4.27), то НМЗ використовувати недоцільно:

$$\boldsymbol{M} = \varnothing . \tag{4.29}$$

Етап 5 – визначення допустимої помилки навчання НМ. На даному етапі в результаті аналізу характеристик об'єкту захисту визначаються мінімально допустимі величини пропуску кібератаки та хибного розпізнавання кібератаки. Менша із величин використовується як мінімально допустима помилка навчання НМ. Таким чином, для визначення ε використовується правило

$$\textit{Якщо } \delta_1 \le \delta_2 \to \varepsilon = \delta_1 \textit{ інакше } \varepsilon = \delta_2 . \tag{4.30}$$

де δ_1 – мінімально допустима величина пропуску кібератак,

δ_2 – мінімально допустима величина хибного розпізнавання кібератак,

ε – мінімально допустима помилка навчання НММ.

В першому наближенні $\varepsilon = 0{,}05$.

Етап 6 – визначення часових обмежень процесу розробки НМ. На даному етапі визначається допустима тривалість розробки (T_f) та допустимий термін навчання НМ (t_d). T_f визначається на основі експертного оцінювання. Для визначення t_d використовується залежність

$$P(t_d) \geq P_n, \tag{4.31}$$

де $P(t_d)$ – ймовірність безвідмовної роботи апаратно-програмних засобів, що забезпечують навчання НМ на протязі t_d,

P_n – допустима ймовірність безвідмовної роботи НМЗ.

В першому наближенні $P_n = 10^{-5}$, відповідно (4.29) та [32], $t_d \approx 10^5$ с. Тобто, допустимий термін навчання НМЗ приблизно дорівнює одній добі.

Етап 7 – визначення мінімального обсягу навчальної вибірки.. Даний етап орієнтований на визначення мінімально допустимої кількості навчальних прикладів для НМ. Розрахунок відбувається так:

$$P_{min} = 20N_x. \tag{4.32}$$

де P_{min} – мінімально допустима кількість навчальних прикладів,

N_x – кількість вхідних параметрів НММ.

Етап 8 – розрахунок терміну навчання. На цьому етапі відбувається розрахунок t_{min} – терміну навчання НММ на мінімально допустимій кількості навчальних прикладів. Для ТК, РБФ, PNN та MPNN

$$t_{min} \approx 0{,}1\tau e^{-\varepsilon} P_{min}(N_X + N_Y), \tag{4.33}$$

де τ – тривалість однієї обчислювальної операції процесу навчання.

Для БШП:

$$t_{min} \approx 0{,}001\tau e^{-\varepsilon} P_{min}^2 \left(N_X + N_Y\right)^2 . \quad (4.34)$$

Етап 9 – перевірка терміну навчання. Етап передбачає для всіх НМЗ, що входять до $\boldsymbol{M}$, порівняння допустимого терміну навчання з терміном навчання на мінімально допустимій кількості навчальних прикладів. Входження *j*-го НМЗ до множини ефективних НМЗ з допустимим терміном навчання визначається за допомогою наступного правила:

$$t_{min}\left(m_j\right) \geq t_d \rightarrow m_j \notin \boldsymbol{M}^{(tn)}, \quad (4.35)$$

де $\boldsymbol{M}^{(tn)}$ – множина ефективних НМЗ з допустимим терміном навчання,

m_j – *j*-ий НМЗ.

В протилежному випадку $m_j \in \boldsymbol{M}^{(tn)}$. Якщо $\boldsymbol{M}^{(tn)} = \varnothing$, то НМЗ взагалі використовувати недоцільно.

Етап 10 – розрахунок максимально допустимої тривалості формування навчальної вибірки. На даному етапі для кожної $m_j^{(tn)} \in \boldsymbol{M}^{(tn)}$ розраховується максимально допустима тривалість формування навчальної вибірки ($T_{j,max}$). Зазначимо, що оскільки в MPNN в якості навчальних прикладів використовуються експертні дані, то для НМЗ на базі цієї НММ тривалість формування навчальної вибірки дорівнює тривалості розробки продукційних правил. Для розрахунку $T_{j,max}$ використовується вираз

$$T_{j,max} = T_f - t_j , \quad (4.36)$$

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

Результатом виконання етапу є сформована множина

$$\{T_{1,max},..,T_{L,max}\}, \quad (4.37)$$

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

Етап 11 – визначення терміну формування навчальної вибірки. Етап орієнтовано на аналіз характеристик об'єкту захисту та умов задачі оцінювання для визначення терміну формування навчальної вибірки з мінімально допустимою кількістю начальних прикладів:

$$T_d = f(\boldsymbol{O},\boldsymbol{Y}), \quad (4.38)$$

де T_d – термін формування навчальної вибірки.

Етап 12 – перевірка терміну формування навчальної вибірки. На даному етапі для кожної $m_j^{(tn)} \in \boldsymbol{M}^{(tn)}$ проводиться порівняння $T_{j,max}$ та T_d. Множина НМЗ, які доцільно використовувати для оцінки ПБ, формується за допомогою виразів (4.39, 4.40).

$$Якщо\ T_{j,max} > T_d \rightarrow m_j^{(tn)} \notin \boldsymbol{Mz}, \quad (4.39)$$

$$Якщо\ T_{j,max} < T_d \rightarrow m_j^{(tn)} \in \boldsymbol{Mz}, \quad (4.40)$$

де $\boldsymbol{Mz}$ – множина НМЗ, які доцільно використовувати для оцінки ПБ.

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

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 відомі 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 етапів 10-12 слід оцінити можливість формування мінімальної навчальної вибірки на основі доступних баз даних образів кібератак.

Розглянемо використання методу на конкретному прикладі розпізнавання кібератак типу IP-спуфінг.

Етап 1. Відповідно [267], 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. Номенклатура вхідних параметрів НМ буде відповідати вказаним функціональним параметрам. Кількість вхідних параметрів $N_X = 5$. Для виявлення кібератаки 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 2. Реєстрацію наведених ПБ можливо здійснити мережевими екранами та СВА. При цьому в якості доступної бази даних можливо використати KDD-99.

Етап 3. В першому наближенні мережа MPNN виключена із розгляду. Тому $m_{MPNN} \notin \boldsymbol{M}$.

Етап 4. Оскільки результат другого етапу позитивний, то $\boldsymbol{M} \neq \varnothing$.

Етап 5. Базуючись на результатах [35], визначено, що допустима помилка навчання НМ $\varepsilon = 0{,}05$.

Етап 6. На основі експертного оцінювання визначено, що термін, на протязі якого ризик від реалізації кібератаки не перевищує встановлену

межу, становить $T_a = 30$ діб. Підставивши отриману величину в вираз (2.103) отримано $T_f \le 30$. Таким чином, максимально допустима тривалість розробки НМ становить 30 діб. Відповідно [13, 15], допустимий термін навчання НМ становить $t_d \approx 10^5$ с (24 години).

Етап 7. Підставивши $N_X = 5$ в вираз (4.32), отримаємо:

$$P_{min} \ge (10..20) \times N_x = 20 \times 5 = 100. \qquad (4.41)$$

Таким чином, мінімальна кількість навчальних прикладів дорівнює $P_{min} = 100$.

Етап 8. Для розрахунку терміну навчання НММ виду ТК, РБФ, PNN на мінімально допустимій навчальній вибірці в вираз (4.33) підставлено $P_{min} = 100$, $N_X = 5$, $N_Y = 1$, $\chi = 1$, $\tau = 10^{-2}$. Отримано:

$$t_{min} \approx 0{,}1\tau e^{-\varepsilon} P_{min}(N_X + N_Y) = 0{,}1 \times 10^{-2} \times e^{-1 \times 0{,}05} \times 100 \times (5+1) = 5{,}71. \qquad (4.42)$$

Таким чином, термін навчання ТК, РБФ, PNN на мінімально допустимій навчальній вибірці становить 5,71 с.

Для розрахунку терміну навчання БШП ці ж дані підставлено в вираз (4.33). Отримано:

$$t_{min} \approx \mu_2 \tau e^{-\chi\varepsilon} P^2 (N_X + N_Y)^2 = 0{,}001 \times 10^{-2} \times e^{-1 \times 0{,}05} \times 100^2 \times (5+1)^2 = 34{,}24 \qquad (4.43)$$

Тому термін навчання БШП на мінімально допустимій навчальній вибірці становить 34,24 с.

Етап 9. Підстановка термінів навчання БШП, ТК, РБФ, PNN в вираз (4.35) вказує на входження вказаних НММ до множини $\boldsymbol{M}^{(tn)}$. Відповідно [256], допустимий термін навчання НММ становить $t_d \approx 10^5$ с (24 години).

Етап 10. В якості доступної бази даних зареєстрованих ПБ можливо використати KDD-99. Відповідно, $\boldsymbol{Mz} = \boldsymbol{M}^{(tn)}$. Таким чином, в першому наближенні для розпізнавання кібератак типу IP-спуфінг можливо використовувати НМЗ на базі НММ виду БШП, ТК, РБФ та PNN.

З використанням спрощеного варіанту запропонованого методу проведено визначення доцільності застосування НМЗ для виявлення типових кібератак на ІС. Аналіз [253, 255, 256] дозволяє стверджувати, що в сучасних умовах в якості доступних джерел статистичних даних ПБ, призначених для формування начальної вибірки НМ, доцільно використовувати параметри, котрі реєструються операційними системами, мережевими серверами та такими СЗІ, як міжмережеві екрани, антивіруси, системи захисту від спаму та DLP-системи.

Для кожної ІС ведуться відповідні бази даних, куди записуються зареєстровані величини вказаних ПБ. Крім того, бази даних СЗІ містять величини ПБ, які сигналізують про відсутність/наявність атаки на ІС. Тому, використавши наведену в [93] класифікацію, в першому наближені визначено, що номенклатура та обсяг зареєстрованих статистичних параметрів дозволяють сформувати навчальну вибірку НМ, призначених для виявлення наступних видів кібератак:

– мережевих кібератак, що реалізуються на транспортному та прикладному рівнях стека протоколів TCP/IP;

– ШПЗ – комп’ютерних вірусів та троянів;

– спаму;

– витоків текстової інформації за допомогою листів електронної пошти.

Очевидно, що визначений перелік кібератак потребує подальшої деталізації, реалізувати яку в повному обсязі заважає велика кількість відомих підтипів атак, постійна поява нових підтипів атак та необхідність використання експертних даних при визначенні максимально допустимої тривалості розробки НМЗ. Тому реалізована тільки часткова деталізація.

Використавши спрощений варіант методу, розглянуто доцільність використання НММ для виявлення кібератак наступних типів: СП, Dos-атак, веб-орієнтованих скриптових вірусів та троянів (ВСВ) та несанкціонованого отримання прав доступу незареєстрованим користувачем (R2L). Розглядались такі підтипи R2L, як phf та multihop [121]. В якості вхідних параметрів НМ використано:

– СП, Dos-атаки типу neptune та типу smurf – кількість одночасних підключень, швидкість обробки запитів, кількість пакетів з однаковими адресами відправника та адресата, кількість віртуальних каналів;

– ВСВ – назви потенційно небезпечних операторів мови програмування JavaScript;

– R2L – параметри мережевих з'єднань (тривалість, тип протоколу, мережевий сервіс, кількість отриманих байтів, кількість переданих байтів, статус з'єднання і т.ін.).

Результати розрахунків показані на рис. 4.1 та наведені в табл. 4.1.

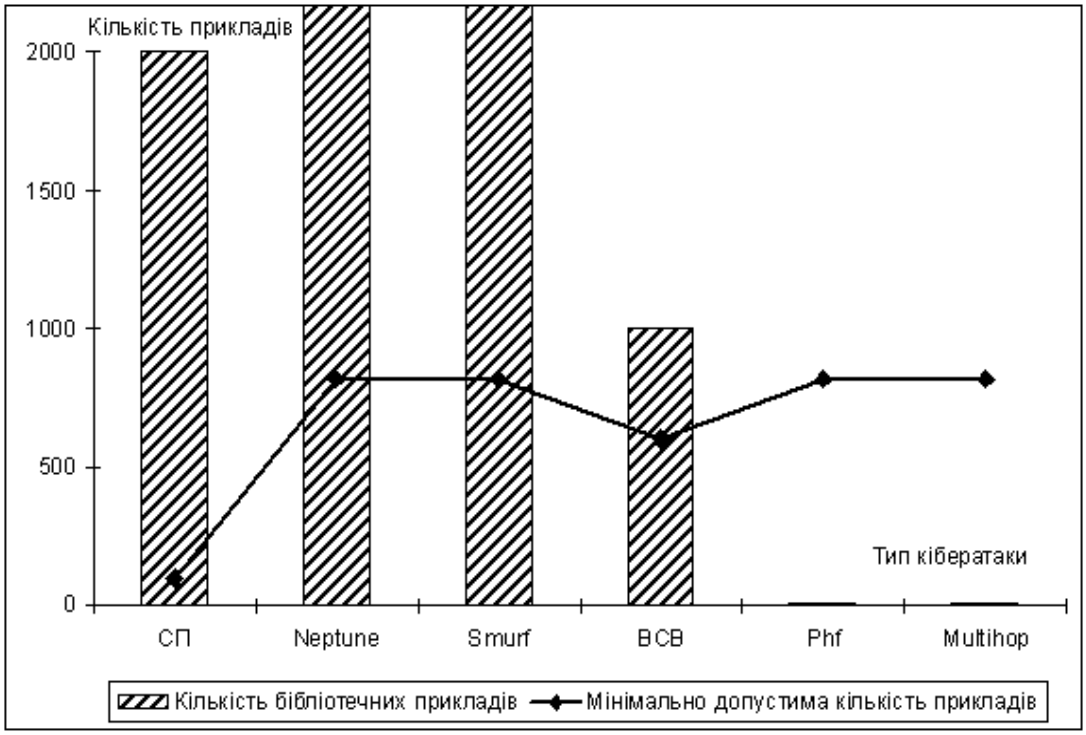


Рис. 4.1 Оцінка можливості застосування нейромережевих засобів для виявлення кібератак

Таблиця 4.1

Проміжні результати оцінювання

Етап оцінювання	Вид кібератаки					
	СП	Dos-атак		ВСВ	R2L	
		Neptune	Smurf		Phf	Multihop
Кількість вхідних параметрів	5	41		30	41	
Допустима помилка навчання	0,05					
Тип НМ	БШП					
Мінімальна кількість навчальних прикладів	100	820		600	820	
Допустимий термін навчання, с	$\approx 10^5$					
Тривалість однієї навчальної ітерації, с	$\approx 10^{-2}$					
Термін навчання, с	3,5	112,8		32,9	112,8	
Оцінка можливості навчання за допустимий термін	Позитивна					
Приблизна кількість прикладів кібератак в базі даних	2000	10^6	3×10^6	10^3	4	7
Оцінка можливості формування навчальної вибірки на основі бази даних	Позитивна				Негативна	

Аналіз даних табл. 4.1 та рис. 4.1 дозволяє позитивно оцінити можливість навчання НММ на мінімально допустимій кількості навчальних прикладів за допустимий термін навчання та можливість формування мінімальної навчальної вибірки на основі існуючих баз даних для кібератак типу СП, Neptune, Smurf та ВСВ. Для кібератак типу phf та multihop оцінка можливості формування мінімальної навчальної вибірки на основі існуючих баз даних негативна.

Тому, відповідно спрощеного варіанту запропонованого методу, НМЗ

на основі БШП, ТК, РБФ та PNN доцільно використовувати для розпізнавання кібератак типу СП, Neptune, Smurf та ВСВ і недоцільно – для розпізнавання атак типу phf та multihop.

4.3. Метод проектування шаблону поведінки параметрів безпеки

Метод проектування шаблону поведінки ПБ базується на запропонованому підході до розпізнавання ПК та розроблених марківських моделях ШП. Відповідно даного методу, розробка ШП ПБ складається із п'яти етапів:

Етап 1 – вирівняння ряду. Етап орієнтований на розрахунок вирівняного ряду зареєстрованих статистичних даних:

$$\hat{X}(t) = X'(t) - Y(t) - \overline{X},\ t \in [0,T], \tag{4.44}$$

де $X'(t)$ – ряд даних,

$Y(t)$ – тренд,

$\overline{X}$ – середнє значення ряду.

Етап 2 – розрахунок параметрів ЛМ. На етапі розраховуються параметри ЛМ, призначені для моделювання складових періодичного ряду:

$$\left\langle \{AB\}_K, \{BA\}_K, \left\{\left|p^{(AB)}\right|\right\}_K, \left\{\left|p^{(BA)}\right|\right\}_K \right\rangle, \tag{4.45}$$

де $\{AB\}_K, \{BA\}_K$ – множини розроблених стаціонарних інтервалів для кожного із значимих періодів,

K – кількість періодів,

$\left\{\left|p^{(AB)}\right|\right\}_K, \left\{\left|p^{(BA)}\right|\right\}_K$ – перехідні ймовірностей для кожного із ЛМ.

Структурна схема розрахунку параметрів ЛМ показана на рис.4.2.

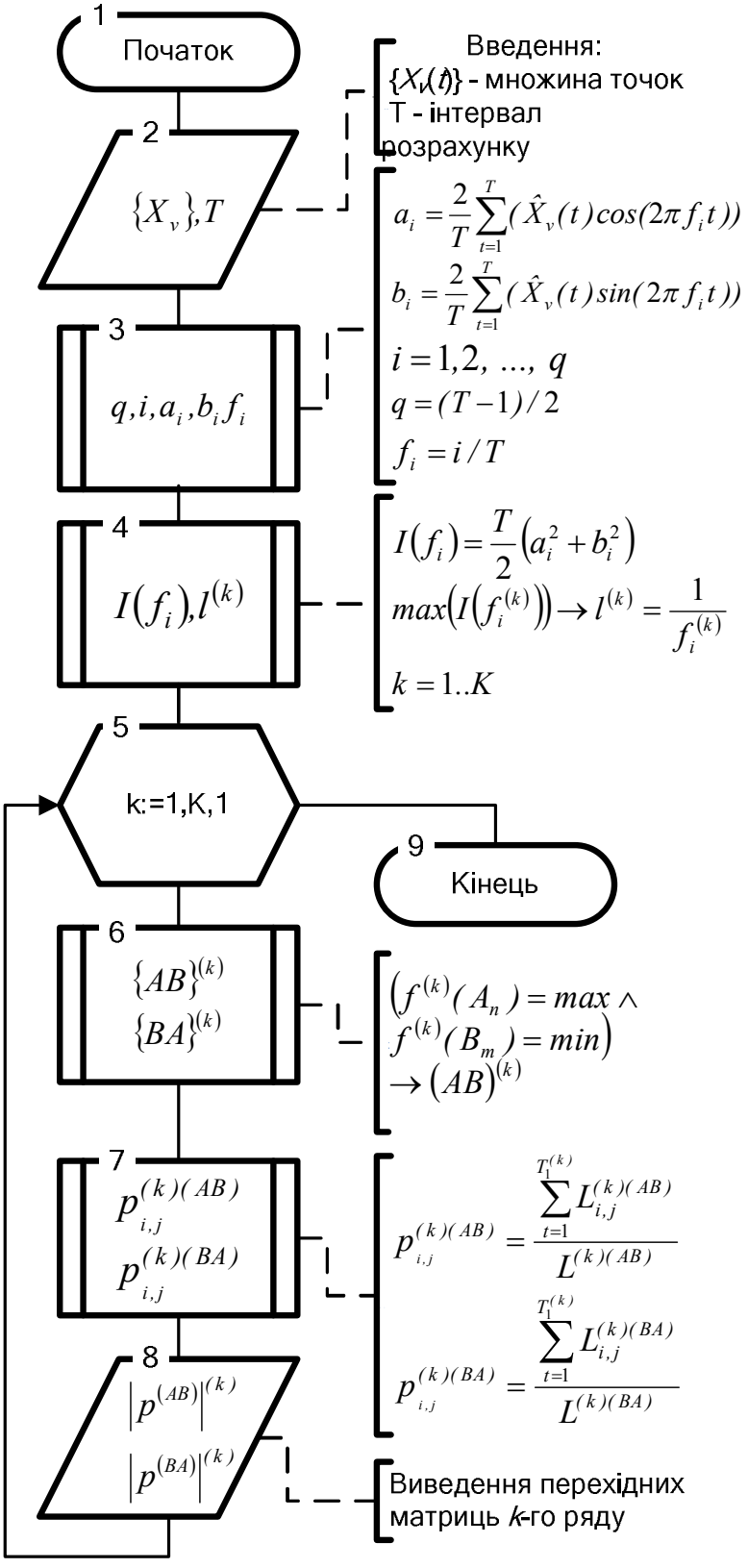


Рис. 4.2. Схема розрахунку параметрів ЛМ

Не враховуючи процедури вводу та виводу даних, розрахунок

параметрів ЛМ реалізується за чотири кроки, що відповідають 3-7 вершинам структурної схеми:

Крок 1 – розрахунок періодичних складових. На цьому кроці, що відповідає третій вершині рис.4.2, за допомогою спектрального аналізу методом Фур'є розраховують періодичні складові вирівняного ряду:

$$\hat{X}(t) = a_0 + \sum_{i=1}^{q} \left(a_i c_i(t) + b_i s_i(t)\right) + e(t), \tag{4.46}$$

де a_0 – коефіцієнт,

$c_i(t) = cos(2\pi f_i t)$,

$s_i(t) = sin(2\pi f_i t)$,

$f_i = i / T$ – i-а гармоніка основної частоти $1/T$,

a_i, b_i – коефіцієнти регресії, що вказують на ступінь кореляції функцій $c_i(t)$ та $s_i(t)$ з статистичними даними,

$q = (T-1)/2$,

T – кількість точок ряду,

$e(t)$ – випадкова складова,

$2\pi f_i$ – кругова частота.

Розрахунок періодичних складових полягає у визначенні коефіцієнтів регресії:

$$a_i = \frac{2}{T} \sum_{t=1}^{T} \left(\hat{X}(t) c_i(t)\right), i = 1,2,...q, \tag{4.47}$$

$$b_i = \frac{2}{T} \sum_{t=1}^{T} (\hat{X}(t) s_i(t)), i = 1,2,...q, \tag{4.48}$$

Крок 2 – визначення значимих періодів. Виконання кроку, що

відповідає четвертій вершині рис.4.2, орієнтоване на визначення значимих періодів вирівняного ряду $\hat{X}(t)$. Для цього на відрізку $[0,T]$ за допомогою формули (4.49) будують періодограму вирівняного ряду даних

$$I(f_i) = T \times \left(a_i^2 + b_i^2\right)/2 . \qquad (4.49)$$

Найбільші величини періодограми відповідають значимим періодам процесу:

$$max\left(I\left(f_i^{(k)}\right)\right) \to l^{(k)} = 1/f_i^{(k)}, k = 1,2,...K , \qquad (4.50)$$

де K – кількість значимих періодів,

$f_i^{(k)}$ – i-та частота ряду даних, що відповідає k-му значимому періоду,

$l^{(k)}$ – k-ий значимий період.

Якщо кількість значимих періодів $K = 1$, то використовується марківська модель одноперіодичного ШП. Якщо кількість значимих періодів $K > 1$, то використовується марківська модель багатоперіодичного ШП.

Крок 3 – визначення нестаціонарних точок. Робота даного кроку, якому відповідає вершина 6 рис. 4.2, полягає у визначенні нестаціонарних точок в межах кожного k-го періоду ($k \in 1,...,K$). Для цього розраховують значення t, які відповідають точкам максимумів та мінімумів функції $f_k(t)$:

$$\left(f^{(k)}(A_n) = max \wedge f^{(k)}(B_m) = min\right) \to (AB)^{(k)} \forall\ n = 1...N, m = 1...M, k = 1...K. \qquad (4.51)$$

де $\{A\}_N$ – множина точок максимумів $f_k(t)$,

$\{B\}_M$ – множина точок мінімумів $f_k(t)$.

Крок 4 – розрахунок ймовірностей переходів. На цьому кроці, якому

відповідає вершина 7 рис.4.2, розраховують матриці ймовірностей переходів. Статистичні дані на інтервалах *AB* використовують для розрахунку $\left|p^{(AB)}\right|$, а статистичні дані на інтервалах *BA* – для $\left|p^{(BA)}\right|$. Для розрахунку ймовірностей переходів із стану *i* в *j* на *AB* та *BA* кожного з *k*-ого негомогенного ЛМ, що відповідає *k*-му виділеному одноперіодичному ряду використовують вирази:

$$p_{i,j}^{(k)(AB)} = \sum_{t=1}^{T_1} L_{i,j}^{(k)(AB)} \Big/ L^{(k)(AB)}, \tag{4.52}$$

$$p_{i,j}^{(k)(BA)} = \sum_{t=1}^{T_2} L_{i,j}^{(k)(BA)} \Big/ L^{(k)(BA)}, \tag{4.53}$$

де $p_{i,j}^{(k)(AB)}$ – ймовірність переходу між станами *i* та *j* на інтервалах типу *BA* на *k*–му періоді,

$p_{i,j}^{(k)(BA)}$ – ймовірність переходу між станами *i* та *j* на інтервалах типу *AB* на *k*–му періоді,

$L^{(k)(AB)}$ – загальна кількість реалізацій, що перейшли із стану *i* в *j* між *t*-ою та (*t*-1)-ою реєстрацією на *AB* на *k*–му періоді,

$L^{(k)(BA)}$ – загальна кількість реалізацій, що перейшли із стану *i* в *j* між *t*-ою та (*t*-1)-ою реєстрацією на *BA* на *k*–му періоді,

T_1 – кількість реєстрацій на інтервалах типу *AB*,

T_2 – кількість реєстрацій на інтервалах типу *BA*.

Етап 3 – розрахунок ймовірностей станів ЛМ. На даному етапі для кожного *k*-го ЛМ розраховують ймовірність і-го стану в довільний момент часу ($P_i^{(k)}(t)$). Для цього матриці ймовірностей переходів $\left|p^{(AB)}\right|$ та $\left|p^{(BA)}\right|$ підставляються в систему рівнянь Колмогорова-Чепмена виду (3.39) та (3.40).

Етап 4 – розрахунок ймовірностей станів ММ. На даному етапі для

марківської моделі розраховують ймовірність кожного із станів. Використовують вираз:

$$P_i(t) = \frac{1}{K} \sum_{k=1}^{K} P_i^{(k)}(t), \tag{4.54}$$

де $P_i(t)$ – ймовірність i-го стану марківської моделі в момент часу t,

$P_i^{(k)}(t)$ – ймовірність i-го стану для k-го ЛМ в момент часу t.

Етап 5 – розрахунок поведінки. На цьому етапі для заданого t розраховується очікуване значення ПБ:

$$\hat{X}(t) = M(t) + Y(t) + \overline{X}, \tag{4.55}$$

де $M(t)$ – математичне сподівання ПБ, розраховане за допомогою побудованої марківської моделі.

Розроблений метод став основою для створення прикладної програми MarkPr, призначеної побудови комп'ютерних моделей ШП ПБ ІС. Розробка спеціалізованої програми викликана особливістю досліджень, які складно виявити за допомогою універсальних розрахункових програм.

Програма MarkPr адаптована до моделювання нормалізованих ПБ, базується на виразах (3.38-3.36, 4.44-4.55) та дозволяє визначити основні параметри гомогенних (однорідних), негомогенних (неоднорідних), поглинаючих та ергодичних ЛМ. Результати розрахунків записуються в файл та виводяться на екран комп'ютера у вигляді графіку математичного сподівання ПБ і графіків ймовірностей розподілу ПБ по станам в точках:

$$0{,}2 \times T;\ 0{,}4 \times T;\ 0{,}6 \times T;\ 0{,}8 \times T;\ T, \tag{4.56}$$

де T – інтервал розрахунку.

В тих же точках виводяться величини ймовірностей розподілу по станам. Графік математичного сподівання показано напівжирною лінією, а графіки розподілу ймовірностей показано звичайними лініями. При побудові, показаних графіків розподілу ймовірностей вважається, що в межах стану ПБ вважається розподіленим по нормальному закону. Сірим кольором виділено площу розподілу, обмежену кожним із графіків розподілу ймовірностей. Для зручності вертикальна вісь рисунків має два ряди позначок – величину приведеного ПБ (x) та номер стану ЛМ (N). Прийнято, що величини станів, на які розділяється діапазон можливих значень параметру, рівні між собою. Інші параметри ЛМ – кількість квантів, ймовірності (інтенсивності) переходів, термін функціонування та початковий розподіл ймовірностей перебування, можуть бути довільно задані.

Для прикладу, на рис. 4.3 та рис. 4.4 показано графіки математичного сподівання та розподілу ймовірностей перебування марківської моделі ШП на протязі 600 етапів розрахунку. Рис. 4.3 відповідає стаціонарному ШП, що моделювався ЛМ з матрицею ймовірностей переходів

$$\pi = \begin{vmatrix} 0{,}97 & 0{,}01 & 0{,}01 & 0{,}01 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0{,}01 & 0{,}96 & 0{,}01 & 0{,}01 & 0{,}01 & 0 & 0 & 0 & 0 & 0 \\ 0{,}01 & 0{,}01 & 0{,}95 & 0{,}01 & 0{,}01 & 0{,}01 & 0 & 0 & 0 & 0 \\ 0{,}01 & 0{,}01 & 0{,}01 & 0{,}94 & 0{,}01 & 0{,}01 & 0{,}01 & 0 & 0 & 0 \\ 0 & 0{,}01 & 0{,}01 & 0{,}01 & 0{,}94 & 0{,}01 & 0{,}01 & 0{,}01 & 0 & 0 \\ 0 & 0 & 0{,}01 & 0{,}01 & 0{,}01 & 0{,}94 & 0{,}01 & 0{,}01 & 0{,}01 & 0 \\ 0 & 0 & 0 & 0{,}01 & 0{,}01 & 0{,}01 & 0{,}94 & 0{,}01 & 0{,}01 & 0{,}01 \\ 0 & 0 & 0 & 0 & 0{,}01 & 0{,}01 & 0{,}01 & 0{,}95 & 0{,}01 & 0{,}01 \\ 0 & 0 & 0 & 0 & 0 & 0{,}01 & 0{,}01 & 0{,}01 & 0{,}96 & 0{,}01 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0{,}01 & 0{,}01 & 0{,}01 & 0{,}98 \end{vmatrix} \quad (4.57)$$

Графік рис.4.4 відповідає марківській моделі двохперіодичного ШП. В марківській моделі використано два ЛМ з матрицями переходів заданих виразами (4.57, 4.58). Зазначимо, що матриця (4.58) отримана з (4.57) шляхом

подвоєння величин ймовірностей переходів із *i*-го стану в (*i*-1)-й, (*i*-2)-й та (*i*-3)-й стани.

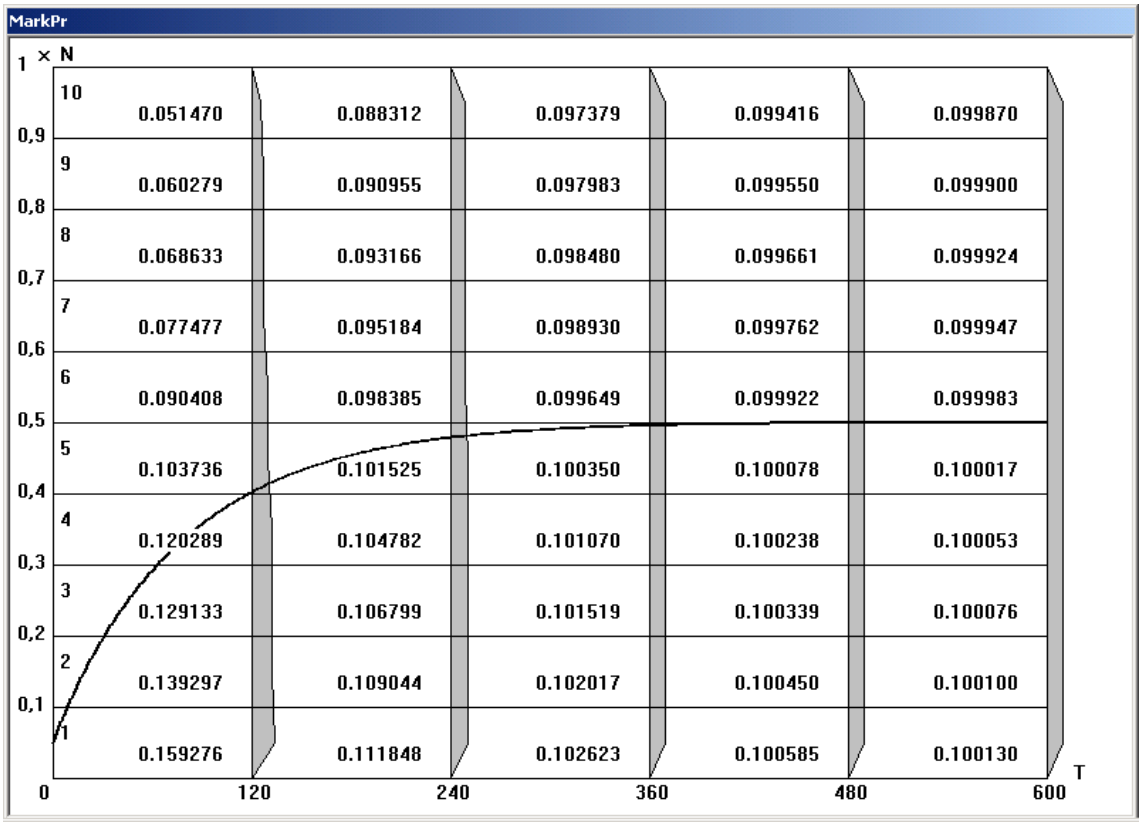


Рис. 4.3. Параметри стаціонарного ЛМ характерного ШНП

$$\pi = \begin{vmatrix} 0{,}97 & 0{,}01 & 0{,}01 & 0{,}01 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0{,}02 & 0{,}96 & 0{,}01 & 0{,}01 & 0{,}01 & 0 & 0 & 0 & 0 & 0 \\ 0{,}02 & 0{,}02 & 0{,}95 & 0{,}01 & 0{,}01 & 0{,}01 & 0 & 0 & 0 & 0 \\ 0{,}02 & 0{,}02 & 0{,}02 & 0{,}94 & 0{,}01 & 0{,}01 & 0{,}01 & 0 & 0 & 0 \\ 0 & 0{,}02 & 0{,}02 & 0{,}02 & 0{,}94 & 0{,}01 & 0{,}01 & 0{,}01 & 0 & 0 \\ 0 & 0 & 0{,}02 & 0{,}02 & 0{,}02 & 0{,}94 & 0{,}01 & 0{,}01 & 0{,}01 & 0 \\ 0 & 0 & 0 & 0{,}02 & 0{,}02 & 0{,}02 & 0{,}94 & 0{,}01 & 0{,}01 & 0{,}01 \\ 0 & 0 & 0 & 0 & 0{,}02 & 0{,}02 & 0{,}02 & 0{,}95 & 0{,}01 & 0{,}01 \\ 0 & 0 & 0 & 0 & 0 & 0{,}02 & 0{,}02 & 0{,}02 & 0{,}96 & 0{,}01 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0{,}02 & 0{,}02 & 0{,}02 & 0{,}98 \end{vmatrix} \quad (4.58)$$

Нестаціонарні точки марківській моделі двохперіодичного ШП

відповідають t_1=121, t_2=241, t_3=361, t_4=481 етапам розрахунку.

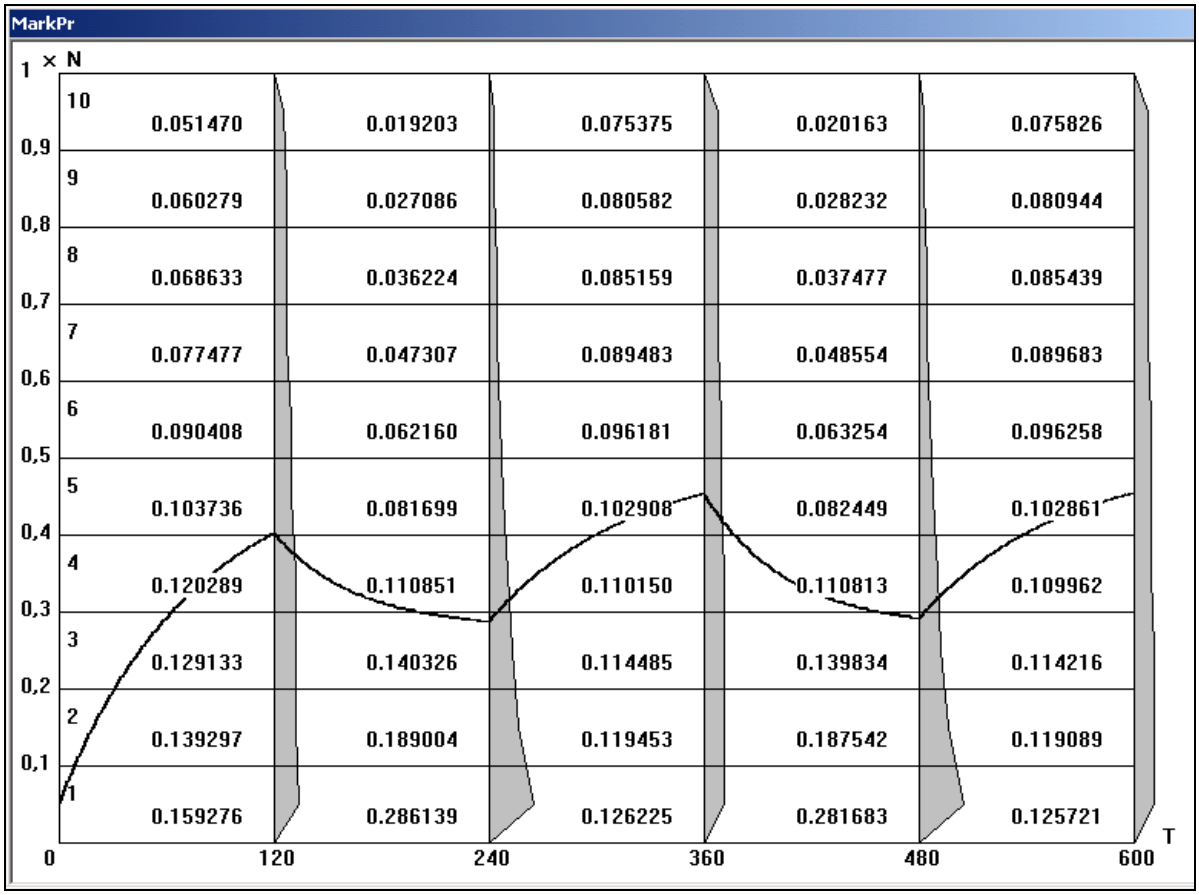


Рис. 4.4. Графіки параметрів одноперіодичної марківської моделі

Приклад використання методу для розробки одноперіодичної та двохперіодичної марківської моделі ПБ веб-серверу наведено в п. 3.3.

Зазначимо, що застосування марківської моделі ШП повинно враховувати специфіку захисту ресурсів Інтернет-орієнтованих ІС та традиційних сфер застосування ШП. Найбільш перспективним та важливим є застосування ШП для розпізнавання мережевих кібератак на веб-сервери з метою викликати відмову в обслуговуванні та/або отримати несанкціонований доступ до ресурсів ІС. Хоча не слід виключати і можливість розпізнавання активності ШПЗ. Однак останній напрям потребує серйозного доопрацювання. Тому при формуванні множини ПБ для побудови ШП слід розглядати тільки ті контрольовані параметри ІС, що можуть

вказувати на зміну рівня захищеності їх ресурсів внаслідок реалізації мережевих кібератак. Ці параметри повинні відповідати характеристикам працездатного стану компонентів ресурсів та/або характеристикам мережевої кібератаки. До першого класу параметрів можливо віднести завантаження центрального процесора, довжину черги мережевих запитів, кількість відкритих файлів, завантаження оперативної пам'яті, вільний обсяг жорсткого диску, кількість процесів, що відбуваються в операційній системі. Зазначимо, що цей клас параметрів в основному дозволяє діагностувати зміну рівня захищеності мережевих ресурсів ІС внаслідок реалізації атаки з метою викликати відмову в обслуговуванні. До другого класу параметрів відносяться: інтенсивність загального, вхідного та вихідного мережевого трафіку та трафіку по кожному із мережевих протоколів (SNMP, ARP, TCP, IP, ICMP, HTTP), інтенсивність мережевих запитів різного типу, відношення правильних та неправильних мережевих пакетів за одиницю часу, кількість правильних та неправильних спроб вводу парольних даних. Цей клас параметрів можливо застосовувати для розпізнавання атак на відмову в обслуговуванні та з метою отримання несанкціонованого доступу. Доцільно використовувати зареєстровані, фільтровані та приведені значення параметрів. Наприклад, кількість TCP/IP запитів (пакетів) за одиницю часу з однієї IP-адреси або з однієї підмережі. Остаточний перелік ПБ повинен базуватись на специфіці об'єкту захисту. Наприклад, при формуванні номенклатури діагностичних параметрів для веб-серверу можливо враховувати географію запитів, яку легко визначити на основі зворотної IP-адреси. Відповідний параметр – інтенсивність запитів до веб-серверу з обмеженої географічної зони, міг бути досить інформативним при розпізнаванні атаки на відмову в обслуговуванні державних веб-сайтів України, що стались 2007-2012 роках. Адже, по свідченням засобів масової інформації, вказана кібератака була реалізована з трьох достатньо обмежених географічних зон. ШНП та ША можливо сформувати на основі одних і тих самих ПБ. При цьому діапазон зміни ПБ можна визначити на основі

діапазону змін характеристик програмно-апаратного забезпечення ресурсів ІС. Наприклад, мінімальне завантаження центрального процесору X=0%, а максимальне X=100%. Визначення реальних X_{min} та X_{max} дозволяє перейти до моделювання з використанням нормалізованих величин:

$$x = (X - X_{min})/(X_{max} - X_{min}), \tag{4.59}$$

де X – реальна величина ПБ;

x – нормалізована величина ПБ.

Таким чином, $x_{min} = 0$, а $x_{max} = 1$, а $x \in [0,1]$.

Також при формуванні ШП на основі деякого параметру X слід показати можливість застосування апарату марківської апроксимації для моделювання процесу зміни випадкової величини X на піддослідному інтервалі [t_1, t_2]. Для цього необхідно показати, що такий процес є випадковим, відповідає умові відсутності післядії, а функція $X = f(t)$ неперервна на інтервалі [t_1, t_2] [35]. Наприклад, для формування ШНП веб-серверу в якості ПБ може використовуватись значення кількості звернень за одну секунду. Відповідно [115, 267, 268], в умовах нормальної експлуатації вказаний параметр неперервний, а його величина є випадковою і залежить тільки від величини в теперішній момент часу, що доводить відсутність післядії. Розвиток даного методу може бути спрямований на застосування в ШП декількох ПБ ІС. Також модель ШП може стати основою управляємого ЛМ, призначеного для оптимізації параметрів засобів захисту ІС.

4.4. Метод визначення ефективності розробки нейромережевих засобів оцінювання параметрів безпеки

Запропонований метод базується на розробленому підході до визначення ефективності розробки НМЗ оцінювання ПБ та розробленій моделі створення ефективних НМЗ. Також в методі використано базові

критерії оцінки ефективності НМЗ, визначені в процесі аналізу відомих НМЗ, що використовуються в СЗІ для виявлення кібератак.

Вхідними даними методу є кортеж виду:

$$\langle \boldsymbol{У}, \boldsymbol{О}, \boldsymbol{М}, D_{min} \rangle, \tag{4.60}$$

де $\boldsymbol{У}$ – множина умов задачі оцінювання ПБ,

$\boldsymbol{О}$ – множина характеристик об'єкту захисту,

$\boldsymbol{М}$ – множина доступних НМЗ,

N_m – кількість доступних НМЗ,

D_{min} – мінімально допустима величина інтегральної ефективності.

Елементи $\boldsymbol{У,О,М}$ визначаються виразами (3.108), (3.107) та (3.106).

Реалізація методу полягає у виконанні п'яти етапів:

Етап 1 – визначення базових критеріїв оцінки ефективності. На цьому етапі для кожного $m_n \in \boldsymbol{М}$ за допомогою експертних даних визначаються величини елементів базової множини критеріїв оцінки ефективності, розроблених в результаті аналізу сучасних НМЗ, що використовуються для виявлення кібератак:

$$\boldsymbol{\Phi} = \{\phi_{ов}, \phi_{ота}, \phi_{бва}, \phi_{ооа}, \phi_{боа}, \phi_{омн}, \phi_{веп}, \phi_{мна}, \phi_{дв}\}, \tag{4.61}$$

Компоненти $\boldsymbol{\Phi}$ призначені для оцінки НМЗ з точки зору $\phi_{ов}$ – можливості попередньої обробки вхідних параметрів, $\phi_{ота}$ – однокритеріальної оптимізації виду архітектури НММ, $\phi_{бва}$ – багатокритеріальної оптимізації виду НММ, $\phi_{ооа}$ – однокритеріальної оптимізації параметрів НММ, $\phi_{боа}$ – багатокритеріальної оптимізації параметрів НММ, $\phi_{омн}$ – оптимізації методу навчання НММ, $\phi_{веп}$ – можливості використання експертних правил для навчання НММ, $\phi_{мна}$ – можливості застосування в НМЗ різноманітних класичних і перспективних

видів НММ, $\phi_{дв}$ – визначення принципової оцінки доцільності застосування НММ. В першому наближенні для визначення величин базових критеріїв оцінки ефективності використовується дискретна трибальна шкала – -1,0,1. Критерій дорівнює -1, коли в НМЗ можливість не забезпечується, 0 – коли забезпечується частково і 1 – коли забезпечується повністю. В подальшому підхід до визначення величин критеріїв може бути вдосконалений.

Для кожного *n*-го НМЗ результатом виконання етапу є множина величин критеріїв $\boldsymbol{\Phi_n}$.

Етап 2 – визначення напрямків вдосконалення НМЗ. На цьому етапі за допомогою формули (4.62) проводиться аналіз величин $\phi_i \in \boldsymbol{\Phi}, i = 1,2..,9$.

$$\phi_i < 1. \tag{4.62}$$

Якщо для *i*-го критерію вираз (4.62) справджується, то це вказує на можливість вдосконалення НМЗ у відповідному напрямку.

Етап 3 – визначення вагових коефіцієнтів важливості базових критеріїв ефективності. Етап спрямований на аналіз $\boldsymbol{У}$ та $\boldsymbol{О}$ для визначення вагових коефіцієнтів важливості елементів множини базових критеріїв ефективності відносно інтегральних критеріїв:

$$\boldsymbol{D} = \left\{d_{ткк}, d_{одв}, d_{анв}, d_{вуз}, d_{ооп}\right\}, \tag{4.63}$$

де $d_{ткк}$ – точність класифікації кібератак,

$d_{одв}$ – можливість оцінки доцільності застосування НМЗ,

$d_{анв}$ – рівень забезпечення адаптації до нових видів кібератак,

$d_{вуз}$ – пристосованість до варіативності умов застосування,

$d_{ооп}$ – пристосованість до функціонування при обмежених обчислювальних ресурсах.

Етап виконується за п’ять кроків, кожен із яких співвідноситься із

визначенням коефіцієнтів важливості стосовно одного із інтегральних критеріїв.

Крок 1 – розрахунок вагових коефіцієнтів для $d_{ткк}$. Розраховуються вагові коефіцієнти базових критеріїв $\alpha_{1,ота},\alpha_{1,бва},\alpha_{1,омн},\alpha_{1,опа},\alpha_{1,бпа},\alpha_{1,мна}$, необхідних для розрахунку залежності (2.11).

Крок 2 – розрахунок вагових коефіцієнтів для $d_{одв}$. Розраховуються вагові коефіцієнти базових критеріїв $\alpha_{2,одв}$, необхідні для розрахунку залежності (2.12).

Крок 3 – розрахунок вагових коефіцієнтів для $d_{анв}$. Розраховуються вагові коефіцієнти базових критеріїв $\alpha_{3,вап},\alpha_{3,мна}$, необхідні для розрахунку залежності (2.13).

Крок 4 – розрахунок вагових коефіцієнтів для $d_{вуз}$. Розраховуються вагові коефіцієнти базових критеріїв $\alpha_{4,ота},\alpha_{4,бва},\alpha_{4,мна}$, необхідні для розрахунку залежності (2.14).

Крок 5 – розрахунок вагових коефіцієнтів для $d_{оап}$. Розраховуються вагові коефіцієнти базових критеріїв $\alpha_{5,по},\alpha_{5,ота},\alpha_{5,бва},\alpha_{5,омн},\alpha_{5,опа},\alpha_{5,бпа},\alpha_{5,мна}$, необхідні для розрахунку залежності (2.15).

Етап 4 – розрахунок інтегральних критеріїв ефективності. На цьому етапі проводиться розрахунок величини кожного із інтегральних критеріїв $d_{ткк}$, $d_{одв}$, $d_{анв}$, $d_{вуз}$ $d_{оап}$. Для розрахунку використовуються наступні вирази:

$$d_{ткк}= 2^{\alpha_{1,ота}\phi_{ота}+\alpha_{1,бва}\phi_{бва}+\alpha_{1,омн}\phi_{омн}+\alpha_{1,опа}\phi_{опа}+\alpha_{1,бпа}\phi_{бпа}+\alpha_{1,мна}\phi_{мна}}, \quad (4.64)$$

$$d_{одв} = 2^{\alpha_{2,одв}\phi_{одв}}, \quad (4.65)$$

$$d_{анв} = 2^{\alpha_{3,вап}\phi_{вап}+\phi_{мна}\alpha_{3,мна}}. \quad (4.66)$$

$$d_{вуз} = 2^{\alpha_{4,ота}\phi_{ота}+\alpha_{4,бва}\phi_{бва}+\alpha_{4,мна}\phi_{мна}}. \quad (4.67)$$

$$d_{ооп} = 2^{\alpha_{5,по}\phi_{по}+\alpha_{5,ота}\phi_{ота}+\alpha_{5,бпа}\phi_{бпа}+\alpha_{5,омн}\phi_{омн}+\alpha_{5,опа}\phi_{опа}+\alpha_{5,бва}\phi_{бва}+\alpha_{5,мна}\phi_{мна}}. \quad (4.68)$$

Етап 5 – розрахунок вагових коефіцієнтів інтегральних критеріїв. Етап спрямований на аналіз ***У*** та ***О*** для визначення вагових коефіцієнтів інтегральних критеріїв ефективності. В результаті аналізу формується множина вагових коефіцієнтів інтегральних критеріїв ефективності виду:

$$\boldsymbol{H} = \{\gamma_1, ...\gamma_5\}, \quad (4.69).$$

де γ_1 – ваговий коефіцієнт і-го інтегрального критерію ефективності.

Етап 6 – розрахунок генерального критерію ефективності. На цьому етапі розраховується генеральний критерій ефективності НМЗ. Для цього використовується вираз:

$$D^{\Sigma} = \sum_{i=1}^{5} \gamma_i d_i, \quad (4.70)$$

де $d_1=d_{ткк}$, $d_2=d_{одв}$, $d_3=d_{анв}$, $d_4=d_{вуз}$, $d_5=d_{ооп}$.

Етап 7 – оцінка ефективності. На даному етапі проводиться остаточне оцінювання ефективності НМЗ. Для цього розрахована величина D^{Σ} порівнюється із мінімально допустимою величиною D_{min}:

$$D^{\Sigma} < D_{min}, \quad (4.71)$$

Якщо нерівність (4.71) справджується, то НМЗ може використовуватись тільки після виправлення недоліків, визначених на другому етапі. Серед декількох доступних НМЗ більш ефективним

вважається той, у якого показник D^{Σ} більший. Таким чином, вперше створено метод, котрий дозволяє розрахувати і порівняти інтегральну ефективність розробки НМЗ оцінки ПБ та визначити напрямки вдосконалення таких засобів. Застосування методу до НМЗ, наведених в [1, 4, 7, 15, 19, 37, 48, 58, 59, 86, 87, 100, 111, 134, 146, 199, 212, 247-250, 252], дозволило визначити, що типовими недоліками більшості з них є низька пристосованість до використання всієї множини перспективних НММ, неможливість подання експертних даних в НММ, недостатнє обґрунтування доцільності використання НММ та вибору оптимального виду НММ.

4.5. Висновки до четвертого розділу

В даному розділі вирішувалась науково-практична задача розробки методів побудови нейромережевих засобів оцінювання параметрів безпеки Інтернет-орієнтованих інформаційних систем. В процесі вирішення отримано наступні результати:

– Вперше розроблено метод подання експертних знань для нейромережевих засобів оцінки параметрів безпеки, в якому за рахунок розробленого математичного забезпечення детермінування параметрів статистично подібних кібератак, продукційних правил представлення навчальних прикладів та структури і вагових коефіцієнтів синаптичних зв’язків нейромережевої моделі типу MPNN забезпечується оперативність розпізнавання та розширення множини видів кібератак, характеристики яких не представлені в статистичних даних. Проведені експериментальні дослідження показали, що застосування запропонованого методу дозволяє в 2 рази підвищити точність розпізнавання кібератак класу U2R.

– Вперше розроблено метод визначення часових характеристик використання нейромережевих засобів, в якому завдяки розробленим аналітичним залежностям для визначення очікуваного терміну розробки, допустимих термінів формування навчальної вибірки та навчання

нейромережевої моделі, запропонованим співвідношенням між очікуваним і допустимим терміном розробки та очікуваним і допустимим терміном навчання, розробленій множині допустимих видів нейромережевих моделей отримана можливість визначення доцільності застосування нейромережевих засобів оцінки параметрів безпеки для виявлення кібератак на заданий об’єкт захисту. Показано, що нейромережеві засоби доцільно використовувати для розпізнавання кібератак типу СП, Neptune, Smurf та ВСВ і недоцільно – для розпізнавання атак типу phf та multihop.

– Вперше розроблено метод проектування шаблону поведінки, який використовується для навчання нейромережевих моделей, в якому за рахунок застосування багатоперіодичних рядів динаміки, розробленого математичного забезпечення для розрахунку періодичних складових та розробленої негомогенної марківської моделі забезпечується в 1,5-2 рази зменшення похибки шаблону, що є основою для зменшення терміну формування навчальної вибірки та зменшення похибок класифікації нейромережевих моделей при розпізнаванні поступових кібератак. Метод використано для розробки марківських моделей шаблонів поведінки веб-серверу.

– Вперше розроблено метод визначення ефективності розробки нейромережевих засобів оцінки параметрів безпеки, який за рахунок застосування запропонованих критеріїв оцінки ефективності, що відображають ступінь виконання основних вимог до побудови та застосування нейромережевих засобів, запропонованих вагових коефіцієнтів важливості критеріїв ефективності та розробленого генерального критерію ефективності нейромережевих засобів дозволяє, відповідно до визначених показників, обрати найбільш ефективний засіб. За допомогою даного методу визначено, що типовими недоліками більшості відомих нейромережевих засобів є низька пристосованість до використання всієї множини нейромережевих моделей, неможливість подання в них експертних даних, недостатнє обґрунтування доцільності використання та вибору оптимального виду нейромережевої моделі.

РОЗДІЛ 5
НЕЙРОМЕРЕЖЕВІ СИСТЕМИ ОЦІНЮВАННЯ ПАРАМЕТРІВ БЕЗПЕКИ ІНТЕРНЕТ-ОРІЄНТОВАНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

5.1. Комплексна методологія нейромережевого оцінювання параметрів безпеки інформаційних систем

В результаті інтеграції запропонованих підходів, моделей та методів з відомими моделями та методами створення НМЗ для розпізнавання кібератак побудовано комплексну методологію оцінювання ПБ (рис. 5.1). Вхідними даними методології є кортеж, що складається із множини умов задачі оцінювання, характеристик об'єкту захисту та доступних видів НМЗ:

$$\langle \boldsymbol{У}, \boldsymbol{O}, \boldsymbol{M} \rangle, \tag{5.1}$$

Множина умов задачі оцінювання ПБ $\boldsymbol{У}$ визначається виразом (3.108), множина характеристик об'єкту захисту $\boldsymbol{O}$ визначається виразом (2.6), а множина доступних видів НМЗ $\boldsymbol{M}$ – виразом (3.106). Також в методології використовуються експертні дані:

– $\langle \boldsymbol{C}_1, \boldsymbol{C}_2, ..., \boldsymbol{C}_K \rangle$ – кортеж вагових коефіцієнтів значимості ПБ та $\boldsymbol{B} = \{\beta_1, ... \beta_K\}$ – множина мінімальних значень коефіцієнтів вагомості ПБ, визначені в моделі інтеграції ПБ;

– R_{max} – максимальна приведена різниця номенклатур ПБ, наведена в пілході до класифікації подібних кібератак;

– $\boldsymbol{E}$ – множина критеріїв оптимізації, $\boldsymbol{V}$ – множина вагових коефіцієнтів критеріїв оптимізації виду НММ та k_E – коефіцієнт відхилення, наведені в підході до визначення оптимального виду НММ;

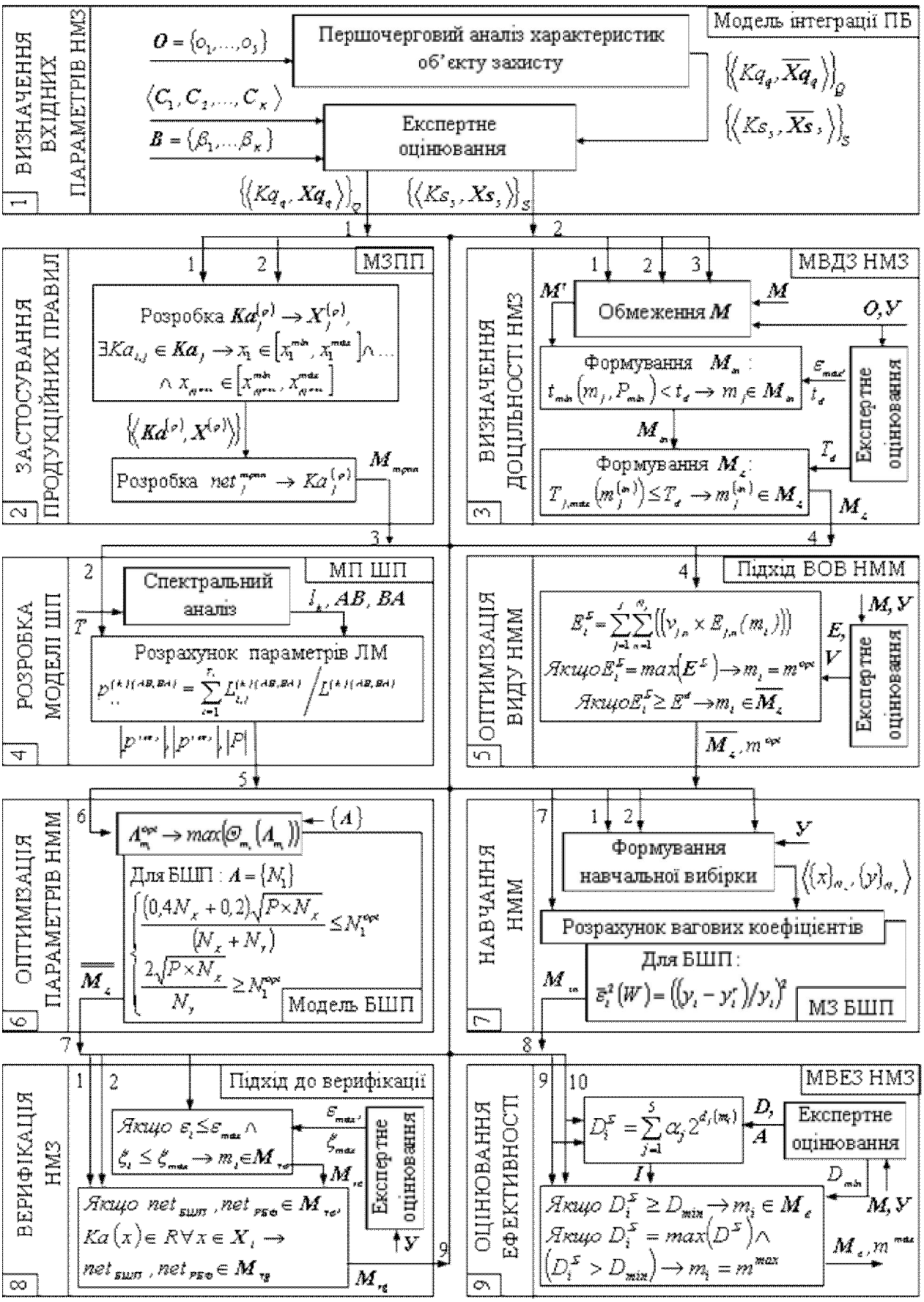


Рис. 5.1. Схема комплексної методології нейромережевої оцінки ПБ

– $\boldsymbol{\Phi}$ – множина значень базових критеріїв оцінки ефективності НМЗ, D_{min} – мінімально допустима ефективність НМЗ, $\boldsymbol{A}$, $\boldsymbol{H}$ – множини значень вагових коефіцієнтів базових та інтегральних критеріїв оцінки ефективності, наведені в методі визначення ефективності розробки НМЗ оцінювання ПБ;

– $\boldsymbol{\Lambda} = \{\lambda_1, \lambda_2, ...\}$ – множина оптимізуємих параметрів НММ, що визначається на основі аналізу конкретного виду НММ;

– ε_{max} – максимально допустима помилка розпізнавання НММ, ξ_{max} – максимально допустима обчислювальна складність НММ та T_a – допустимий термін створення НМЗ, що визначаються на основі аналізу поставленої задачі розпізнавання кібератак.

В базовому випадку обробку експертних даних пропонується проводити методом парних порівнянь за допомогою виразів (3.16-3.30). В подальшому можливо використати інші методи обробки експертних даних, наведені в [90, 110].

Результатом методології є визначення параметрів найбільш ефективних НММ оцінювання ПБ для виявлення кібератак на обраний об’єкт захисту. Її реалізація полягає у виконанні дев’яти етапів:

Етап 1 – визначення вхідних параметрів НМЗ. Виконання етапу полягає у застосуванні розробленої моделі інтеграції ПБ для визначення очікуваних множин ПК $\boldsymbol{Ks}$ і НК $\boldsymbol{Kq}$ та відповідних їм множин ПБ $\boldsymbol{Xq}$ і $\boldsymbol{Xs}$, що будуть використані в якості вхідних параметрів НМЗ. Таким чином, вихідною інформацією етапу є вирази

$$\left\{\left\langle Kq_q, \boldsymbol{Xq}_q \right\rangle\right\}_Q, \tag{5.2}$$

$$\left\{\left\langle Ks_s, \boldsymbol{Xs}_s \right\rangle\right\}_S, \tag{5.3}$$

де $\boldsymbol{Xq}_q$, $\boldsymbol{Xs}_s$ – множини ПБ для розпізнавання q-ої НК та s-ої ПК.

.Вхідною інформацією етапу є множини $\boldsymbol{O}$, $\boldsymbol{B}$ та кортеж $\langle \boldsymbol{C}_1, \boldsymbol{C}_2, ..., \boldsymbol{C}_K \rangle$. Етап виконується за два кроки.

Крок 1 – <u>першочерговий аналіз характеристик об'єкту захисту</u>. Першочерговий аналіз множини $\boldsymbol{O}$ полягає у побудові 1-4 процесів моделі інтеграції ПБ. Використовуються вирази (3.31-3.34). Результатом аналізу є

$$\left\{ \left\langle Kq_q, \overline{\boldsymbol{Xq}}_q \right\rangle \right\}_Q, \tag{5.4}$$

$$\left\{ \left\langle Ks_s, \overline{\boldsymbol{Xs}}_s \right\rangle \right\}_S, \tag{5.5}$$

де $\overline{\boldsymbol{Xq}}_q$, $\overline{\boldsymbol{Xs}}_s$ – в першому наближенні множини ПБ, що застосовуються для розпізнавання q-ої НК та s-ої ПК.

Крок 2 – <u>експертне оцінювання</u>. Крок орієнтовано на встановлення відповідності між k-им видом кібератаки та множиною ПБ, що використовується для її розпізнавання:

$$Ka_k \to \boldsymbol{X}_k. \tag{5.6}$$

Для цього за допомогою експертного оцінювання (5.4, 5.5.) визначається кортеж $\langle \boldsymbol{C}_1, \boldsymbol{C}_2, ..., \boldsymbol{C}_K \rangle$ та множина $\boldsymbol{B}$. Для встановлення відповідності (5.6) використовується математичний апарат п'ятого процесу моделі інтеграції ПБ.

Етап 2 – <u>застосування продукційних правил</u>. Призначенням даного етапу є навчання НММ оцінки ПБ за рахунок подання в них експертних знань про відповідність величин ПБ з наявністю/відсутністю очікуваних кібератак. Вхідними даними етапу являються вирази (5.2,5.3). В базовому варіанті подання експертних знань реалізується за допомогою розробленого методу застосування продукційних правил для навчання НМ типу MPNN. В

подальшому можуть використовуватись і інші методи подання експертних знань. Етап виконується за 2 кроки. Крок 1 відповідає 3,4 етапу, а крок 2 відповідає 5-10 етапам методу застосування продукційних правил.

Крок 1 – визначення ПБ для розпізнавання подібних кібератак. Призначанням даного кроку є визначається множини ПБ, котрі використовуються для розпізнавання множини подібних кібератак:

$$\left\{\left\langle \boldsymbol{Ka}^{(p)}, \boldsymbol{X}^{p} \right\rangle\right\}. \tag{5.7}$$

Для визначення (5.7) використовується математичний апарат (4.3-4.11).

Крок 2 – розробка моделей MPNN. Крок орієнтовано на розробку множини MPNN, кожна з яких призначена для розпізнавання окремого виду подібних кібератак. В процесі розробки використовується математичний апарат (4.12-4.22).

Вихідною інформацією етапу є:

$$\boldsymbol{M}_{mpnn} = \left\{net_1^{mpnn},...,net_{M_{mpnn}}^{mpnn}\right\}, \tag{5.8}$$

де net_j^{mpnn} – j-та MPNN, призначена для розпізнавання $\boldsymbol{Ka}_j^{(p)} \in \overline{\boldsymbol{Ka}}^{(p)}$.

Етап 3 – визначення доцільності застосування НМЗ. На даному етапі визначається множина НМЗ $\boldsymbol{M}_z$, які доцільно застосувати для оцінки інтегрованих ПБ. Вхідними даними етапу є множини $\boldsymbol{O}$, $\boldsymbol{У}$, $\boldsymbol{M}$, $\boldsymbol{M}_{mpnn}$, $\langle \boldsymbol{Kq}, \boldsymbol{Xq} \rangle$ та $\langle \boldsymbol{Ks}, \boldsymbol{Xs} \rangle$. Етап реалізується за рахунок виконання 2-12 етапів методу визначення доцільності застосування НМЗ. Відповідний математичний апарат задається виразами (4.26-38). Для формування $\boldsymbol{M}_z$ використовуються правила (4.39-40).

Етап 4 – розробка моделі шаблону поведінки. Етап орієнтовано на

розробку марківської моделі ШП, що використовується для розпізнавання очікуваних ПК. Вхідними даними етапу є множина параметрів ПБ $\boldsymbol{Xs}$, що залежать від терміну експлуатації об'єкту захисту і використовуються для розпізнавання ПК. Розробка марківської моделі реалізується за допомогою створеного методу проектування ШП ПБ. Застосовується математичний апарат (4.44-4.53). Виходом етапу являються множини перехідних матриць $\boldsymbol{p}^{(AB)}$, $\boldsymbol{p}^{(BA)}$ та матриця ймовірностей станів $|P(t)|$ марківської моделі ШП ПБ.

Етап 5 – оптимізація виду НММ. На даному етапі визначається множина оптимальних видів НММ виду

$$\overline{\boldsymbol{M}_z} = \left\{m_1^{opt},...,m_I^{opt}\right\}. \tag{5.9}$$

Вхідними даними являються множини $\boldsymbol{M_z}$, $\boldsymbol{E}$, $\boldsymbol{V}$. Етап базується на розробленому підході до оптимізації та виконується за три кроки:

Крок 1 – розрахунок інтегрального критерію оптимізації НММ. На даному кроці для елементів $\boldsymbol{M_z}$ розраховується множина інтегральних критеріїв оптимізації:

$$\boldsymbol{E}^{\boldsymbol{\Sigma}} = \left\{E_1^{\Sigma},...,E_{N_m}^{\Sigma}\right\}, \tag{5.10}$$

$$E_i^{\Sigma} = \sum_{j=1}^{J}\sum_{n=1}^{N}\left(\left(v_{j,n} \times E_{j,n}(m_i)\right)\right),\ i = 1,2,...,I, \tag{5.11}$$

де $E_{j,n}$ – оцінка n-ого критерію в j-ій категорії для i-ої НММ,

$v_{j,n} \in \boldsymbol{V}$ – ваговий коефіцієнт n j-ого критерію оптимізації,

I – кількість допустимих НММ,

J – кількість категорій критеріїв,

N_J – кількість критеріїв в J-ій категорії.

Крок 2 – визначення оптимального виду моделі. На даному етапі

визначається оптимальний вид НММ. Для цього використовується правило:

$$Якщо\ E_i^{\Sigma} = max\left(\boldsymbol{E}^{\boldsymbol{\Sigma}}\right) \rightarrow m_i = m^{opt}. \tag{5.12}$$

Крок 3 – формування множини оптимальних видів НММ. Для формування множини $\overline{\boldsymbol{M}_z}$ використовуються вирази:

$$Якщо\ E_i^{\Sigma} \geq E^d \rightarrow m_i \in \boldsymbol{M}^{\boldsymbol{opt}}, \tag{5.13}$$

$$E^d = k_E E^{\Sigma}\left(m^{opt}\right), \tag{5.14}$$

де $k_E = 0{,}8$ (в першому наближенні).

Етап 6 – оптимізація параметрів НММ. Етап орієнтовано на визначення $\overline{\overline{\boldsymbol{M}_z}}$ – множини оптимальних видів НММ з оптимізованими параметрами. Вхідною інформацією етапу є множина $\overline{\boldsymbol{M}_z}$. Використано критерій оптимізації виду:

$$\Theta(\Lambda) \rightarrow max, \tag{5.15}$$

де Θ – обчислювальна потужність моделі.

Розрахунок оптимальних величин $\{\lambda_1, \lambda_2, ...\}$ проводиться методами, специфічними для виду НММ. У випадку використання БШП, який в більшості випадків входить до складу $\overline{\boldsymbol{M}_z}$, етап виконується з використанням розробленої структурної моделі. Діапазон оптимальних параметрів визначаються виразами (3.77,3.78).

Етап 7 – навчання НММ. Етап орієнтовано на розрахунок $\boldsymbol{M}_{zn}$ множини вагових коефіцієнтів синаптичних зв'язків НММ, що входять до множини $\overline{\overline{\boldsymbol{M}_z}}$.

Етап виконується за два кроки:

Крок 1 – формування навчальної вибірки. На даному кроці формується навчальна вибірка НМ. Для НМ, що навчаються «з вчителем», навчальна вибірка представляє собою кортежі виду $\left\langle \{x\}_{N_x}, \{y\}_{N_y} \right\rangle$, а для НМ, що самонавчаються, множини $\{x\}_{N_x}$. У випадку розпізнавання ***Kq*** множина вхідних параметрів $\{x\}_{N_x}$ формується на основі ***Xq***, а у випадку розпізнавання ***Ks*** – на основі множини ***Xs*** та множини відхилень ***Xs*** від ШП $\boldsymbol{DXs}(t)$. При цьому відхилення *i*-го ПБ в момент часу *t* від ШП розраховується так:

$$DXs_i(t) = MXs_i(t) - Xs_i(t), \tag{5.16}$$

де $Xs_i(t)$ – величина *i*-го ПБ в момент часу *t*,

$MXs_i(t)$ – математичне сподівання *i*-го ПБ, розраховане за допомогою марківської моделі ШП.

Обсяг навчальної вибірки P_n розраховується за допомогою (4.32). Із навчальної вибірки виділяється тестова вибірка, обсяг якої складає $P_t = 0{,}05P_n$.

Крок 2 – реалізація процесу навчання. На даному кроці в процесі подання навчальних прикладів розраховуються вагові коефіцієнти синаптичних зв'язків. Розрахунок реалізується за допомогою методів, характерних для виду НММ. Для БШП для корекції вагових коефіцієнтів використовується розроблені вирази (2.48, 2.49, 2.63, 2.64).

Етап 8 – верифікація НМЗ. Етап орієнтовано на верифікацію НМЗ з позицій достатньої обчислювальної потужності та можливості нейромережевої апроксимації піддослідної функції зміни ПБ.

Для кожної очікуваної кібератаки $Ka_i \in \boldsymbol{Ka}$ даний етап реалізується за два кроки:

Крок 1 – визначення достатньої обчислювальної потужності. Виконання даного кроку полягає у реалізації правила:

$$Якщо\ \exists m_i \in \boldsymbol{M}_z\ \varepsilon_i \le \varepsilon_{max} \wedge \xi_i \le \xi_{max} \to m_i \in \boldsymbol{M}_{ve}, \tag{5.17}$$

де ε_i – помилка узагальнення для m_i,

ξ_i – обчислювальна складність навчання m_i,

$\boldsymbol{M}_{ve}$ – множина НМЗ верифікованих на експериментальних даних.

В першому наближенні ε_i дорівнює помилці навчання m_i на тестовій вибірці, а ξ_i дорівнює кількості навчальних ітерацій m_i.

Крок 2 – доведення гладкості функції. Крок виконується за умови

$$net_{БШП} \wedge / \vee net_{РБФ} \in \boldsymbol{M}_{ve}, \tag{5.18}$$

де $net_{БШП}, net_{РБФ}$ – НМЗ на основі БШП та РБФ.

Виконання кроку базується на розробленому підході і полягає у доведенні того, що

$$Ka_i = f_i(\boldsymbol{X}_i) \to гладка\ функція, \tag{5.19}$$

де $\boldsymbol{X}_i$ – множина ПБ, що використовуються для розпізнавання Ka_i. Доведення (5.19) реалізується за допомогою експертного оцінювання множин $\boldsymbol{У, О, X}_i$. Якщо гладкість функції доведена, то $\boldsymbol{M}_{vg}$ – множина гарантовано верифікованих НМЗ складається із БШП та РБФ. В протилежному випаку $\boldsymbol{M}_{vg} = \varnothing$.

Вихідною інформацією етапу є $\boldsymbol{M}_{ve}$ – множина НМЗ верифікованих експериментально та $\boldsymbol{M}_{vg}$ – множина гарантовано верифікованих НМЗ.

Етап 9 – оцінка ефективності НМЗ. Призначенням етапу є розробка

множини ефективних НМЗ та визначення шляхів їх можливого вдосконалення. Для цього використовується розроблений метод оцінювання ефективності. Вхідними даними етапу є $\langle \boldsymbol{У, О, M_{ve}, M_{vg}}, D_{min} \rangle$. Етап виконується за 2 кроки. Крок 1 відповідає 1,3-6 етапам, а крок 2 – 7 етапу методу оцінювання ефективності.

Крок 1 – розрахунок показників ефективності. На даному кроці визначаються величини елементів $\boldsymbol{Ф, D, A}$, за допомогою яких для кожного $m_i \in \boldsymbol{M_{ve}}$ розраховується інтегральний показник ефективності D_i^{Σ}. Для визначення елементів $\boldsymbol{Ф, D, A}$ використовуються вирази (4.55,4.57-4.62). Для розрахунку D_i^{Σ} використовується вираз (4.63).

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

$$Якщо\ D_i^{\Sigma} > D_{min} \rightarrow m_i \in \boldsymbol{M_e}. \qquad (5.20)$$

Правило для визначення найбільш ефективного НМЗ виглядає так:

$$Якщо\ D_i^{\Sigma} = max(D) \wedge \left(D_i^{\Sigma} > D_{min}\right) \rightarrow m_i = m^{max}. \qquad (5.21)$$

Виходом даного етапу є $\boldsymbol{M_e}$ – множина ефективних НМЗ та m^{max} – найбільш ефективний НМЗ.

Проведено порівняння ефективності відомих нейромережевих методів розпізнаваня кібератак та запропонованої комплексної методології нейромережевого оцінювання ПБ ІС – КМНО. Для цього застосовано розроблений метод оцінки ефективності. Зазначимо, що величини базових критеріїв оцінки ефективності мають наступні значення: $ф_{по}$=0, $ф_{ота}$=1, $ф_{бва}$=1, $ф_{опа}$=1, $ф_{бпа}$=0, $ф_{омн}$=0, $ф_{веп}$=1, $ф_{мна}$=1, $ф_{дв}$=1.

Визначені в першому наближенні вагові коефіцієнти базових критеріїв оптимізації такі: $\alpha_{1,ота}=0{,}5$, $\alpha_{1,бва}=1$, $\alpha_{1,опа}=0{,}5$, $\alpha_{1,бпа}=1$, $\alpha_{1,омн}=1$, $\alpha_{1,мна}=0{,}5$, $\alpha_{2,одв}=1$, $\alpha_{3,веп}=1$, $\alpha_{3,мна}=0{,}5$, $\alpha_{4,ота}=0{,}5$, $\alpha_{4,бва}=1$, $\alpha_{4,мна}=1$, $\alpha_{5,по}=0{,}5$, $\alpha_{5,ота}=0{,}5$, $\alpha_{5,бва}=1$, $\alpha_{5,опа}=0{,}5$, $\alpha_{5,бпа}=1$, $\alpha_{5,омн}=0{,}5$, $\alpha_{5,мна}=0{,}5$.

В першому наближенні прийнято, що всі вагові коефіцієнти інтегральних критеріїв дорівнюють $\gamma=1$, а для розрахунку генерального критерію ефективності застосовано вираз (4.63).

Визначені критерії критерії ефективності відомих нейромережевих методів та розробленої комплексної методології наведені в табл. 5.1.

Таблиця 5.1

Оцінка ефективності нейромережевих моделей та методів

№	Модель, метод	Критерій					
		$d_{ткк}$	$d_{одв}$	$d_{анв}$	$d_{вуз}$	$d_{ооп}$	D^{Σ}
1	2	3	4	5	6	7	8
1	ВФПК	0,176777	0,5	1,414214	0,25	0,125	2,46599
2	МКН	0,088388	0,5	0,353553	0,353553	0,088388	1,383883
3	МТК	0,125	0,5	0,353553	0,353553	0,125	1,457107
4	НШІС	0,176777	0,5	0,353553	0,353553	0,088388	1,472272
5	НПВІ	0,5	0,5	0,353553	0,707107	0,353553	2,414214
6	ВНДБД	0,088388	0,5	0,353553	0,25	0,088388	1,28033
7	НФС	2,828427	0,5	0,353553	0,707107	1,414214	5,803301
8	АППТ	0,044194	0,5	0,5	0,353553	0,0625	1,460248
9	ПСК	0,176777	0,5	0,353553	0,25	0,125	1,40533
10	НСВВ	0,176777	0,5	0,353553	0,25	0,088388	1,368718
11	ТВМА	0,088388	0,5	0,353553	0,25	0,088388	1,28033

Таблиця 5.1(продовження)

1	2	3	4	5	6	7	8
12	РАМТ	0,125	0,5	0,707107	0,353553	0,0625	1,74816
13	ВМА	0,088388	0,5	1,414214	0,25	0,0625	2,315102
14	ВМА	0,176777	0,5	0,353553	0,25	0,125	1,40533
15	ВМА	0,25	0,5	0,353553	0,707107	0,25	2,06066
16	ПСКТ	0,176777	0,5	0,353553	0,25	0,125	1,40533
17	ПВМА	0,25	0,5	0,353553	0,353553	0,176777	1,633883
18	АСВА	2,828427	0,5	0,353553	0,707107	2	6,389087
19	ВАОП	0,044194	0,5	0,353553	0,176777	0,044194	1,118718
20	СВДА	0,176777	0,5	0,353553	0,25	0,088388	1,368718
21	БНМ	0,25	0,5	0,353553	0,25	0,0625	1,416053
22	ВКМА	0,044194	0,5	0,353553	0,176777	0,044194	1,118718
23	**КМНОПБ**	**2,828427**	**2**	**2,828427**	**5,656854**	**8**	**24,62742**

Аналіз даних табл. 5.1 вказує на те, що використання запропонованої комплексної методології оцінювання ПБ дозволяє підвищити генеральний показник ефективності розробки в 3,85 рази по відношенню до найкращих нейромережевих методів.

5.2. Система оцінювання параметрів безпеки для розпізнавання кібератак на ресурси Інтернет-орієнтованих інформаційних систем

За допомогою запропонованої комплексної методології розроблено структуру нейромережевої системи оцінки параметрів безпеки для розпізнавання кібератак на ресурси Інтернет-орієнтованих ІС (див. рис. 5.1). До складу даної нейромережевої системи входять:

– ППВПК – підсистема первинного визначення параметрів кібератак, призначена для формування множини очікуваних видів кібератак,

попереднього визначення переліку ПБ та навчальної вибірки НММ;

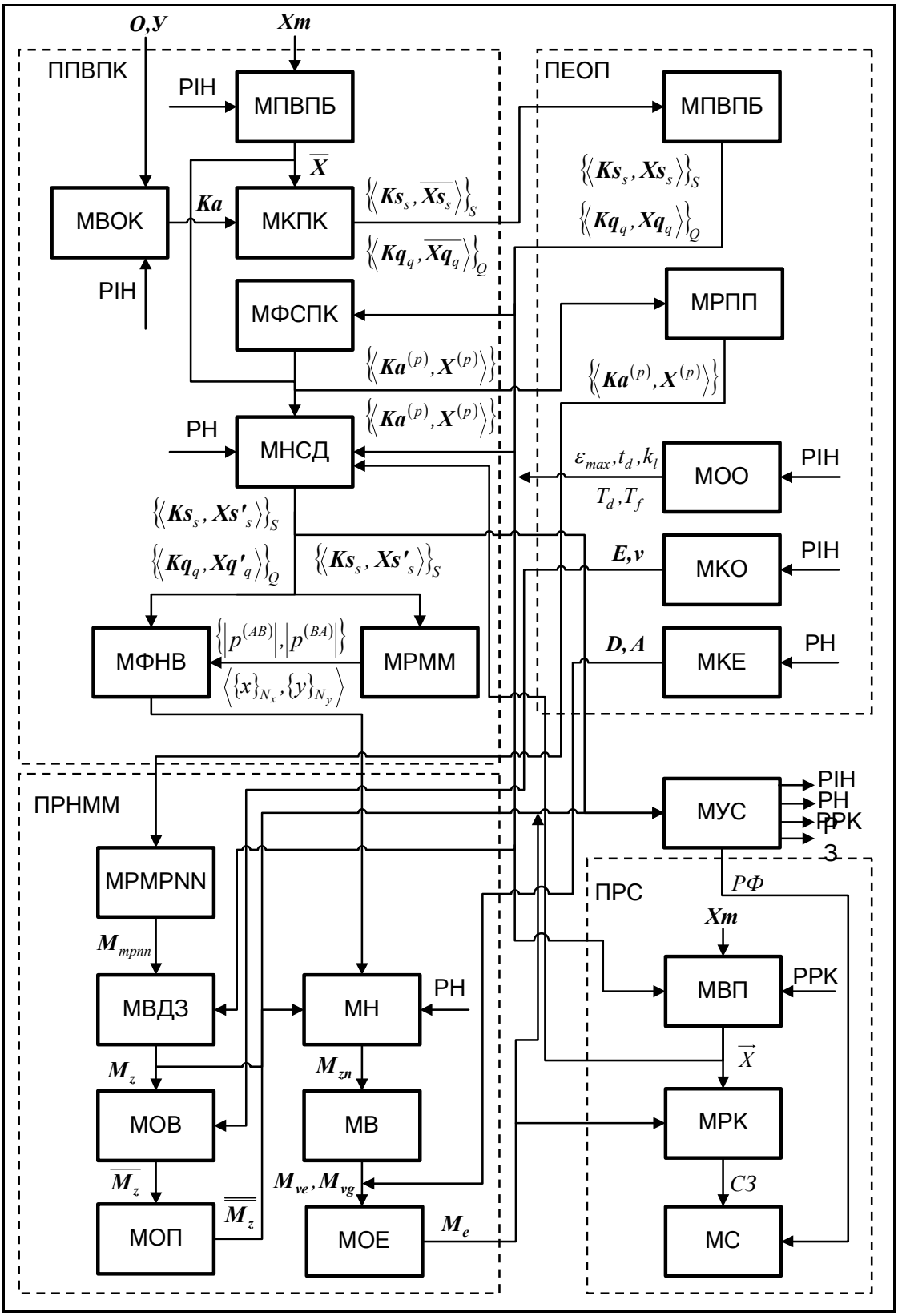


Рис. 5.2. Структура нейромережевої системи оцінювання параметрів безпеки

– ПЕОП – підсистема експертної оцінки параметрів НМЗ, в якій на основі експертних даних для кожного виду кібератак формуються відповідні множини ПБ, розроблюються продукційні правила для розпізнавання кібератак, визначаються обмеження та критерії ефективності НММ;

– ПРНММ – підсистема розробки нейромережевих моделей, що призначена для визначення параметрів НММ, їх верифікації та оцінки ефективності;

– ПРС – підсистема розпізнавання та сигналізації, в якій реалізується застосування розроблених НММ для розпізнавання кібератак та виробляється інформація для адміністратора системи;

– МУС – модуль управління системою, що служить для переведення системи в наступні режими функціонування: РІН –ініціалізації налаштувань; РН – навчання НММ; РРК – розпізнавання кібератак; РЗ – зупинки.

Призначення окремих модулів розробленої НМС, що входять до складу означених підсистем, наведені в табл.5.2.

Таблиця 5.2

Склад нейромережевої системи оцінювання параметрів безпеки

Назва підсистеми	Назва модулю	Призначення модулю
1	2	3
ППВПК	МВОК	Визначення множини очікуваних кібератак
	МПВПБ	Попереднього визначення ПБ
	МКПК	Класифікації параметрів кібератак
	МФСПК	Формування множин подібних кібератак
	МНСД	Накопичення статистичних даних
	МРММ	Розробки марківських моделей ШП
	МФНВ	Формування навчальної вибірки НММ
ПЕОП	МІПБ	Інтеграції ПБ

Таблиця 5.2 (продовження)

1	2	3
ПЕОП	МОО	Визначення обчислювальних обмежень
	МКО	Визначення значень критеріїв оптимізації виду та параметрів НММ
	МКЕ	Визначення значень критеріїв ефективності
	МРПП	Розробки продукційних правил
ПРНММ	МРМРNN	Розробки MPNN
	МВДЗ	Визначення доцільності застосування НМЗ
	МОВ	Оптимізації виду НММ
	МОП	Оптимізації параметрів НММ
	МН	Навчання НММ
	МВ	Верифікації НММ
	МОЕ	Оцінювання ефективності НМЗ
ПРС	МС	Сигналізації про стан функціонування та стан захищеності
	МРК	Розпізнавання кібератак
	МВП	Визначення вхідних параметрів НМЗ

Розроблена нейромережева система починає функціонувати в режимі ініціалізації налаштувань. Для цього відповідно моделі інтеграції ПБ, на основі характеристик обраного об’єкту (***О***) захисту та умов задачі захисту (***У***) в модулі визначення множини очікуваних кібератак формується множина очікуваних кібератак $\boldsymbol{Ka}$, а в модулі попереднього визначення ПБ на основі хостових та мережевих параметрів $\boldsymbol{Xm}$ визначається множина ПБ $\overline{\boldsymbol{X}}$, що може використовуватись для розпізнавання $\boldsymbol{Ka}$.

Множини $\boldsymbol{Ka}$ та $\overline{\boldsymbol{X}}$ поступають в модуль класифікації параметрів кібератак, в якому кібератаки класифікуються на поступові $\boldsymbol{Ks}$ і неочікувані $\boldsymbol{Kq}$ та визначаються множини ПБ $\overline{\boldsymbol{Xs}}$ та $\overline{\boldsymbol{Xq}}$, що можуть бути використані для розпізнавання кожної із очікуваних кібератак.

Таким чином, виходом даного модулю є множини $\left\{\left\langle \boldsymbol{Ks}_s, \overline{\boldsymbol{Xs}_s}\right\rangle\right\}_S$ та $\left\{\left\langle \boldsymbol{Kq}_q, \overline{\boldsymbol{Xq}_q}\right\rangle\right\}_Q$, які подаються в модуль інтеграції ПБ. В даному модулі за допомогою експертного оцінювання для кожної із очікуваних кібератак остаточно визначається множина ПБ, яка буде використовуватись для їх розпізнавання. Таким чином, виходом модулю є портрети ПК та НК: $\left\{\left\langle \boldsymbol{Ks}_s, \boldsymbol{Xs}_s\right\rangle\right\}_S$, $\left\{\left\langle \boldsymbol{Kq}_q, \boldsymbol{Xq}_q\right\rangle\right\}_Q$.

Паралельно з модулем інтеграції ПБ спрацьовують модулі визначення обчислювальних обмежень, значень критеріїв оптимізації НММ та значень критеріїв ефективності. Результатом їх спрацювання є: допустима помилка розпізнавання (ε_{max}), допустима кількість навчальних обчислювальних ітерацій НММ (ξ_{max}), допустимий термін навчання (t_d), коефіцієнт обсягу статистичних даних (k_l), допустимий термін розробки НММ (T_f), термін формування навчальної вибірки (T_d), критерії оптимізації виду НММ ($\boldsymbol{E}$), коефіцієнти значимості критеріїв оптимізації ($\boldsymbol{v}$), критерії ефективності НМЗ ($\boldsymbol{Ф, D}$) та коефіцієнти значимості критеріїв ефективності ($\boldsymbol{A, H}$).

Отримані в модулі інтеграції ПБ портрети кібератак поступають в модуль формування множин подібних кібератак, в якому формуються множини кортежів подібних кібератак та відповідних їм ПБ – $\left\{\left\langle \boldsymbol{Ka}^{(p)}, \boldsymbol{X}^{(p)}\right\rangle\right\}$. Вказані множини подаються в модуль розробки продукційних правил, де на основі експертних даних для кожної множини подібних кібератак створюються множини продукційних правил – $\left\{\left\langle \boldsymbol{Ka}^{(p)}, \boldsymbol{R}\left(\boldsymbol{X}^{(p)}\right)\right\rangle\right\}$.

Множина $\left\{\left\langle \boldsymbol{Ka}^{(p)}, \boldsymbol{R}\left(\boldsymbol{X}^{(p)}\right)\right\rangle\right\}$ передається в модуль розробки MPNN. В цьому модулі на основі запропонованого методу застосування продукційних правил для кожної множини статистично подібних кібератак розроблюється відповідна MPNN. Виходом модулю є $\boldsymbol{M}_{mpnn}$ – множина розроблених MPNN.

Отримана множина $\boldsymbol{M}_{mpnn}$ разом з $\left\{\left\langle \boldsymbol{Ks}_s, \boldsymbol{Xs}_s\right\rangle\right\}_S$, $\left\{\left\langle \boldsymbol{Kq}_q, \boldsymbol{Xq}_q\right\rangle\right\}_Q$, ε_{max}, t_d,

T_d та T_f подається в модуль визначення доцільності застосування НМЗ. В цьому модулі за допомогою розробленого одноіменного методу формується $\boldsymbol{M}_z$ – множина НМЗ, які доцільно використовувати для розпізнавання очікуваних кібератак.

Якщо НМЗ використовувати недоцільно, то ця інформація передається в модуль управління системою, який за допомогою модулю сигналізації надає адміністратору системи відповідний сигнал та зупиняє функціонування системи. Якщо НМЗ використовувати доцільно, то визначена $\boldsymbol{M}_z$ передається в модуль оптимізації виду НММ, в якому за допомогою розробленого підходу до визначення оптимального виду НММ, на основі $\boldsymbol{M}_z$, $\boldsymbol{E}$ та $\boldsymbol{v}$ відбувається визначення $\overline{\boldsymbol{M}_z}$ – множини оптимальних видів НММ. В свою чергу $\overline{\boldsymbol{M}_z}$ передається в модуль оптимізації параметрів НММ, видом якого є $\overline{\overline{\boldsymbol{M}_z}}$ – множина оптимальних видів НММ з оптимізованими параметрами. Якщо до складу $\overline{\boldsymbol{M}_z}$ входить БШП, то оптимізація його параметрів відбувається за допомогою розробленої моделі.

Після спрацювання модулю оптимізації параметрів НММ система переходить в режим навчання. Спрацьовує модуль накопичення статистичних даних. Вихідними даними цього модулю є множини ПК і НК та множини відповідних їм накопичених даних щодо ПБ: $\left\{\left\langle \boldsymbol{Ks}_s, \boldsymbol{Xs'}_s \right\rangle\right\}_S$ та $\left\{\left\langle \boldsymbol{Kq}_q, \boldsymbol{Xq'}_q \right\rangle\right\}_Q$. Множина $\left\{\left\langle \boldsymbol{Ks}_s, \boldsymbol{Xs'}_s \right\rangle\right\}_S$ передається в модуль розробки марківських моделей ШП. В цьому модулі за допомогою розробленого методу проектування ШП реалізується визначення множини перехідних ймовірностей, визначаються $\hat{X}(t), t \in T$ – очікувані значення ПБ на протязі заданого терміну функціонування T.

Отримані множини $\left\{\left\langle \boldsymbol{Ks}_s, \boldsymbol{Xs'}_s \right\rangle\right\}_S$, $\left\{\left\langle \boldsymbol{Kq}_q, \boldsymbol{Xq'}_q \right\rangle\right\}_Q$ та $\hat{X}(t)$ подаються в модуль формування навчальної вибірки НММ, в якому формується

$\left\langle \{x\}_{N_x}, \{y\}_{N_y} \right\rangle$ – кортеж множин вхідних (x) та вихідних (y) параметрів, що використовуються в модулі навчання для визначення вагових коефіцієнтів синаптичних зв'язків НММ, що входять до складу множини $\overline{\overline{\boldsymbol{M}_z}}$. Виходом модулю навчання є $\boldsymbol{M}_{zn}$ – множина НММ, які пройшли навчання.

НММ, що входять до складу $\boldsymbol{M}_{zn}$, подаються в модуль верифікації, в якому реалізується 8 етап запропонованої комплексної методології. Виходом модулю верифікації є $\boldsymbol{M}_{ve}$ – множина НМЗ верифікованих експериментально та $\boldsymbol{M}_{vg}$ – множина гарантовано верифікованих НМЗ. При цьому $\boldsymbol{M}_{vg} \subset \boldsymbol{M}_{ve}$.

Множини $\boldsymbol{M}_{ve}$ та $\boldsymbol{M}_{vg}$ передаються в модуль оцінювання ефективності НМЗ, який функціонує на основі розробленого методу визначення ефективності застосування НМЗ. Результатом його спрацювання є $\boldsymbol{M}_e$ – множина параметрів ефективних НММ та $\boldsymbol{m}^{max}$ – множина параметрів найбільш ефективної НММ. Множини $\boldsymbol{M}_e$ та $\boldsymbol{m}^{max}$, які передаються в модуль розпізнавання кібератак та в модуль управління, котрий переводить систему в режим розпізнавання кібератак.

В режимі розпізнавання кібератак контрольовані поточні хостові та мережеві параметри ІС подаються на вхід модулю визначення вхідних параметрів НМЗ. Виходом модулю є $\overrightarrow{X}$ – вектор вхідних параметрів НММ. Вказаний вектор подається в модуль розпізнавання кібератак, в якому за допомогою НММ, що входять до множини $\boldsymbol{M}_e$, формується сигнал *СЗ* про стан захисту. Сигнал *СЗ* за допомогою модулю сигналізації передається адміністратору системи.

Крім того, вектор $\overrightarrow{X}$ поступає в модуль накопичення статистичних даних. Якщо поточний обсяг накопичених статистичних даних в k_l разів перевищує обсяг даних останнього навчання, то система може бути переведена в режим навчання для уточнення параметрів НММ.

5.3. Система розпізнавання шкідливого програмного забезпечення та класифікації листів електронної пошти

Проектування системи розпізнавання Веб-орієнтованого ШПЗ та класифікації листів електронної пошти проведено за допомогою запропонованої комплексної методології оцінювання ПБ ІС. Також при проектуванні застосовано структурні рішення створеної системи оцінювання ПБ для розпізнавання кібератак на ресурси Інтернет-орієнтованих ІС.

Відповідно комплексній методології на першому етапі проектування системи проведено визначення множини ПБ, оцінювання яких дозволить розпізнати очікувану множину кібератак $\boldsymbol{Ka} = \{\boldsymbol{Ka}_{spz}, \boldsymbol{Ka}_{sp}, \boldsymbol{Ka}_{in}\}$, де $\boldsymbol{Ka}_{spz}$ – скриптове Веб-орієнтоване ШПЗ, $\boldsymbol{Ka}_{sp}$ – спам, $\boldsymbol{Ka}_{in}$ – витоки текстової інформації. Зазначимо, що механізм реалізації очікуваних кібератак не залежить від терміну функціонування ІС. Тому ці кібератаки відносяться до НК, що не дозволяє створювати для них марківські моделі ШП.

ПБ для розпізнавання НК виду $\boldsymbol{Ka}_{spz}$. Формування множини ПБ для розпізнавання кібератак вказаних типів базується на висновках [164] про те, що ПБ повинні відображати здатність скриптових вірусів до саморозповсюдження та характерні спільні ознаки скриптового ШПЗ. При цьому ПБ повинні підлягати моніторингу за допомогою розповсюджених засобів – антивірусних сканерів та поведінкових аналізаторів. Визначені в [184] середовище, засоби та ознаки розповсюдження скриптових вірусів, пристосованих до операційної системи Windows, показані на рис. 5.3. Крім того, визначено, що характерні спільні ознаки скриптового ШПЗ можливо розділити на групи: автоматизації запуску, ігнорування помилок, маскування та деструктивних функцій. Приблизний перелік ознак представлено в табл. 5.3. Зазначимо, що деякі ознаки, вказані в табл. 5.3 та на рис. 5.3, можуть мати спільні прояви. Наприклад, визначення скриптом параметрів програмно-апаратного забезпечення атакуємої ІС може вказувати як на прояв

деструктивних функцій, так і на прояв саморозповсюдження.

Рис. 5.3. Ознаки розповсюдження скриптових вірусів

Таблиця 5.3

Спільні ознаки веб-орієнтованого скриптового ШПЗ

Група ознак	Перелік проявів
Автоматизації запуску	Визначення подій, запуск на певну подію
Ігнорування помилок	Використання операторів перехвату та обробки виключних подій після виникнення помилок
Маскування	Обфусикація скрипта, поліморфізм скрипта, зміна настройок системи безпеки браузеру
Деструктивні функції	Визначення параметрів програмно-апаратного забезпечення, форматування жорстких дисків, модифікація та знищення файлів, встановлення паролів на файли, використання мережевих з'єднань, програмного та апаратного забезпечення

У випадку моніторингу програмного коду кількість ПБ дорівнює кількості потенційно небезпечних операторів мови програмування, на якій написане ШПЗ. У випадку моніторингу подій в операційній системі кількість ПБ дорівнює кількості потенційно небезпечних функцій. При цьому ПБ мають дискретний характер і можуть приймати мати лише два значення: 1 – якщо ознака присутня та -1 в протилежному випадку.

Відзначимо, що відповідно результатам п.1.2, Веб-орієнтоване скриптове ШПЗ в основному написане на мовах програмування JavaScript та VB. ПБ, призначені для розпізнавання Веб-орієнтованого скриптового ШПЗ, написаного на JavaScript, були отримані шляхом аналізу документованих конструкцій цієї мови програмування. Виходячи із особливостей JavaScript аналізувались тільки ті оператори, які дозволяють визначати читати, записувати, передавати інформацію, запускати зовнішні програми, маскувати небезпечні дії, визначати можливі вразливості атакованої ІС та ініціювати деструктивні функції шляхом перехоплення дій користувача. Також в процесі аналізу використовувались відомі сигнатури ШПЗ, написаного на JavaScript. В результаті сформовано перелік потенційно небезпечних операторів:

– метод eval та оператор function дозволяють виконати переданий їм код JavaScript та запустити зовнішню програму;

– об'єкт cookie надає доступ до cookie-файлів;

– объект XMLHttpRequest призначений для підтримки функціонування технології AJAX;

– метод createElement генерує об'єкт атрибута для будь-якого дескриптора HTML (або XML), зазначеного в якості параметра;

– параметр script в поєднанні з методом createElement() створює на Веб-сторінці новий скриптовий об’єкт.

– параметр iframe в поєднанні з методом createElement створює на Веб-сторінці новий плаваючий фрейм;

– метод appendChild поміщає новостворений елемент на Веб-сторінку;

– метод fromCharCode повертає рядок, що містить набір значень символів у форматі Unicode.

– метод charCodeAt повертає Unicode код символу, що знаходиться на вказаній позиції.

– метод charAt повертає символ із заданим індексом;

– метод toString повертає текстове представлення об'єкту;

– об'єкт ActiveXObject використовується для запуску об'єктів ActiveX

– параметр WSHostAccess, переданий об’єкту ActiveXObject, використовується для доступу до серверу сценаріїв ОС Windows;

– функції escape та unescape призначені для кодування/декодування текстового аргументу;

– функція write записує на Веб-сторінку текст або HTML-код, переданий як параметр.

– метод javaEnabled об’єкту navigator дозволяє визначити наявність віртуальної машини Java;

– метод plugins об’єкту navigator дозволяє визначити набір встановлених в браузері плагинів;

– оператор onClick призначений для визначення обробника подій;

– об’єкти onkeypress, onKeyDown та onKeyUp надають можливість перехоплення інформації, що вводиться користувачем з клавіатури;

– об’єкти onMouseDown та onMouseUp надають можливість перехоплення інформації, що вводиться користувачем за допомогою "мишки";

– тег дозволяє використовувати шари на Веб-сторінці;

– об’єкти File, FileReader, FileList та Blob використовуються для доступу до файлової системи локального комп’ютера;

– об'єкт location містить інформацію про місцезнаходження поточного документа, також використовується для завантаження інших документів;

– метод open, об’єкту window створює нове вікно браузера та

завантажує в нього визначений документ.

В табл. 5.4 наведено загрози від використання перерахованих потенційно небезпечних операторів. В першому наближенні перелік потенційно небезпечних операторів JavaScript складає 30 найменувань. Слід зазначити, що даний перелік може бути суттєво розширений внаслідок більш детального аналізу документації JavaScript та сигнатур Веб-орієнтованого скриптового програмного забезпечення. Особливу увагу слід звернути на ті оператори JavaScript, які використовуються для маскування шкідливого програмного коду шляхом його обфусифікації [121].

Таблиця 5.4

Загрози від потенційно небезпечних операторів JavaScript

Оператор	Загроза					
	читання/запис файлових об'єктів	передача даних	маскування ШПЗ	визначення вразливостей ПЗ	ініціалізвція деструктивних дій	завантаження/ запуск ШПЗ
eval, function, script, ActiveXObject, iframe, SHostAccess, location	-	-	-	-	-	+
File, FileReader, FileList, Blob, cookie	+	-	-	-	-	-
XMLHttpRequest	-	+	-	-	-	-
createElement, unescape, appendChild, write charAt, fromCharCode, escape, toString, charCodeAt	-	-	+	-	-	-
javaEnabled, plugins	-	-	-	+	-	-
onClick, onkeypress, , KeyDown, onKeyUp, onMouseDown, onMouseUp	-	-	-	-	+	-

Аналогічним чином проведений аналіз документованих можливостей VB дозволив отримати перелік ПБ, призначених для розпізнавання Веб-орієнтованого скриптового ШПЗ, написаного на цій мові програмування. При цьому кількість потенційно небезпечних операторів VB, а відповідно і кількість ПБ, дорівнює 105.

ПБ, пов’язані з викликами системних функції операційної системи Windows, отримані шляхом аналізу її документованих характеристик. Розглянуто тільки ті потенційно небезпечні системні функції, які безпосередньо застосовуються для маніпуляцій з файловою системою, системним реєстром, динамічною пам’яттю, процесами, потоками, мережевими з’єднаннями, службами операційної системи, параметрами безпеки об’єктів Windows та для обробки системних помилок. В результаті сформовано наступний перелік потенційно небезпечних функцій:

– ReadFile та ReadFileEx – читають дані із файлу;

– WriteFile та WriteFileEx – записують дані в файл;

– CopyFile та CopyFileEx – копіює файл та присвоює копії нове ім’я;

– MoveFile та MoveFileEx – перейменовують файл або каталог;

– DeleteFile – знищує файл;

– CreateFile – створює файл;

– CreateDirectory – створює каталог;

– SetCurrentDirectory – встановлюють визначений каталог як поточний;

– GetCurrentDirectory – повертає абсолютний шлях до поточного каталогу;

– GetSystemDirectory – повертає шлях до системного каталогу;

– GetWindowsDirectory – повертає шлях до каталогу, в якому розміщуються файли операційної системи;

– SetFilePointer – встановлює значення вказівника файлу;

– GetFileSize та GetFileSizeEx – повертають розмір файлу;

– GetCompressedFileSize – повертає розмір стисненого файлу;

– SetEndOfFile – змінює розмір файлу;

– GetFileType – дозволяє розрізняти дискові файли, символьні файли (принтери, консолі) та канали;

– GetFileAttributes – повертає атрибути файлів та каталогів;

– GetTempFileName – задає ім’я тимчасового файлу;

– LockFileEx та LockFile – блокують файл;

– UnlockFileEx – знімає блокування з файлу;

– RegOpenKeyEx – відкриває підрозділ системного реєстру;

– RegEnumKeyEx – повертає ім’я підрозділу системного реєстру;

– RegEnumValue та RegQueryValueEx – повертають назву та значення параметру в системному реєстрі;

– RegEnumKeyEx – перераховує підрозділи відкритого розділу системного реєстру та дату і час останньої модифікації зміни даного підрозділу;

– RegCreateKey та RegCreateKeyEx – створюють новий розділ системного реєстру;

– ReportError та RaiseException – використовуються для обробки помилок при виконанні системних функцій;

– RegDeleteKey видаляє вказаний ключ. Ця функція не може видалити ключ, який є підключем;

– RegDeleteValue видаляє іменоване значення із зазначеного ключа реєстру;

– RegFlushKey записує всі атрибути зазначеного відкритого ключа до реєстру;

– RegLoadKey створює підключ в HKEY_USER або HKEY_LOCAL_MACHINE і записує туди інформацію з зазначеного файлу;

– RegReplaceKey – замінює резервний файл ключа іншим файлом;

– RegSetKeySecurity встановлює безпеку для відкритого ключа реєстру;

– RegUnLoadKey вивантажує зазначений ключ і його підключі з реєстру;

– LoadLibrary та LoadLibraryEx – завантажують динамічні бібліотеки;

– FreeLibrary – звільняє динамічні бібліотеки;

– GetProcAddress – повертає адресу входу в бібліотеку;

– CreateProcess – створює новий процес;

– CreateThread – створює новий потік;

– ExitProcess, TerminateProcess – завершують процес і всі його потоки;

– ResumeThread та SuspendThread – призупиняють та відновлюють потік;

– CreateRemoteThread – створює потік в іншому процесі;

– SetPriorityClass та SetThreadPriority – встановлюють пріоритети процесу та потоку;

– WSASocket та socket – створює сокет;

– bind – співвідносить сокет з номером порту;

– listen – переводить сокет в стан прослуховування;

– accept – повертає новий підключений сокет;

– RegisterServiceCtrlHandlerEx – реєструє обробник служби;

– SetServiceStatus – повертає стан служби;

– OpenSCManager – повертає дескриптор диспетчера управління службами;

– CreateService – реєструє службу;

– StartService – запускає службу;

– ControlService – активація обробника управаляючих команд служби;

– GetUserName – повертає ім’я користувача;

– AddAccessAllowedAce та AddAccessDeniedAce – дозволяють змінювати список розмежувань прав доступу;

– GetFileSecurity – читає параметри доступу до файлових об’єктів;

– SetFileSecurity – встановлює параметри доступу до файлових об’єктів.

Загальна кількість функцій наведеного переліку, а значить, і кількість вхідних параметрів дорівнює 71. Множина ПБ може бути розширена за рахунок системних функцій, що використовуються в ШПЗ для обходу системи захисту. В подальших дослідженнях доцільно врахувати послідовність системних викликів та особливості нових версій ОС Windows.

ПБ для розпізнавання НК виду Ka_{sp} та Ka_{in}. Відповідно результатів п. 1.2, з точки зору інтелектуального аналізу даних задача розпізнавання спаму та витоків інформації зводиться до змістовного аналізу текстової частини електронних листів. Критерієм розпізнавання спаму може бути відповідність змісту електронного листа до області інтересів користувачів. Критерієм розпізнавання витоків інформації може бути приналежність змісту переданого тексту до конфіденційної інформації ІС. Виходячи з можливостей потенційних експлуатантів системи захисту та результатів [122, 250], формування області інтересів користувачів та області конфіденційної інформації в ІС можливо реалізувати за допомогою одного або декількох фрагментів тексту на природній мові. При розпізнавані спаму в якості таких фрагментів можуть використовуватися спеціальним чином оброблені цільові листи, а також безпосередньо введений текст. У випадку розпізнавання витоку інформації в якості таких фрагментів можуть використовуватись конфіденційні текстові документи. Базуючись на результатах [11, 28], визначено, що в якості вхідних ПБ слід використовувати частоти зустрічі в тексті інформативних слів в канонічній формі. Розрахунок вказаних частот пропонується реалізувати так:

$$\mu_i^j = n_i^j / N_i , \qquad (5.22)$$

де μ_i^j – частота зустрічі канонічної форми,

n_i^j – кількість всіх словоформ *j*-го слова в *i*-му тексті,

N_i – кількість слів в *i*-му тексті.

Тому ПБ будуть мати неперервний числовий характер в діапазоні від 0 до 1. Крім того, відповідно [28], до складу множини ПБ ввійшли: назва тематики тексту та відносна кількість інформативних слів. Останній параметр розраховується так:

$$I_w = I / S , \qquad (5.23)$$

де I_w – відносна кількість інформативних слів реферату листа,

I – загальна кількість інформативних слів реферату листа,

S – загальна кількість слів реферату листа.

Застосовувався цей параметр для покращення розпізнавання беззмістовних текстів.

Таким чином, кількість ПБ при розпізнаванні $\boldsymbol{Ka}_{sp}$ та $\boldsymbol{Ka}_{in}$ дорівнює

$$N_1 = K + 2 , \qquad (5.24)$$

де N_1 – кількість ПБ;

K – кількість інформативних слів у піддослідних текстах.

Під поняттям канонічної форми слова слід розуміти запис слова в такому вигляді, який дозволяє формувати будь-яку із його словоформ. Причиною застосування канонічних форм є те, що в українській та російській мовах, на які орієнтована система розпізнавання, більшість слів можуть бути представлені в декількох словоформах без зміни своєї інформативності. Для

отримання канонічних форм слів пропонується застосовувути методику [66, 67, 82, 258-268], яка передбачає використання словників словоформ.

Результати аналізу [36, 46] свідчать про те, що в українській та російській мовах кількість загальновживаних інформативних слів не перевищує 10000. Такою ж буде і кількість ПБ – вхідних параметрів НММ. При цьому, відповідно виразу (4.32), мінімальна кількість навчальних прикладів НММ дорівнює 200000, що вказує на тривалий терміну формування навчальної вибіркі та необхідність застосування значних обчислювальних потужностей для реалізації НММ. Разом з тим, кількість ПБ можливо зменшити за рахунок використання тільки тих інформативних слів, які використовуються в цільових листах та конфіденційних текстах даного користувача. Однак це призводить до необхідності адаптації множини ПБ під потреби користувача.

Визначення вхідних та вихідних параметрів НММ. Для встановлення співвідношення між ПБ на вхідним параметром НММ слід провести їх нумерацію для кожної із очікуваних видів кібератак. Номер ПБ дорівнює номеру відповідного вхідного параметру НММ.

Для збільшення гнучкості системи розпізнавання визначено, що в першому наближенні вихід НММ повинен вказувати на три можливих стани захищеності: кібератака, безпечний стан, підозрілий стан. Наприклад, до підозрілого слід віднести програмне забезпечення, в якому знайдено тільки окремі ознаки ШПЗ, наприклад, зашифрований програмний код. Реалізувати такий вихід НММ можливо за рахунок одного вихідного елементу. Встановлено, що вихід НММ (Y) при класифікації кібератак дорівнює 1, а при класифікації безпечного стану дорівнює -1. В інших випадках класифікацію можливо провести відповідно наступного виразу:

$$\begin{cases} \Delta_1 < Y \rightarrow кібератака, \\ Y < \Delta_2 \rightarrow безпечний\,стан, \\ \Delta_2 \le Y \le \Delta_1 \rightarrow підозрілийстан. \end{cases} \qquad (5.25)$$

В першому наближені Δ_1=0,5, а Δ_2=-0,5.

Оптимізація виду НММ при розпізнаванні Ka_{vb} та Ka_{js}. Для визначення значущих критеріїв оптимізації виду НММ сформовано наступний перелік основних характеристик задач розпізнавання ШПЗ:

– кількість вхідних параметрів, в якості яких передбачається використовувати імена потенційно небезпечних операторів, принципово обмежена;

– кількість навчальних прикладів може бути обмежена;

– в навчальних прикладах допустимі помилки;

– навчальні приклади можуть бути корельовані між собою;

– в навчальній вибірці неможливо відобразити сигнатури всіх шкідливих та безпечних програм;

– в навчальній вибірці можливо пропорційно представити приклади, що відповідають шкідливим та безпечним програмам;

– доцільно використання дискретних вхідних параметрів;

– обсяг навчальної вибірки може значно перевищувати кількість вхідних параметрів;

– НММ можливо навчати в лабораторних умовах, що дозволяє тривалий термін навчання;

– в навчальних прикладах можливо представити очікуваний вихідний сигнал (ШПЗ/безпечне ПЗ), що дозволяє використати навчання "з вчителем";

– процес навчання повинен бути максимально автоматизованим;

– бажана, однак необов’язкова, можливість донавчання НМ;

– якість навчання та обсяг пам’яті НМ мають бути максимально високими;

– необхідно забезпечити екстраполяцію результатів навчання за межі навчальної вибірки;

– результати навчання мають бути незмінними;

– слід забезпечити можливість інтерпретації вихідної інформації НМ у вигляді ймовірності;

– інтерпретація виходу НМ у графічному вигляді не обов'язкова;

– вербалізація НМ не обов'язкова;

– навчена НМ повинна максимально швидко проводити класифікацію;

– обсяг програмної реалізації не має суттєвого значення;

– сфера застосування розроблюваної НМ відноситься до розпізнавання образів;

– пристосованість до автономного функціонування необов'язкова.

Таким чином, множина значущих критеріїв визначається виразом:

$$\left\{E_{1,3}, E_{1,4}, E_{1,7}, E_{2,3}, E_{2,4}, E_{2,5}, E_{3,1}, E_{3,2}, E_{4,1}, E_{4,3}, E_{5,1}, E_{6,1}\right\}, \qquad (5.26)$$

де E – критерії оптимізації, наведені в табл. 2.1

Використавши дані табл. 2.3, 2.4 та вираз (2.26), визначено, що оптимальним видом НММ являється БШП, для якого інтегральний критерій ефективності має максимальну величину $E_{\Sigma} = 12$. В розрахунках (2.26) прийнято, що критерії (5.26) мають однакову значимість.

Оптимізація виду НММ при розпізнаванні Ka_{sp} та Ka_{in}. Перелік основних характеристик задач розпізнавання спаму та витоків текстової інформації має наступний вигляд:

– виконання вимоги використання навчальних прикладів з необмеженою кількістю вхідних параметрів не є обов'язковим, оскільки в результаті аналізу зібраних статистичних даних виявлено, що приблизна кількість інформативних слів за якою можливо виявити тему електронного листа, а значить, і кількість вхідних параметрів – 1000;

– кількість навчальних прикладів може бути обмежена;

– в навчальних прикладах допустимі помилки;

– навчальні приклади можуть бути корельовані між собою;

– в навчальній вибірці неможливо відобразити всі можливі комбінації інформативних слів для різних типів текстів;

– в навчальній вибірці складно пропорційно представити приклади, що відповідають безпечним, підозрілим та забороненим текстам;

– доцільно використання неперервних вхідних параметрів;

– в багатьох випадках обсяг навчальної вибірки може бути меншим від кількості вхідних параметрів;

– НММ повинна враховувати специфіку конкретного користувача, а значить, не може бути навчена в лабораторних умовах. Тому одним із найбільш важливих критеріїв оптимізації є забезпечення короткого терміну навчання;

– в навчальних прикладах можливо представити очікуваний вихідний сигнал (безпечний/підозрілий/заборонений текст), що дозволяє використати навчання "з вчителем";

– процес навчання повинен бути максимально автоматизованим;

– НММ повинна оперативно реагувати на нові види тексту, не представлені в навчальній вибірці, тому обов’язковою є можливість донавчання НММ. Слід зазначити, що альтернативним шляхом оперативного реагування є повне перенавчання НМ. Такий підхід можна реалізувати для НМ з коротким терміном навчання;

– оскільки в системі класифікації передбачено використання буферного класу підозрілих текстів, а одиничні факти неправильної класифікації як в задачі розпізнавання спаму, так і в задачі розпізнавання витоків, не призводять до катастрофічних наслідків, то вимоги до якості навчання НММ можуть бути не надто високими;

– виходячи з того, що для клієнтських систем розпізнавання очікувана кількість навчальних прикладів знаходиться в межах 1000, вимоги до обсягу пам’яті НММ не високі;

– оскільки в навчальній вибірці можливо представити практично всі

типи безпечних текстів, то забезпечення екстраполяції результатів навчання за межі навчальної вибірки є бажаною, але не обов'язковою умовою;

– результати навчання мають бути незмінними;

– слід забезпечити можливість інтерпретації вихідної інформації НММ у вигляді ймовірності;

– оскільки в сучасних системах якість автоматичної класифікації текстової інформації не повністю задовольняє практичним вимогам, то доцільна інтерпретація виходу НММ у графічному вигляді. За рахунок цього користувач отримає можливість автоматизованої класифікації листів.

– вербалізація НМ не обов'язкова;

– навчена НМ повинна максимально швидко проводити класифікацію;

– обсяг програмної реалізації не має суттєвого значення;

– сфера застосування розроблюваної НМ відноситься до сфери аналізу текстової інформації;

– пристосованість до автономного функціонування за рахунок автоматизації навчання необов'язкова.

Таким чином, множина значущих критеріїв визначається виразом:

$$\left\{E_{1,3}, E_{1,4}, E_{1,6}, E_{1,8}, E_{1,9}, E_{2,1}, E_{2,3}, E_{2,4}, E_{2,7}, E_{3,1}, E_{4,1}, E_{4,2}, E_{5,1}, E_{6,2}\right\}, \qquad (5.27)$$

де E – критерії оптимізації, наведені в табл. 2.1.

Шляхом експертного оцінювання визначено, що найбільш значимими в (2.27) є критерії,які характеризують :

– $E_{2,1}$ – короткий термін навчання;

– $E_{4,2}$ – можливість інтерпретації виходу у графічному вигляді;

– $E_{6,2}$ – апробованість в задачах аналізу текстової інформації.

Використавши дані табл. 2.3., 2.4 та вираз (2.26), визначено, що в першому наближенні оптимальними видом НММ є ТК, для якої інтегральний

критерій ефективності має максимальну величину $E_{\Sigma} = 3{,}8$. В розрахунках (2.26) прийнято, що вагові коефіцієнти для критеріїв $E_{2,1}$, $E_{4,2}$, $E_{6,2}$ дорівнюють $r_{2,1} = r_{4,2} = r_{6,2} = 1$, а вагові коефіцієнти інших критеріїв дорівнюють 0,1.

Застосування експертних даних для розпізнавання. Крім описаних НММ, для розпізнавання множини всіх очікуваних видів кібератак в системі передбачено використання мереж MPNN, що функціонують на основі експертних даних у вигляді продукційних правил. Таким чином, кількість мереж MPNN дорівнює кількості очікуваних видів кібератак. Для сторення MPNN застосовано наведний в п. 4.1 метод. При цьому для кожного конкретного випадку класифікації електронних листів для розпізнавання спаму та витоків інформації слід створювати свої продукційні правила. Оскільки характеристики спаму і витоків в значній мірі залежать від специфіки ІС. Розроблено ряд продукціних правил для розпізнавання скриптового ШПЗ. Приклад одного із продукційних правил для визначення небезпечного скрипта, написаного на VB, виглядає так:

Якщо On Error Resume Next=1∧CreateObject=1∧Outlook.Application=1 ∧WScript=1∧Send=1→ШПЗ.

Дане правило отримане на основі аналізу програмного коду шкідливого скрипта, який розповсюджувався в листах електронної пошти і призначався для розповсюдження спаму. Розробка універсальних продукційних правил для розпізнавання веб-орієнтованого скриптового ШПЗ на основі аналізу коду скрипта та системних викликів є стати темою подальших досліджень.

Розробка НМС. Проведені дослідження дозволили розробити показану на рис. 5.4 структуру НМС розпізнавання Веб-орієнтованого шкідливого програмного забезпечення, спаму та витоків текстової інформації. Ця НМС складається із наступних частин:

– ПАВД – підсистема аналізу вхідних даних, за рахунок якої формується навчальна вибірка та формується множина вхідних параметрів НММ;

– ПЕО – підсистема експертної оцінки, в якій на основі експертих даних визначається вихідний сигнал для навчальних прикладів та формуються продукційні правила, що характеризують стан захищеності;

– ПРС – підсистема розпізнавання та сигналізації, за рахунок якої реалізується розпізнавання кібератак та сигналізація про стан захищеності;

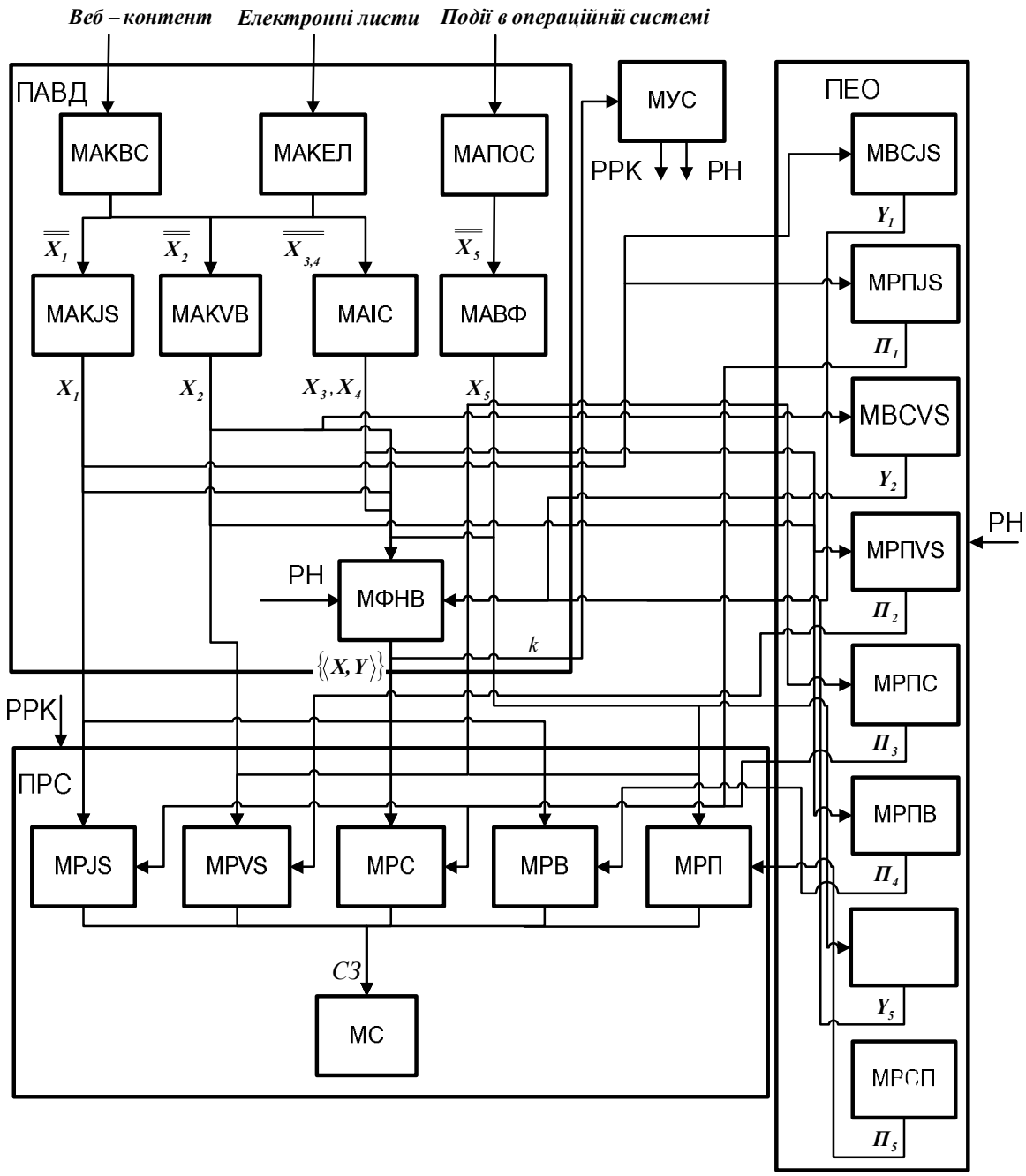


Рис. 5.4. Структура системи розпізнавання шкідливого програмного забезпечення та класифікації листів електронної пошти

– МУС – модуль управління системою, що служить для переведення системи в наступні режими функціонування: РН – навчання НММ; РРК – розпізнавання кібератак.

Призначення окремих модулів розробленої НМС наведені в табл.5.5.

Таблиця 5.5

Склад нейромережевої системи розпізнавання ШПЗ та класифікації листів електронної пошти

Назва підсистеми	Назва модулю	Призначення модулю
1	2	3
ПАВД	МАКВС	Аналіз контенту Веб-сайтів для визначення в ньому операторів JavaScript ($\overline{\overline{\boldsymbol{X}}}_1$) та VB ($\overline{\overline{\boldsymbol{X}}}_2$)
	МАКЕЛ	Аналіз контенту електронних листів для визначення в ньому інформативних слів в канонічній формі ($\overline{\overline{\boldsymbol{X}}}_{3,4}$)
	МАПОС	Аналіз подій в операційній системі для визначення найменувань викликів системних функцій ($\overline{\overline{\boldsymbol{X}}}_5$)
	МАКJS	Аналіз коду JavaScript для визначення в ньому потенційно небезпечних функцій ($\boldsymbol{X}_1$)
	МАКVB	Аналіз коду VB для визначення в ньому потенційно небезпечних функцій ($\boldsymbol{X}_2$)
	МАІС	Аналіз множини інформативних слів з метою визначення в ній інформативних слів, що використовуються для розпізнавання спаму ($\boldsymbol{X}_3$) та витоків інформації ($\boldsymbol{X}_4$)
	МАВФ	Аналіз викликів операційної системи для визначення в них небезпечних функцій ($\boldsymbol{X}_5$)

Таблиця 5.5 (продовження)

1	2	3
	МФНВ	Формування навчальної вибірки ($\{\langle \boldsymbol{X},\boldsymbol{Y} \rangle\}$)
ПЕО	МВСJS, МВСVS,	Визначення вихідних сигналів НММ для розпізнавання ШПЗ JavaScript ($\boldsymbol{Y}_1$) та VB ($\boldsymbol{Y}_2$)
ПЕО	МРПJS, МРПVB	Розробки продукційних правил для розпізнавання ШПЗ JavaScript ($\boldsymbol{\Pi}_1$) та VB ($\boldsymbol{\Pi}_2$)
	МРПС, МРПВ	Розробки продукційних правил для розпізнавання спаму ($\boldsymbol{\Pi}_3$) та витоків текстової інформації ($\boldsymbol{\Pi}_4$)
	МВСП	Ввизначення вихідних сигналів НММ для розпізнавання поведінки ШПЗ ($\boldsymbol{Y}_5$)
	МРПП	Розробки продукційних правил для розпізнавання поведінки ШПЗ ($\boldsymbol{\Pi}_5$)
ПРС	МРJS, МРVS, МРС, МРВ, МРП,	Розпізнавання ШПЗ JavaScript, VB, спаму, витоків текстової інформації, розпізнаваня поведінки ШПЗ
	МС	Сигналізації

Функціонуваня системи розпізнавання ШПЗ та класифікації листів електронної пошти в режимі навчання та в режимі розпізнавання відповідає функціонуванню системи оцінювання ПБ для розпізнавання кібератак на ресурси Інтернет-орієнтованих ІС з урахуванням описаних особливостей структури та використаних НММ.

Експериментальні дослідження.

Для розпізнавання веб-орієнтованого скриптового ШПЗ, написаного на JavaScript, сформовано навчальну вибірку з 1000 прикладів. Крім ШПЗ, як до навчальної, так і до тестової вибірки включені безпечні скрипти.

Фрагментарно навчальні дані наведені в табл. 5.6. Використано ДШП з 30 вхідними та одним вихідним нейроном. З використанням розробленої моделі БШП визначено, що оптимальна кількість схованих нейронів $N_1^{opt} = 80$. При цьому кількість синаптичних зв'язків

$$L_w = 30 \times 80 + 80 = 2480. \quad (5.28)$$

Для вхідних елементів вибрано лінійну функцію активації, а для схованих елементів – сигмоїдальну з a=0,1. Навчання ДШП реалізоване методом оберненого розповсюдження помилок.

Таблиця 5.6

Навчальні дані для розпізнавання ШПЗ, написаного на JavaScript

Номер прикладу	Вихідний сигнал *Y*	Вхідні параметри															
		eval	function	script	ActiveXObject	iframe	location	File	FileReader	FileList	Blob	cookie	XMLHttpRequest	javaEnabled	plugins	unescape	escape
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	-1	-1	1	1	1	1	1	-1	-1	1	-1
3	1	1	1	1	1	1	-1	-1	1	1	1	1	1	-1	-1	1	-1
4	1	1	1	1	-1	1	-1	-1	1	1	-1	-1	-1	-1	-1	-1	1
5	1	1	1	1	-1	1	-1	-1	1	1	-1	-1	-1	1	-1	1	1
6	-1	1	-1	-1	1	-1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
7	-1	1	1	-1	1	-1	-1	-1	1	1	1	1	1	-1	-1	1	-1
8	-1	-1	1	-1	1	-1	1	-1	-1	-1	1	-1	-1	-1	1	-1	-1
9	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1
10	-1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1	1	-1	-1	-1	-1	-1

Експерименти здійснювались за допомогою розробленого Windows-застосунку, опис якого наведено в додатку А на персональному комп'ютері з процесором Intel Core Quard з тактовою частотою 2,4 ГГц та обсягом оперативної пам'яті 3,5ГБ. Термін навчання склав 58 с, кількість ітерацій 705, а середня відносна похибка навчання $5{,}2\times10^{-7}$. Пошук оптимуму здійснено методом градієнтного спуску.

Після навчання ДШП були пред'явлені 20 тестових прикладів, отриманих шляхом зміни деяких параметрів навчальних прикладів. Кількість таких параметрів для кожного з прикладів знаходиться в межах від 3 до 10. Результати розпізнавання тестових прикладів представлені в табл. 5.7 та в табл. 5.8.

Таблиця 5.7

Результати розпізнавання ДШП тестових прикладів скриптів JavaScript

№ прикладу	Похибка виходу	№ прикладу	Похибка виходу
Скриптове ШПЗ		Безпечні скрипти	
1	0,003523	11	0,012763
2	0,002787	12	0,064381
3	0,025412	13	0,004235
4	0,417221	14	0,021688
5	0,004672	15	0,011725
6	0,008214	16	0,237191
7	0,022722	17	0,003257
8	0,010452	18	0,008125
9	0,026932	19	0,017351
10	0,014956	20	0,33576

При використанні правила класифікації ШПЗ виду (5.25) допустимою є похибка δ<0,5. Виходячи з цього дані табл. 5.6 вказують на те, що всі тестові приклади розпізнані правильно.

В табл. 5.8 представлені максимальна (δ_{max}^{p}) та середня помилки (δ_{s}^{p}) виходу ДШП при тестуванні JavaScript-скриптів. Аналіз даних табл. 5.8 показує, що максимальна та середня помилка виходу знаходиться в допустимих межах – величини δ_{max}^{p} і δ_{s}^{p} менші ніж 0,5. Таким чином, результати проведених експериментів підтверджують доцільність використання ДШП для розпізнавання Веб-орієнтованого скриптового ШПЗ, написаного на мові програмування JavaScript. В порівнянні з відомими антивірусними сканерами [48, 87, 121] помилки неправильного розпізнавання вірусу та неправильної класифікації безпечних програм зменшилась на 5-10%.

Таблиця 5.8

Максимальна та середня помилка виходу ДШП при розпізнаванні скриптів JavaScript

Розпізнаний клас	δ_{max}^{p}	δ_{s}^{p}
Скриптове ШПЗ	0,3454137	0,0225612
Безпечні скрипти	0,1722524	0,0053734
Для всіх тестових прикладів	0,3454137	0,013967

Для розпізнавання Веб-орієнтованого скриптового ШПЗ, написаного на VB, сформовано навчальну вибірку з 2100 прикладів, фрагментарно показану в табл. 5.9. Для розпізнавання використано ДШП з 1050 вхідними та одним вихідним нейроном. З використанням розробленої моделі БШП визначено, що оптимальна кількість схованих нейронів для такого ДШП становить $N_1^{opt} = 250$. При цьому, кількість синаптичних зв'язків

$$L_w = 105 \times 250 + 250 = 26355. \qquad (5.29)$$

Інші параметри ДШП та умови проведення експеримнету ті ж самі, що і при розпізнаванні ШПЗ, написаного на JavaScript. Термін навчання склав

117 с, кількість ітерацій 1606, а середня відносна похибка навчання $8{,}2\times10^{-6}$.

Після навчання ДШП були пред'явлені 40 тестових прикладів, отриманих шляхом зміни деяких параметрів навчальних прикладів. Кількість таких параметрів для кожного з прикладів знаходиться в межах від 3 до 65. Результати розпізнавання тестових прикладів представлені в табл. 5.10 та в табл. 5.11.

Дані табл. 5.10, 5.11 свідчать, що як і у випадку розпізнавання ШПЗ, написанного на JavaScript, всі тестові приклади розпізнані правильно, а помилка виходу знаходиться в допустимих межах і не впливає на достовірність класифікації.

Таблиця 5.9

Навчальні дані для розпізнавання ШПЗ, написаного на VB

Номер прикладу	Вихідний сигнал *Y*	Вхідні параметри																
		CreateObject	OpenTextFile	ScriptFullname	readline	readall	CreateTextFile	Writeline	write	Close	deletefile	Drives	DriveType	GetFolder	SubFolders	path	Files	getfile
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	-1	-1	1	1	1	1	1	-1	-1	1	-1	1
3	1	1	1	1	1	1	-1	-1	1	1	1	1	1	-1	-1	1	-1	1
4	1	1	1	1	-1	1	-1	-1	1	1	-1	-1	-1	-1	-1	-1	1	-1
5	1	1	1	1	-1	1	-1	-1	1	1	-1	-1	-1	1	-1	1	1	-1
6	-1	1	-1	-1	1	-1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
7	-1	1	1	-1	1	-1	-1	-1	1	1	1	1	1	-1	-1	1	-1	-1
8	-1	-1	1	-1	1	-1	1	-1	-1	-1	1	-1	-1	-1	1	-1	-1	-1
9	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1
10	-1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1	1	-1	-1	-1	-1	-1	-1

Таблиця 5.10

Результати розпізнавання ДШП тестових прикладів скриптів VB

№ прикладу	Похибка виходу	№ прикладу	Похибка виходу
Скриптове ШПЗ			
1	0,240763	11	0,000061
2	0,016792	12	0,464983
3	0,006402	13	0,086452
4	0,006528	14	0,011643
5	0,004527	15	0,001076
6	0,000233	16	0,316679
7	0,000239	17	0,002276
8	0,128197	18	0,014845
9	0,042734	19	0,048387
10	0,005533	20	0,43776
Безпечні програми			
21	0,0000103	31	0,0000076
22	0,00001	32	0,000004
23	0	33	0,0000041
24	0,0000103	34	0,00001
25	0,0000073	35	0,000006
26	0,000013	36	0,000007
27	0,000001	37	0,0000058
28	0,0000169	38	0,0000089
29	0,000002	39	0,000021
30	0,000007	40	0,054193

Порівняння похибок розпізнавання розробленого ДШП з відомими антивірусними засобами показало, що похибка пропуску ШПЗ зменшилась на 5-7%, а похибка неправильної класифікації безпечного скриптового

програмного забезпечення зменшилась на 3-5%.

Таблиця 5.11

Максимальна та середня помилка виходу ДШП при розпізнаванні скриптів VB

Розпізнаний клас	δ_{max}^{p}	δ_{s}^{p}
Скриптове ШПЗ	0,4649833	0,0918054
Безпечні скрипти	0,0541930	0,0027173
Для всіх тестових прикладів	0,4649833	0,047261

Для розпізнавання ШПЗ на основі аналізу викликів системних функцій сформовано навчальну вибірку з 710 прикладів, фрагментарно показану в табл. 5.12.

Таблиця 5.12

Навчальні дані НМ поведінкового аналізатора

Номер прикладу	Вихідний сигнал *Y*	Вхідні параметри														
		ReadFile	ReadFileEx	WriteFile	WriteFileEx	CopyFile	CopyFileEx	MoveFile	MoveFileEx	DeleteFile	CreateFile	CreateDirectory	SetCurrentDirectory	GetCurrentDirectory	GetSystemDirectory	GetWindowsDirectory	SetFilePointer
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	-1	-1	1	1	-1	-1	1	-1	-1	1	-1	1	1	-1	1	1
3	1	1	-1	1	-1	1	-1	1	-1	1	-1	-1	1	1	1	1	1
4	1	-1	1	-1	-1	1	-1	1	1	1	-1	1	1	-1	-1	1	1
5	1	1	-1	-1	1	-1	-1	-1	-1	-1	1	-1	1	1	1	1	-1

Для розпізнавання використано ДШП з 71 вхідними та одним вихідним нейроном. З використанням розробленої моделі БШП визначено, що оптимальна кількість схованих нейронів для такого ДШП становить $N_1^{opt} = 100$. При цьому, кількість синаптичних зв'язків $L_w = 71 \times 100 + 100 = 7800$. Інші параметри ДШП та умови проведення експериментеу ті ж самі, що і при розпізнаванні ШПЗ, написаного на JavaScript. Термін навчання – 85 с, кількість ітерацій – 1405, а середня відносна похибка навчання – $4{,}3 \times 10^{-7}$.

Після навчання ДШП були пред'явлені 20 тестових прикладів, отриманих шляхом зміни деяких параметрів навчальних прикладів. Кількість таких параметрів для кожного з прикладів знаходиться в межах від 3 до 10.

Результати розпізнавання тестових прикладів представлені в табл. 5.13 та в табл. 5.14.

Таблиця 5.13

Результати розпізнавання ДШП тестових прикладів в поведінковому аналізаторі

№ прикладу	Похибка виходу	№ прикладу	Похибка виходу
ШПЗ		Безпечне ПЗ	
1	0,005301	11	0,021644
2	0,000323	12	0,237461
3	0,148195	13	0,436173
4	0,210731	14	0,010221
5	0,007426	15	0,003032
6	0,032835	16	0,012423
7	0,001530	17	0,036394
8	0,016543	18	0,086394
9	0,007531	19	0,764384
10	0,004774	20	0,0021895

Таблиця 5.14

Помилка виходу ДШП в поведінковому аналізаторі

Розпізнаний клас	δ_{max}^{p}	δ_{s}^{p}
ШПЗ	0,4759628	0,1442188
Безпечне ПЗ	0,2754522	0,0154855
Для всіх тестових прикладів	0,4759628	0,079852

Дані табл. 513, 5.14 свідчать, що як і у випадку розпізнавання ШПЗ, написанного на JavaScript та VBScript, всі тестові приклади розпізнані правильно, а помилка виходу знаходиться в допустимих межах і не впливає на достовірність класифікації. В порівнянні з відомими поведінковими аналізаторами антивірусних систем [1, 5, 276] помилка неправильного розпізнавання ШПЗ та неправильної класифікації безпечних програм зменшилась на 5-10%..

Для розпізнавання спаму сформовано навчальну вибірку з 100 прикладів. В якості статистичного матеріалу було використано електронні листи по темі запрошення на семінари, реклами побутових послуг та промислових товарів. Листи були отримані автором на протязі декількох тижнів 2014 року. Можна вважати, що листи однієї із вказаних тематик є цільовими, а інші листи – спам.

Попередній аналіз статистичного матеріалу виявив, що кількість слів в канонічній формі в отриманих електронних листах не перевищує 1000. В багатьох випадках листи однієї тематики не значно відрізнялись між собою. Наприклад, було отримано 12 листів з запрошенням відвідати семінар по темі "Логістика". Різниця між листами полягала тільки в даті проведення семінару, а перелік інформативних слів залишився незмінним. З точки зору розпізнавання спаму, означені листи повинні відноситись до одного класу. Тому листи з однаковим набором інформативних слів були виділені в окремі групи. Темі "реклама послуг" відповідає група листів №1, темі "запрошення на семінари" відповідають групи листів №2,3,4,5,6,7,9,12,14, темі "реклама

промислових товарів" – №8,10,11,13.

Зміст та умовна класифікація груп листів представлені в табл. 5.15. Після розрахунку кількості канонічних форм інформативних слів за допомогою (5.23) були розраховані частоти цих слів для кожної із груп отриманих листів. Фрагмент вхідних даних показаний в табл. 5.16.

Таблиця 5.15

Тематика груп листів

№ групи	Зміст групи листів	Тематика групи листів
1	Реклама супутникової антени	Реклама послуг
2	Семінар по темі "Передача житлового будинку в експлуатацію"	Запрошення на семінари
3	Семінар по темі "Як знайти клієнта телефону?"	
4	Семінар по темі "Кодекс адміністративного судочинства України"	
5	Семінар по темі "Сучасні підходи логістики"	
6	Семінар по темі "Сучасний маркетинг"	
7	Семінар по темі "Сучасний менеджмент"	
9	Семінар по темі "Психологія споживача"	
12	Семінар по темі "Судові спори з податковими органами"	
14	Семінар по темі "Збільшення власного грошового потоку"	
13	Реклама плазмового телевізора	Реклама промислових товарів
8	Реклама обігрівача	
10	Реклама охоронної сигналізації	
11	Реклама систем відеонагляду	

Моделювання ТК здійснювалось на основі загальної методики проектування НМ [210]. При її побудові прийнято: розмір топографічної сітки – (16×12), форма сітки зв'язків – гексагон, кількість кластерів – 5, кількість навчальних епох – 500, η=0.1, r=6 на початку навчання, η=0.005, r=1 в кінці навчання. Для програмної реалізації ТК використано пакет Deductor Studio.

Таблиця 5.16

Навчальні дані ТК при розпізнаванні спаму

№ прикладу	Вхідні параметри					
	Відносна кількість інформативних слів	СТОИМ	ГРН	ДОМ	ДЕНЬ	ГОД
1	0,12	0,04	0,04	0	0	0
2	0,35	0	0,0148	0,051	0	0
3	0,2759	0	0	0	0	0
4	0,219	0	0	0	0	0
5	0,2336	0,0263	0,0263	0	0	0,0263
6	0,4211	0	0	0	0	0,0238
7	0,1569	0	0	0	0,053	0
8	0,0256	0	0	0	0	0
9	0,2632	0	0	0	0	0
10	0,0938	0	0	0	0	0
11	0,1136	0	0	0	0	0

Розділена на кластери ТК представлена на рис. 5.5. На рис. 5.5 межі кластерів показані неперервною лінією, а межі комірок показані пунктиром. Кластери пронумеровані буквами A, B, C, D, E, а групи листів з однаковим набором інформативних слів – цифрами від 1 до 14. Відповідно, цифри 8, 10, 11, 13 відповідають групам листів по темі "реклама промислових товарів", а номер 1 відповідає листам по темі "реклама побутових послуг". Всі інші

листи є запрошеннями на семінари.

Таким чином, ТК якісно розділила листи на дві основні теми – реклама (кластери C та D) та запрошення на конференції (кластери A,B,E). Проте не достовірно віднесла до одного кластеру листи з рекламою промислових товарів та листи з рекламою побутових послуг. Однак якість відображення однотипних листів за допомогою ТК дозволяє провести їх приблизну класифікацію самим користувачем.

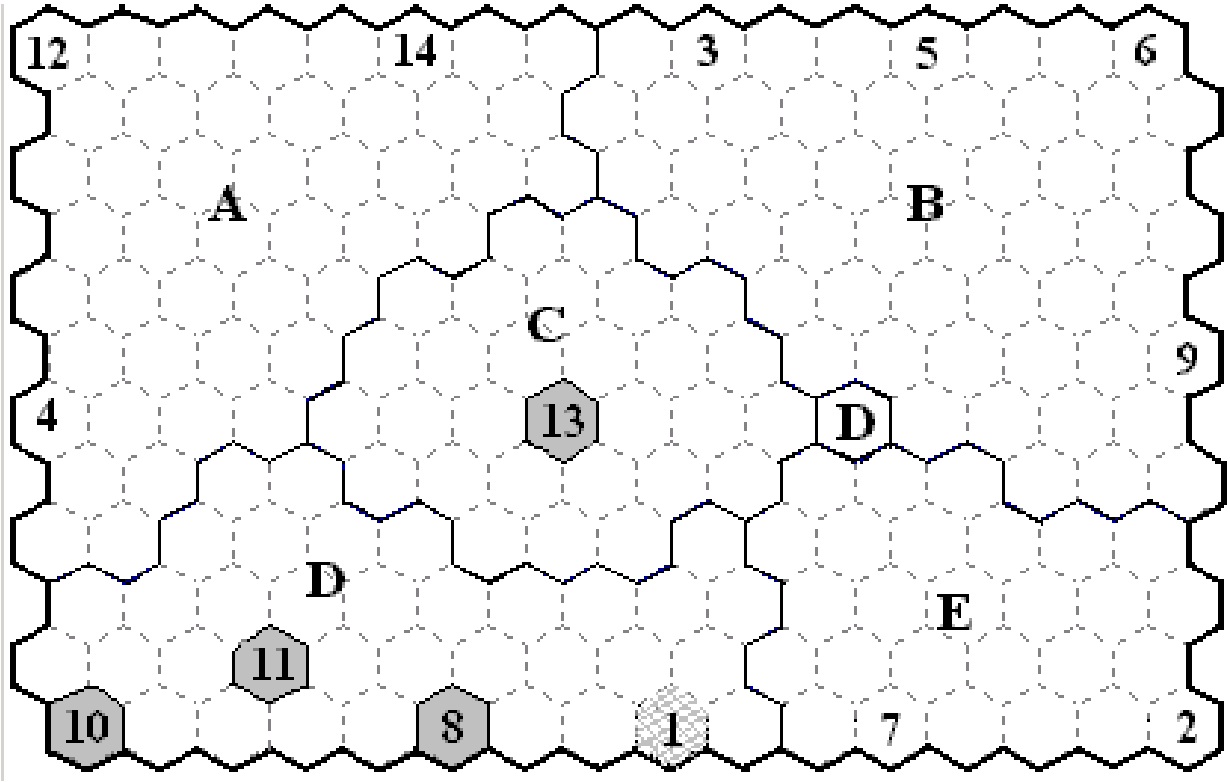


Рис. 5.5. Топографічна карта Кохонена в задачі розпізнавання спаму

Також проведені експерименти спрямовані на апробацію розробленої системи в задачі розпізнавання витоку текстової інформації. В якості статистичного матеріалу було використано 30 текстових документів, які входять до бази даних DLP-системи "Контур інформаційної безпеки". Тематика документів: резюме, бухгалтерські звіти та опис технічних характеристик різноманітної побутової техніки.

Попередній аналіз статистичного матеріалу виявив, що кількість слів в канонічній формі в цих документах не перевищує 500. Після розрахунку кількості канонічних форм інформативних слів за допомогою (5.23) для кожного із документів були розраховані частоти цих слів. Фрагмент вхідних даних показаний в табл. 5.17.

Таблиця 5.17

Навчальні дані ТК при розпізнаванні витоків текстової інформації

№ прикладу	Вхідні параметри			
	Відносна кількість інформативних слів	СТАЖ	ОСВІТА	НАРОДИВСЯ
1	0,433	0	0,02	0
2	0,412	0,03	0,04	0
3	0,527	0	0	0
4	0,493	0	0	0,001
5	0,631	0	0	0
6	0,521	0,04	0,0263	0
7	0,685	0	0	0
8	0,375	0	0	0,0263
9	0,563	0,021	0	0
10	0,675	0	0,004	0

Як і в задачі розпізнавання спаму, моделювання ТК здійснювалось на основі загальної методології проектування НМ [210].

При побудові ТК прийнято: розмір топографічної сітки – (16×12), форма сітки зв'язків – гексагон, кількість кластерів – 3, кількість навчальних епох – 200, η=0.1, r=6 на початку навчання, η=0.005, r=1 в кінці навчання. Розділена на кластери ТК показана на рис. 5.5. На рис. 5.5 межі кластерів показані неперервною лінією, а межі комірок показані пунктиром. Кластери пронумеровані буквами від A, B, C а документи – цифрами від 1 до 30.

Кластер А та документи з номерами від 1 до 10 відповідають темі – "резюме", кластер В та документи з номерами від 11 до 20 відповідають "бухгалтерським звітам", а кластер С та документи з номерами від 21 до 30 – "опис побутової техніки". Аналіз рис. 5.6 дозволяє стверджувати, що ТК в цілому правильно розділила документи на 3 основних теми: резюме, бухгалтерські звіти та опис побутової техніки. Також помітні окремі помилки кластеризації. Наприклад, документ №23, в якому наведено опис телевізора, не правильно віднесено до кластеру А, який відповідає темі "резюме".

Разом з тим, як і у випадку розпізнавання спаму, якість відображення однотипних документів за допомогою ТК дозволяє провести їх приблизну класифікацію адміністратором системи безпеки.

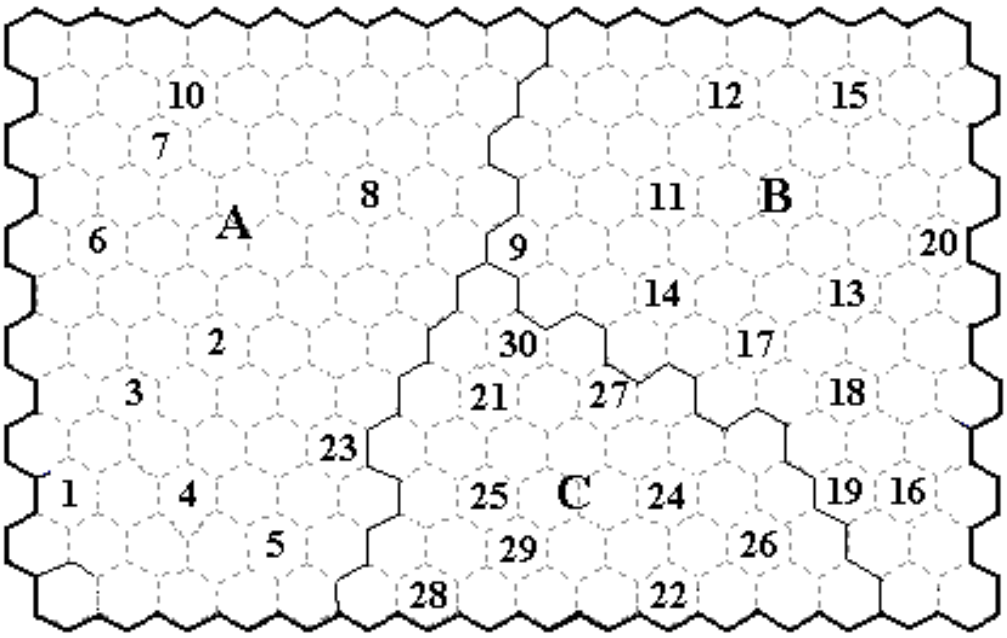


Рис. 5.6. Топографічна карта Кохонена в задачі розпізнавання витоків інформації

Таким чином, результати чисельних експериментів підтверджують перспективність застосування розробленої нейромережевої системи для розпізнавання Веб-орієнтованого скриптового ШПЗ та класифікації листів електронної пошти.

5.4. Система розпізнавання мережевих кібератак

Проектування системи розпізнавання мережевих кібератак проведено за допомогою комплексної методології оцінювання ПБ ІС із застосованнням структурних рішень системи оцінювання ПБ для розпізнавання кібератак на ресурси Інтернет-орієнтованих ІС.

У відповідності із результатами п. 1.2 та доступними статистичними даними, розглянуто кібератаки типу шторм запитів (Ka_z), спрямовані викликати відмову в обслуговувані веб-сервера та кібератаки типу U2R, спрямовані на несанкціоноване підвищення привілеїв користувачів. В свою чергу, відповідно механізму реалізації [1, 7], із кібератак типу U2R виділено наступні підтипи кібератак: buffer_overflow (Ka_b), loadmodule (Ka_l), perl (Ka_p), rootkit (Ka_r).

Таким чином, піддослідна множина кібератак визначалсь виразом:

$$\boldsymbol{Ka} = \{Ka_z,\ Ka_b,\ Ka_l, Ka_p, Ka_r\}, \tag{5.30}$$

Джерелом статистики для Ka_z були дані, наведені в [184], а для Ka_b, Ka_l, Ka_p, Ka_r – база даних KDD-99.

Формування множини ПБ. Відповідно комплексній методології, на першому етапі проектування системи проведено формування множини ПБ, оцінювання яких дозволить розпізнати очікувану множину кібератак. Відповіно результатів п. 4.3, для розпізнавання Ka_s використано кількість запитів до веб-серверу за одну секунду (Xs_z). Оскільки Xs_z залежить від терміну експлуатації Веб-серверу, то Ka_s відноситься до ПК, що дозволяє створити марківську модель ШНП.

Для розпізнавання $\{Ka_b, Ka_l, Ka_p, Ka_r\}$ в якості ПБ використано множину параметрів мережевого з'єднання ($\boldsymbol{Xq}_{U2R}$), представлену в базі

даних KDD-99.

Записи бази даних KDD-99, що відповідають мережевим кібератакам типу U2R, наведені в табл. 5.17.

Таблиця 5.17

Приклади величин ПБ для розпізнавання кібератак типу U2R

Кібератака	Значення параметрів в базі даних KDD-99
1	2
buffer_overflow	184, tcp, telnet, SF, 1511, 2957, 0, 0, 0, 3, 0, 1, 2, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 1, 3, 1.00, 0.00, 1.00, 0.67, 0.00, 0.00, 0.00, 0.00
buffer_overflow	305, tcp, telnet, SF, 1735, 2766, 0, 0, 0, 3, 0, 1, 2, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 2, 4, 1.00, 0.00, 0.50, 0.50, 0.00, 0.00, 0.00, 0.00
buffer_overflow	305, tcp, telnet, SF, 1684, 2378, 0, 0, 0, 3, 0, 1, 2, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 2, 3, 1.00, 0.00, 0.50, 0.50, 0.00, 0.00, 0.00, 0.00
loadmodule	79, tcp, telnet, SF, 281, 1301, 0, 0, 0, 2, 0, 1, 1, 1, 0, 0, 4, 2, 0, 0, 0, 0, 1, 1, 0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 1, 10, 1.00, 0.00, 1.00, 0.30, 0.00, 0.00, 0.00, 0.10
loadmodule	103, tcp, telnet, SF, 302, 8876, 0, 0, 0, 2, 0, 1, 4, 1, 0, 3, 4, 2, 1, 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
loadmodule	103, tcp, telnet, SF, 290, 8154, 0, 0, 0, 2, 0, 1, 4, 1, 0, 3, 4, 2, 1, 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
perl	25, tcp, telnet, SF, 269, 2333, 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, 69, 2, 0.03, 0.06, 0.01, 0.00, 0.00, 0.00, 0.00, 0.00

Таблиця 5.17 (продовження)

1	2
perl	25, tcp, telnet, SF, 263, 2547, 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, 69, 2, 0.03, 0.06, 0.01, 0.00, 0.00, 0.00, 0.00, 0.00
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
rootkit	60, tcp, telnet, SF, 86, 183, 0, 0, 0, 0, 1, 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	60, tcp, telnet, SF, 90, 233, 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, 2, 0.01, 0.02, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00

Зазначимо, що механізм реалізації $\{Ka_b, Ka_l, Ka_p, Ka_r\}$ та характер $\boldsymbol{Xq}_{U2R}$ не залежать від терміну функціонування ІС. Тому ці кібератаки відносяться до НК, що не передбачає створювати для їх розпізнавання марківської моделі ШП.

Визначення вхідних та вихідних параметрів НММ.

Оскільки при розпізнаванні Ka_s можливо використати ШНП, то в якості першого вхідного параметру НММ використано приведену різницю між реальною кількістю звернень до веб-серверу (R_r) та кількістю звернень, розрахованих відповідно до ШНП (R_m):

$$n_1 = (R_r - R_m) / R_m . \qquad (5.31)$$

Другий вхідний параметр призначений для синхронізації ШНП з

реальною кількістю звернень і розраховується як приведений час звернення до веб-серверу:

$$n_2 = t - Round(t/24), \tag{5.32}$$

де t – момент звернення.

При розпізнаванні мережевих кібератак типу U2R, як і у випадку розпізнавання ШПЗ, для встановлення співвідношення між ПБ на вхідним параметром НММ проведено їх нумерацію для кожного із очікуваних видів кібератак.

Номер ПБ дорівнює номеру відповідного вхідного параметру НММ. Таким чином, в НММ, призначених для розпізнавання Ka_s, кількість вхідних параметрів дорівнює 2, а в НММ, призначеній для розпізнавання $\{Ka_b, Ka_l, Ka_p, Ka_r\}$, кількість вхідних параметрів дорівнює 41. Зазначимо, що при розпізнаванні Ka_s тип вхідних даних числовий.

При розпізнаванні U2R параметри №1 та №5-41 також мають числовий тип, а параметр №2-№4 – символьний.

Також по аналогії з системою розпізнавання ШПЗ та класифікації листів електронної пошти, до складу ШВ НММ входить тільки один нейрон. При цьому вихід НММ (Y) при класифікації кібератак дорівнює 1, а при класифікації безпечного стану дорівнює -1. В інших випадках класифікацію слід проводити відповідно виразу (5.25).

Визначення оптимального виду НММ. Особливістю наявної статистики, яка стосується прикладів функціонування веб-серверу, є відсутність в них даних, які відвідають моментам реалізації мережевих кібератак типу Ka_s. Тобто статистичні дані для формування навчальної вибірки НММ відсутні, хоча і достатні для розробки ШП.

Також зазначимо, що при розпізнаванні кібератак, що входять до складу $\{Ka_b, Ka_l, Ka_p, Ka_r\}$, необхідна кількість навчальних прикладів

визначається виразом:

$$P \geq 20 * N_{вх} = 20 * 41 = 82. \quad (5.33)$$

При цьому кількість записів бази даних KDD-99, що відповідають очікуваним кібератакам, наступна: buffer_overflow – 33, loadmodule – 9, perl – 3, rootkit – 10. Таким чином, відповідно розробленого методу, визначення часових обмежень, кількість навчальних даних не достатня для побудови всіх HMM, окрім MPNN.

Тому для розпізнавання всієї множини ***Ка*** використано HMM типу MPNN, що пристосовані до навчання за допомогою експертних даних у вигляді продукційних правил. Мережа $MPNN_{шт}$ використана для розпізнавання Ka_s, а мережа $MPNN_{U2R}$ використана для розпізнавання $\{Ka_b, Ka_l, Ka_p, Ka_r\}$.

Застосування експертних даних для розпізнавання. Для розпізнавання мережевих кібератак типу шторм запитів розроблено 10 продукційних правил. Із них 5 правил відповідає безпечному стану, а 5 – реалізації атаки. Приклади правил наступні.

Приклад 1.

Опис правила: *Якщо в будь-який момент часу реальна кількість звернень перевищує ШНП більш ніж в 1,5 рази, то відбувається кібератака.*

Формалізований запис: *Якщо* $n_1 > 1{,}5 \wedge n_2 \in [0{,}24] \Rightarrow Y = 1$

Приклад 2.

Опис правила: *Якщо в будь-який момент часу реальна кількість звернень знаходиться в межах від 1,25 до 1,4 від ШНП, то ймовірність кібератаки 0,7.*

Формалізований запис: *Якщо* $n_1 \in [1.25{,}1.4] \wedge n_2 \in [0{,}24] \Rightarrow Y = 0.7$

Приклад 3.

Опис правила: *Якщо в момент часу, що кратний інтервалу [4,8], реальна кількість звернень знаходиться в межах від 1,2 до 1,3 від ШНП, то ймовірність кібератаки 0,8.*

Формалізований запис: *Якщо* $n_1 \in [1.1,1.3] \wedge n_2 \in [4,8] \Rightarrow Y = 0.8$.

Приклад 4.

Опис правила: *Якщо в будь-який момент часу реальна кількість звернень знаходиться в межах від 0,8 до 1,1 від ШНП, то ймовірність кібератаки 0,1.*

Формалізований запис: *Якщо* $n_1 \in [0.8,1.1] \wedge n_2 \in [0,24] \Rightarrow Y = 0.1$.

Для розпізнавання кібератак типу U2R розроблено 28 продукційних правил. 14 правил відповідають відсутності кібератак, 4 правила стосуються розпізнавання buffer_overflow, 4 правила – loadmodule, 3 правила – perl і 3 правила – rootkit.

Приклад правила визначення нормального функціонування:

Якщо тривалість з'єднання (duration) = 0 ∧ протокол (protocol_type) – tcp ∧ сервіс (service)– http ∧ flag – SF ∧ кількість отриманих байт (src_bytes) – від 0 до 400 ∧ кількість переданих байт (dst_bytes) – від 100 до 5000 ∧ land – 0 ∧ wrong_fragment – 0 і urgent – 0 і 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 ∧і 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 і dst_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 правила розпізнавання loadmodule.

Якщо тривалість з'єднання (duration) = від 70 до 110 ∧ протокол (protocol_type) – tcp ∧ сервіс (service)– telnet ∧ flag – SF ∧ кількість отриманих байт (src_bytes) – від 250 до 350 ∧ кількість переданих байт (dst_bytes) – від 1000 до 9000 ∧ land – 0 ∧ wrong_fragment – 0 ∧ urgent – 0 ∧ hot – 2 ∧ num_failed_logins – 0 ∧ logged_in = 1 ∧ num_compromised = від 1 до 4 ∧ root_shell – від 0 до 1 ∧ su_attempted = 0 ∧ num_root = від 3 до 4 ∧ num_file_creations – від 2 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 = 1 ∧ count = 0 ∧ srv_count = 0 ∧ serror_rate = 1 ∧ srv_serror_rate = 10 ∧ rerror_rate = 1 ∧ srv_rerror_rate = 0 ∧ same_srv_rate = 0.3 ∧ diff_srv_rate = 0 ∧ srv_diff_host_rate = 0 ∧ dst_host_count = від 1 до 84 ∧ dst_host_srv_count = 10 ∧ dst_host_same_srv_rate = 1 ∧ dst_host_diff_srv_rate = 0.00 ∧ dst_host_same_src_port_rate = 1 ∧ dst_host_srv_diff_host_rate = від 0 до 0.3 ∧ dst_host_serror_rate = 0 ∧ dst_host_srv_serror_rate = 0 ∧ dst_host_rerror_rate = 0 ∧ dst_host_srv_rerror_rate = від 0 до 0.1.

Приклад правила для розпізнавання buffer_overflow наведено в п. 4.1.

Розробка НМС. Проведені дослідження дозволили розробити показану на рис. 5.5 структуру НМС розпізнавання мережевих кібератак.

Основними частинами структури являються:

– ПАВД – підсистема аналізу вхідних даних, за рахунок якої формується множина вхідних параметрів НММ та проводиться накопичення статистичних даних;

– ПЕО – підсистема експертної оцінки, в якій на основі експертих даних формуються продукційні правила, що характеризують стан захищеності;

– ПРС – підсистема розпізнавання та сигналізації, за рахунок якої

реалізується розпізнавання кібератак та відбувається сигналізація про стан захищеності;

– МУС – модуль управління системою, що служить для переведення системи в наступні два режими функціонування:

– РН – навчання НММ;

– РРК – розпізнавання кібератак.

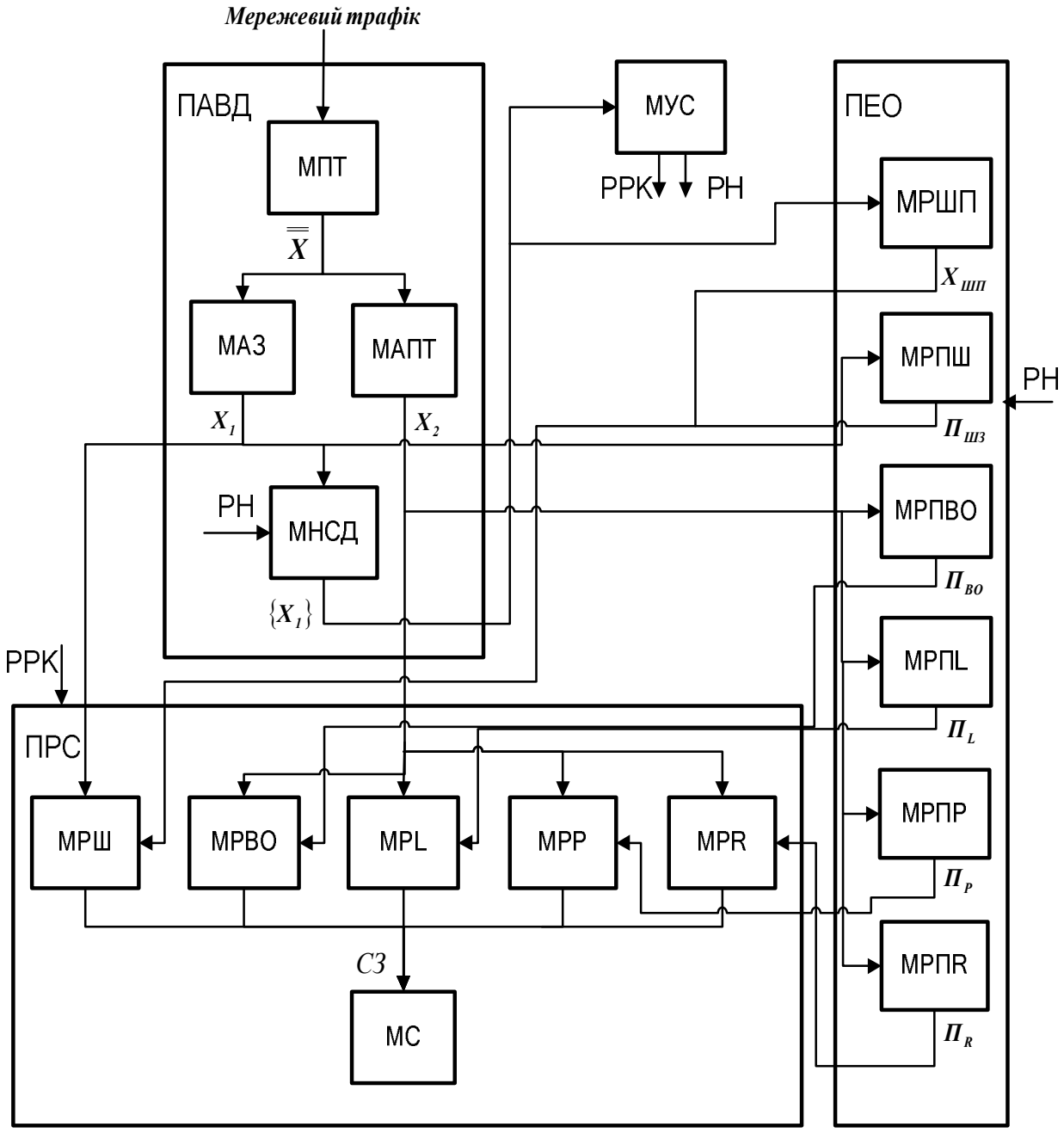


Рис. 5.5. Структура системи розпізнавання мережевих кібератак

Призначення окремих модулів розробленої НМС, що входять до складу означених підсистем, наведено в табл. 5.18.

Таблиця 5.18

Склад нейромережевої системи розпізнавання мережевих кібератак

Назва підсистеми	Назва модулю	Призначення модулю
ПАВД	МПТ	Перехоплення вхідного трафіку Веб-серверу
	МАЗ	Аналізу трафіку для визначення кількості запитів за одну секунду (X_1)
	МАПТ	Аналізу трафіку для визначення множини параметрів мережевих запитів (X_2)
	МНСД	Накопичення множини статистичних даних для формування ШНП ($\{X_1\}$)
ПЕО	МРШП	Розробка ШНП ($\boldsymbol{X}_{ШП}$)
	МРПШ, МРПВО, МРПL, МРПР, МРПR	Розробкапродукційних правил для розпізнавання кібератак виду шторм запитів ($П_{ШТ}$), buffer_overflow ($П_{ВО}$), loadmodule ($П_L$), perl ($П_P$), rootkit ($П_R$).
ПРС	МРШ, МРВО, MPL, MPP, MPR	Розпізнавання кібератак виду шторм запитів, buffer_overflow, loadmodule, perl, rootkit.
	МС	Сигналізації

Експериментальні дослідження. На першому етапі було розроблено марківську модель ШНП кількості звернень до веб-серверу за 1с. Для розробки використано метод, наведний в п. 4.3. Статистичні дані зібрані на протязі 2012 р. в процесі експлуатації веб-серверу однієї із комерційних

установ країн СНД. Із статистики відфільтровані дані, що відповідають атакам, зафіксованим системою визначення атак.

Для прикладу, на рис. 3.4 показано графік зміни кількості звернень на протязі однієї доби при вікні спостережень одна година.

Проведений, відповідно (4.44, 4.46-4.51), аналіз показав відсутність тренду та наявність періодичних складових. Побудована показана на рис. 5.7 періодограма кількості звернень до Веб-серверу.

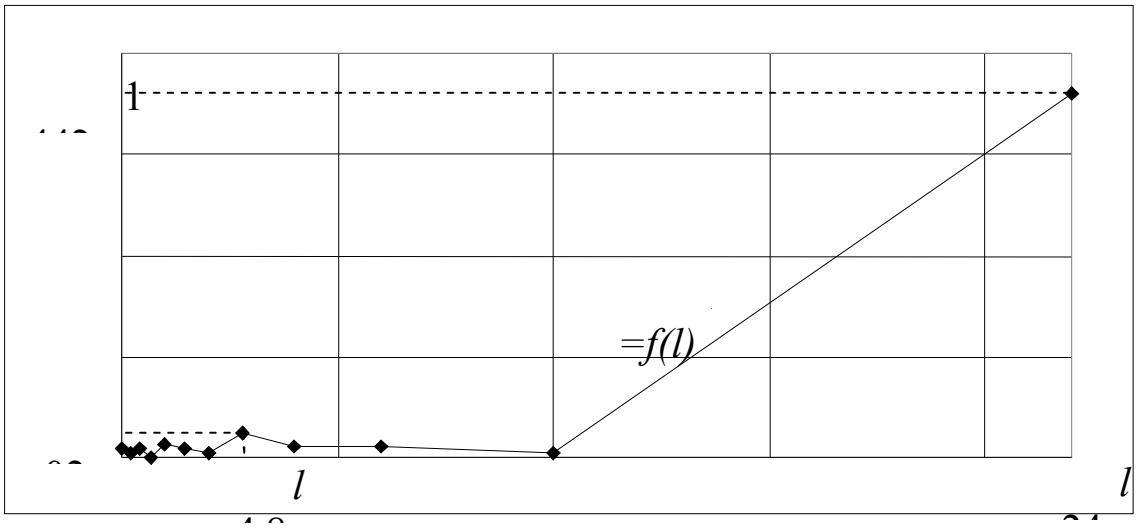


Рис. 5.7. Періодограма кількості звернень до Веб-серверу

На періодограмі простежується два максимуми I_1=102 та I_2=1442, яким відповідають періоди l_1= 4,8 год. та l_2=24 год. Тому статистичні дані розглядались як двохперіодичний ряд даних з періодами 4,8 год. та 24 год. Для розрахунку ймовірностей переходів застосовано вирази (4.52, 4.53) за умови, що X_{max}=30, X_{min}=0, кількість станів N=10, а одна година функціонування Веб-серверу дорівнює 100 крокам процесу.

Для моделювання двохперіодичного ряду створено марківську модель M^{Σ}_{BAB}, яка складається із двох модулів (марківських моделей) M_1 та M_2.

Модуль M_1 призначений для моделювання процесу з періодом l_1=24 год., а модуль M_2 – для моделювання процесу з періодом l_2=4,8 год. В межах

періоду l_1 визначено дві нестаціонарні точки: максимуму $A^{(24)}$=12 год. та мінімуму $B^{(24)}$=0 год. В зв'язку з тривалим терміном (l_1/2=12 год.) та різкими змінами ПБ моделювання на півперіоді проводилось за допомогою чотирьох ЛМ. Для цього кожен півперіод був розділений на два однакових відрізки, кожному із яких відповідав власний ЛМ. Таким чином M_1 складалась із чотирьох ергодичних стаціонарних ЛМ – $L^{(1)}_{1,1}$, $L^{(1)}_{1,2}$, $L^{(1)}_{2,1}$ та $L^{(1)}_{2,2}$. $L^{(1)}_{1,1}$, $L^{(1)}_{1,1}$, призначений для моделювання на інтервалах, кратних [0, 6], , $L^{(1)}_{1,2}$ – на інтервалах, кратних [6, 12], $L^{(2)}_{2,1}$ – на інтервалах кратних [12, 18], а $L^{(2)}_{2,2}$ – на інтервалах, кратних [18, 24].

Також при побудові ЛМ прийнято, що можливі переходи тільки між трьома сусідніми станами. Результати моделювання ШП за допомогою M_1 показані на рис. 5.8 у вигляді графіків математичного сподівання та розподілу ймовірності.

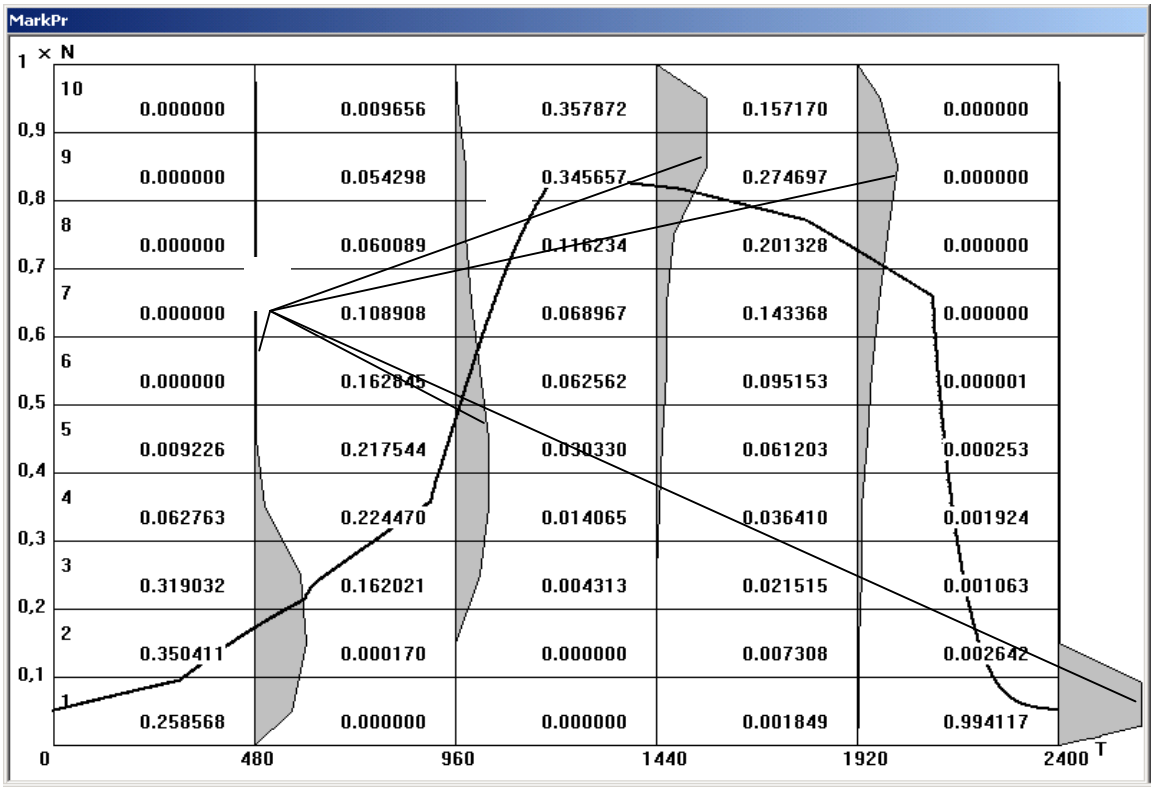


Рис. 5.8. Параметри марківської моделі M_1 з періодом 24 год.

На рис. 5.8 символом А позначено графік математичного сподівання, а символом В – графіки ймовірності розподілу по станам ЛМ.

Модель M_2 складалась із двох ергодичних стаціонарних ЛМ – $L^{(2)}{}_1$ та $L^{(2)}{}_2$. ЛМ $L^{(2)}{}_1$ призначений для моделювання на інтервалах, кратних [0, 2.4], а $L^{(2)}{}_2$ – на інтервалах, кратних [2.4, 4.8].

Результуючий графік математичного сподівання двохперіодичної моделі M^{Σ}_{BAB} показано на рис. 3.4. При цьому середня відносна похибка математичного сподівання становить ≈7 %, що в 1,5-2 рази менше, ніж похибка розповсюджених моделей ШП Веб-серверу [115, 212].

Розробка та апробація НММ. На другому етапі досліджень з використанням методу, наведеного в п. 4.1, розроблено НММ типу MPNN та проведено їх апробацію.

$MPNN_{шт,}$ призначена для розпізнавання шторму запитів, має наступні параметри:

– Кількість вхідних параметрів – $N_x = 2$.

– Кількість вихідних параметрів – N_Y=1.

– Кількість нейронів ШД – 2. Нейрон А відповідає безпечному стану, нейрон В – реалізації кібератак.

– Кількість нейронів ШО також дорівнює 10.

– Кількість нейронів ШД дорівнює 20.

$MPNN_{U2R}$, призначена для розпізнавання кібератак типу U2R, має наступні параметри:

– Кількість вхідних параметрів – $N_x = 41$.

– Кількість вихідних параметрів – N_Y=1.

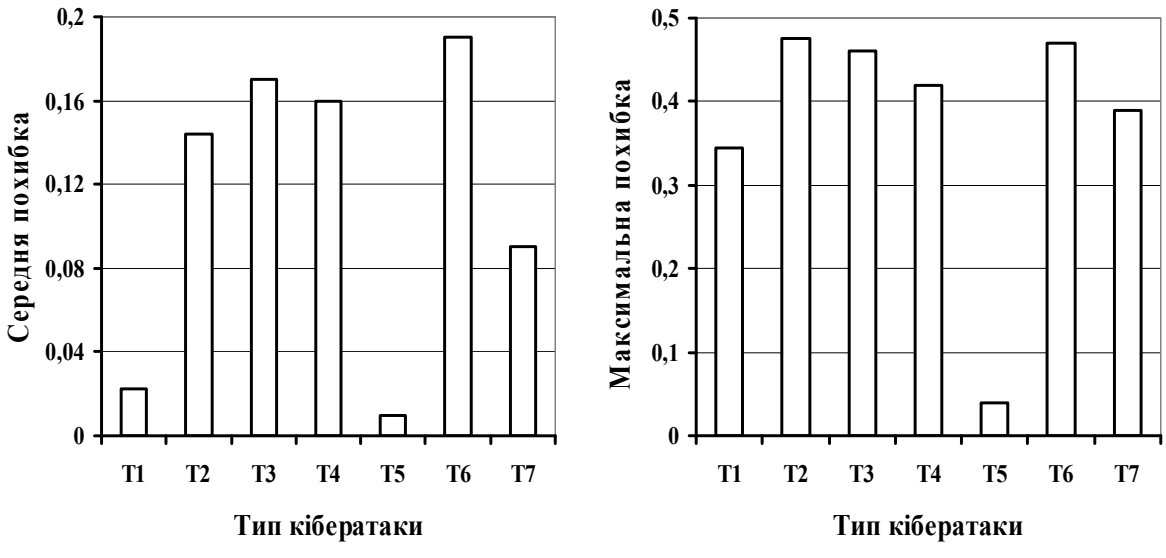
– Кількість нейронів ШД – 5. Нейрон А відповідає безпечному стану, нейрон В – реалізації кібератаки buffer_overflow, нейрон С – loadmodule, D – perl і E – rootkit.

– Кількість нейронів ШО також дорівнює 28.

– Кількість нейронів ШФ дорівнює 1148.

Структура розроблених мереж відповідає рис. 3.10 з урахуванням наведених величин їх параметрів.

Апробація $MPNN_{шт}$ на статистичних даних, що були використані для створення марківської моделі ШП, дозволила виявити реалізовані мережеві кібератаки типу шторм запитів, що підтверджується результатами ретроспективних досліджень функціонування веб-сервера. При цьому середня похибка моделі знаходиться в межах 10%, що на 5-10% краще, ніж у подібних моделях. Апробація розробленої моделі $MPNN_{U2R}$ на даних KDD-99 показала абсолютну точність розпізнавання всіх видів атак класу U2R. Для порівняння отриманих результатів використано роботи [59, 212], в яких для розпізнавання зазначених кібератак використовувались БШП, ТК та спеціальна адаптивна HMM. Точність розпізнавання класу U2R атак ТК становить: для buffer_overflow – 0.0458, для loadmodule – 0.0208, для perl – 0.2857, а для rootkit – 0.0063. При цьому БШП, по причині малого обсягу навчальних даних, взагалі не вдалось навчити розпізнавати жодну з кібератак типу U2R. Точність розпізнавання адаптивної HMM не перевищує 0,5. Таким чином, точність розробленої НМС перевищує відомі аналоги. В підсумку основні характеристики розроблених НМС та гістограми похибок їх вихідних сигналів наведені в табл. 5.18 та рис. 5.9.



а *б*

Рис. 5.9. Гістограми похибок вихідного сигналу розроблених нейромережевих систем

Таблиця 5.18

Характеристики розроблених НМС

НМС	Тип кібератаки	Використані ПБ	Вид НММ
Розпізнавання ШПЗ та класифікації електронних листів	Т1 – наявність шкідливого скрипта JavaScript	Потенційно небезпечні функції JavaScript	ДШП
	Т2 – наявність шкідливого скрипта VBScript	Потенційно небезпечні функції VBScript	
	Т3 – поведінка ШПЗ	Потенційно небезпечні функції API Windows	
	Т4 – спам	Частоти зустрічі в тексті інформативних слів	ТК
	Т5 – витік		
Розпізнавання мережевих кібератак	Т6 –мережева кібератака класу U2R	Параметри мережевих з’єднань	MPNN
	Т7 – шторм запитів	Кількість запитів за 1 с	

5.5. Висновки до розділу

В даному розділі вирішувалась науково-практична задача розробки нейромережевих систем оцінювання параметрів безпеки Інтернет-орієнтованих інформаційних систем. В процесі вирішення отримано наступні наукові результати:

– Вперше розроблено комплексну методологію нейромережевої оцінки параметрів безпеки, яка за рахунок взаємопов’язаного використання

розроблених підходів до верифікації нейромережевих засобів, визначення оптимального виду нейромережевої моделі, розроблених моделей створення ефективних нейромережевих засобів оцінки параметрів безпеки, інтеграції параметрів безпеки та методів подання експертних знань, проектування шаблонів поведінки, визначення часових характеристик використання та визначення ефективності розробки нейромережевих засобів забезпечує можливість їх верифікації, дозволяє розширити функціональні можливості та, відповідно до розробленого інтегрального критерію, обрати найбільш ефективний нейромережевий засіб. Шляхом порівняльних розрахунків визначено, що використання запропонованої методології дозволяє до 4 разів підвищити генеральний критерій ефективності розробки нейромережевих засобів по відношенню до подібних найдосконаліших методів.

– На основі комплексної методології розроблено структуру нейромережевої системи оцінки параметрів безпеки для розпізнавання кібератак, яка за рахунок використання модулів класифікації параметрів кібератак, формування статистично подібних кібератак, формування параметрів розробленої марківської моделі шаблону поведінки, підсистеми первинного визначення параметрів кібератак, модулів інтеграції параметрів безпеки, визначення обчислювальних обмежень, розрахунку критеріїв оптимізації виду нейромережевої моделі та показників ефективності, формування продукційних правил підсистеми експертного оцінювання параметрів нейромережевих засобів, модулів розробки MPNN, визначення доцільності застосування, оптимізації виду та верифікації нейромережевих моделей підсистеми розробки нейромережевих моделей забезпечує верифікацію отриманих результатів та підвищення ефективності інструментальних засобів розпізнавання кібератак завдяки зменшенню похибок класифікації кібератак, оперативній адаптації до умов застосування та нових типів кібератак.

– З використанням комплексної методології та структурних рішень системи оцінки параметрів безпеки розроблено системи та відповідне

програмне забезпечення для розпізнавання веб-орієртованого шкідливого програмного забезпечення, класифікації листів електронної пошти та розпізнавання мережевих кібератак. При розпізнаванні веб-орієртованого шкідливого програмного в якості параметрів безпеки використовуються визначені потенційно небезпечні оператори скриптових мов програмування та виклики потенційно небезпечних функцій операційної системи. Параметрами безпеки у випадку класифікації листів електронної пошти для виявлення спаму та витоків текстової інформації являються частоти зустрічі в тексті листа інформативних слів в канонічній формі. При розпізнаванні мережевих кібератак параметрами безпеки послужили кількість запитів до мережевих ресурсів за одну секунду та параметри мережевих запитів. Визначено, що найбільш ефективними видами нейромережевих моделей для розпізнавання шкідливого програмного забезпечення є багатошаровий персептрон, для класифікації листів електронної пошти – карта Кохонена, а для розпізнавання мережевих кібератак – мережа MPNN, пристосована для навчання за допомогою експертних даних. Для розпізнавання мережевої кібератаки на веб-сервер типу шторм запитів використано розроблену марківську модель шаблону нормальної поведінки, що дозволило адаптувати систему розпізнавання до складного типового характеру кількості мережевих запитів. Експериментальне дослідження розроблених систем показало можливість забезпечення зменшення середньої похибки розпізнавання відомих кібератак на 5-10% по відношенню до похибок аналогічних систем. Крім того, експерименти підтвердили адекватність розроблених систем щодо можливості оперативно розпізнавати нові види кібератак при обмежених обчислювальних ресурсах та варіативності умов застосування.

Основна наукова новизна результатів полягає в вперше розробленій комплексній методології нейромережевої оцінки параметрів безпеки та створених на її основі структурних рішеннях нейромережевих систем.

ВИСНОВКИ

У дисертації запропоноване нове вирішення актуальної науково-прикладної проблеми, що полягає у створенні комплексної методології розробки широкодоступних ефективних нейромережевих засобів оцінювання параметрів безпеки Інтернет-орієнтованих інформаційних систем, які за рахунок теоретично обґрунтованого вибору характеристик дозволяють оперативно розпізнавати нові види кібератак при обмежених обчислювальних ресурсах та варіативності умов застосування. Проведені дослідження дозволяють зробити наступні висновки:

1. Визначено, що недоліки сучасних нейромережевих засобів оцінки параметрів безпеки Інтернет-орієнтованих інформаційних систем для розпізнавання кібератак спричинені недосконалістю теоретико-методологічних підходів до розробки нейромережевих систем оцінювання параметрів безпеки, які не в повній мірі адаптовані до умов застосування та нових типів кібратак. Обґрунтовано перспективність створення комплексної методології нейромережевої оцінки параметрів безпеки, для розробки якої необхідно розвинути теоретичні положення, моделі та методи побудови нейромережевих засобів.

2. Отримали подальший розвиток теоретичні положення побудови нейромережевих засобів оцінювання параметрів безпеки, які за рахунок вперше розроблених підходів до розпізнавання поступових та неочікуваних кібератак, визначення оптимального виду нейромережевої моделі, доцільності застосування та ефективності розробки нейромережевих засобів, класифікації подібних кібератак, застосування продукційних правил для подання експертних знань, верифікації нейромережевих моделей, запропонованих критеріїв оцінки ефективності нейромережевих засобів, критеріїв вибору оптимального виду нейромережевої моделі та застосуванню розробленого функціоналу приведеної помилки навчання багатошарового персептрону дозволяють вдосконалювати нейромережеві засоби шляхом їх

адаптації до поступових і неочікуваних кібератак, умов використання, навчання за допомогою експертних даних, зменшувати похибки класифікації та надають можливість верифікації отриманих рішень.

3. Отримали подальший розвиток моделі нейромережевих засобів оцінки параметрів безпеки, які за рахунок застосування розроблених теоретичних положень побудови нейромережевих засобів**,** експертного оцінювання вагомості параметрів безпеки, введення в модель MPNN нейронного шару фільтрації з лінійною біполярною з насиченням функцією активації, розроблених аналітичних залежностей для розрахунку параметрів ланцюгів Маркова, призначених для прогнозування параметрів безпеки на стаціонарних інтервалах, та для оцінки оптимальної кількості схованих нейронів, кількості обчислювальних навчальних операцій, обсягу пам’яті і помилки навчання багатошарового персептрону дозволяють: визначити перелік параметрів безпеки, які доцільно оцінювати нейромережевими засобами; створювати шаблони поведінки, адаптовані до складного характеру параметрів безпеки; в 1,5-6 разів зменшити ресурсоємність процесу визначення оптимальної структури багатошарового персептрону; апріорно оцінювати обчислювальні ресурси, необхідні для реалізації нейромережевої моделі; за допомогою експертних даних навчати нейромережеву модель; формалізувати процес створення ефективних нейромережевих засобів, що є основою для підвищення ефективності методів їх розробки.

4. Вперше розроблено метод подання експертних знань для нейромережевих засобів оцінки параметрів безпеки, що за рахунок розробленого математичного забезпечення детермінування параметрів подібних кібератак, продукційних правил представлення навчальних прикладів та структури і вагових коефіцієнтів синаптичних зв’язків нейромережевої моделі типу MPNN дозволяє забезпечити оперативність розпізнавання та розширити множину типів кібератак, характеристики яких не представлені в статистичних даних. Апробація методу на сигнатурах

кібератак, представлених в базі даних KDD-99, показала абсолютну повноту класифікації кібератак типу U2R, що в 5 разів перевищує результати інших відомих нейромережевих методів.

5. Вперше розроблено метод визначення часових характеристик використання нейромережевих засобів, в якому завдяки використанню розроблених аналітичних залежностей для визначення очікуваного терміну їх розробки, допустимих термінів формування навчальної вибірки та навчання нейромережевої моделі, запропонованих співвідношень між очікуваним і допустимим терміном розробки та очікуваним і допустимим терміном навчання, розробленій множині допустимих видів нейромережевих моделей отримана можливість визначення доцільності застосування нейромережевих засобів оцінки параметрів безпеки для виявлення очікуваних кібератак на заданий об’єкт захисту. Доведена можливість застосування нейромережевих засобів для розпізнавання типових Інтернет-орієнтованих кібератак: сканування портів, Dos-атак, IP-спуфінгу та веб-орієнтованих скриптових вірусів та троянів.

6. Вперше розроблено метод проектування шаблону поведінки, який використовується для навчання нейромережевих моделей, в якому за рахунок застосування багатоперіодичних рядів динаміки, розробленого математичного забезпечення для розрахунку періодичних складових та розробленої негомогенної марківської моделі забезпечується зменшення похибки шаблону в 1,5-2, що є основою для зменшення терміну формування навчальної вибірки та зменшення похибок класифікації нейромережевих моделей при розпізнаванні поступових кібератак.

7. Вперше розроблено метод визначення ефективності розробки нейромережевих засобів оцінки параметрів безпеки, який за рахунок застосування запропонованих критеріїв оцінки ефективності, що відображають ступінь виконання основних вимог до побудови та застосування нейромережевих засобів, запропонованих вагових коефіцієнтів критеріїв ефективності та розробленого інтегрального критерію ефективності

нейромережевих засобів дозволяє, відповідно до визначених критеріїв, обрати найбільш ефективний засіб. Застосування методу дозволило визначити, що типовими недоліками більшості відомих нейромережевих засобів є недостатня обґрунтованість доцільності використання, низька пристосованість до застосування всієї множини перспективних нейромережевих моделей, неможливість використання експертних даних та емпіричний вибір виду нейромережевої моделі.

8. Вперше розроблено комплексну методологію нейромережевої оцінки параметрів безпеки, яка за рахунок взаємопов'язаного використання розроблених підходів до верифікації нейромережевих засобів, визначення оптимального виду нейромережевої моделі, розроблених моделей створення ефективних нейромережевих засобів оцінки параметрів безпеки, інтеграції параметрів безпеки та методів подання експертних знань, проектування шаблонів поведінки, доцільності застосування та визначення ефективності розробки нейромережевих засобів, що забезпечує можливість їх верифікації, дозволяє розширити функціональні можливості та, відповідно до розробленого генерального критерію, обрати найбільш ефективний нейромережевий засіб. Використання запропонованої методології дозволяє до 4 разів підвищити величину генерального критерію ефективності розробки нейромережевих засобів.

9. На основі комплексної методології розроблено структуру нейромережевої системи оцінки параметрів безпеки для розпізнавання кібератак, яка за рахунок використання модулів класифікації параметрів кібератак, формування подібних кібератак, формування параметрів розробленої марківської моделі шаблону поведінки, підсистеми первинного визначення параметрів кібератак, модулів інтеграції параметрів безпеки, визначення обчислювальних обмежень, розрахунку критеріїв оптимізації виду нейромережевої моделі та показників ефективності, формування продукційних правил підсистеми експертного оцінювання параметрів нейромережевих засобів, модулів розробки MPNN, визначення доцільності

застосування, оптимізації виду та верифікації нейромережевих моделей підсистеми розробки нейромережевих моделей забезпечує верифікацію отриманих результатів та підвищення ефективності інструментальних засобів розпізнавання кібератак завдяки зменшенню похибок класифікації кібератак, оперативній адаптації до умов застосування та нових типів кібератак.

10. З використанням комплексної методології та структурних рішень системи оцінки параметрів безпеки розроблено системи та відповідне програмне забезпечення для розпізнавання шкідливого програмного забезпечення, класифікації листів електронної пошти та розпізнавання мережевих кібератак. Експериментальне дослідження розроблених систем підтвердили їх адекватність щодо можливості забезпечення оперативного розпізнавання нових видів кібератак при обмежених обчислювальних ресурсах та варіативності умов застосування.

11. Зазначені результати впроваджені у діяльність Інституту проблем моделювання в енергетиці ім. Г.Є. Пухова НАН України, Національного авіаційного університету, Національного технічного університету України «КПІ», Національного університету будівництва і архітектури, що підтверджено відповідним актами впровадження, які містяться у додатках до дисертаційної роботи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Абрамов Е. С. Разработка и исследование методов построения систем обнаружения атак: дис. … канд. техн. наук: 05.13.19 / Абрамов Е. С. – Таганрог, 2005. – 199 с.

2. Айвенс К. Компьютерные сети / Айвенс К. ; пер. с. англ. – СПб. : Питер, 2006. – 304 с.

3. Андерсон Т. Статистический анализ временных рядов / Андерсон Т. ; пер. с. англ. И. Г. Журбенко. – М. : МИР, 1976. – 757 с.

4. Артеменко А.В., Головко В. А. Анализ нейросетевых методов распознавания компьютерных вирусов /Материалы секционных заседаний. Молодежный инновационный форум «ИНТРИ» – 2010. — Минск: ГУ «БелИСА», 2010. – С. 47-48.

5. Архипов А. Применение моделей обнаружения аномалий для выявления атак / А. Архипов, А. Ишутин // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні : наук.-техн. конф., 1-3 берез. 2006 р. : тези доп. – К., 2006. – С. 71-72.

6. Барский А. Б. Нейронные сети: распознавание, управление, принятие решений / А. Б. Барский. – М. : Финансы и статистика, 2004. – 176 с.

7. Безобразов С.В. Алгоритмы исскуственных имунных систем и нейронних сетей для обнаружения вредоносных программ / С.В.Безобразов, В.А.Головко//Нейроинформатика. – 2010. – №7. – С. 273-288.

8. Безруков Н. Н. Компьютерная вирусология / Н. Н. Безруков. – К. : Инкомбук, 1990. – 450 с.

9. Беллман Р. Процессы регулирования с адаптацией / Беллман Р. ; пер. с. англ. – М. : Наука, 1964. – 364 с.

10. Беляев А. Системы обнаружения аномалий: новые идеи в защите информации / А. Беляев, С. Петренко // Экспресс-Электроника. – 2004. – № 2. – С. 12–14.

11. Берзон В. Е. Об одном подходе к проблеме автоматического реферирования и автоматического свертывания индексируемых текстов / В. Е. Берзон // НТИ. Сер.2.– 1971, № 10.– С.16-21.

12. Блюменау Д. И. Экстрагирование как один из подходов к автоматизации реферирования / Д. И. Блюменау, И. С. Добронравов, Д. Г. Лахути // НТИ. Сер.2. – 1982. – № 2. – С. 108–128.

13. Богуш В.М. Інформаційна безпека: термінологічний навчальний довідник / В. М. Богуш, В. Г. Кривуца , А. М. Кудін. – К. : ДВК, 2004. – 508 с.

14. Бокс Дж. Анализ временных рядов, прогноз и управление / Бокс Дж., Дженкинс Г. М.; пер. с англ. – М. : Радио и свіязь, 1969. – 408 с.

15. Большев А. К. Алгоритмы преобразования и классификации трафика для обнаружения вторжений в компьютерные сети: авторефер. дисс. на соискание научн. степени канд. техн. наук : спец. 05.13.19 – Методы и системы защиты информации, информационная безопасность / А. К. Большев – Санкт-Петербург, 2011. – 36 с.

16. Браіловський М. М. Захист інформації у банківській діяльності / М. М. Браіловський, Г. П. Лазарєв, В. О. Хорошко. – К. : ПоліграфКонсалтинг, 2004. – 216 с.

17. Бриллинджер Д. Р. Временные ряды. Обработка данных и теорія / Бриллинджер Д. Р. ; пер. с англ. А. В. Булинского. – М. : Мир, 1980. – 536 с.

18. Бурячок В.Л. Кібернетична безпека – головний фактор сталого розвитку сучасного інформаційного суспільства // Сучасна спеціальна техніка. – 2011. – № 3 (26). – С. 104–114.

19. Васильев В.И. Нейронные сети при обнаружении атак в сети Internet (на примере атаки SYNFLOOD) / В.И. Васильев, А.Ф. Хафизов // Нейрокомпьютеры в информационных и экспертных системах. – М.: Радиотехника, 2007. – №6. – С. 34-38.

20. Вапник В. Н. Восстановление зависимостей по эмпирическим данням / В. Н. Вапник. – М. : Наука, 1979. – 448 с.

21. Вентцель Е. С. Элементы теории игр / Е. С. Вентцель. – М. : Гос. изд. физ.-мат. лит., 1961. – 68 с.

22. Вентцель Е. С. Исследование операций / Е. С. Вентцель. – М. : Сов. радио, 1972. – 552 с.

23. Вентцель Е.С. Теория вероятностей / Е. С. Вентцель, Л. А. Овчаров. – М. : Наука,1976. – 378 с.

24. Вилков А.С. Информационная безопасность персональных ЭВМ и мониторинг компьютерных сетей / А.С. Вилков. – М. : МИНИТ ФСБ России, 2005. – 210 с.

25. Волосов К. А. Методика анализа эволюционных систем с распределенными параметрами специальность: дис. … докт. техн. наук: 05.13.01 / Волосов К. А. – М., 2007. – 264 с.

26. Вороновский Г. К. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности / Г. К. Вороновский, К. В. Махотило, С. А. Сергеев. – Харьков: Основа, 1997. – 112 с.

27. Галушкин А. И. Теория нейронных сетей / А. И. Галушкин. – М. : ИПРЖР, 2000. – 416 с.

28. Гареев А. Ф. Применение вероятностной нейронной сети для автоматического рубрицирования текстов / А. Ф. Гареев // Нейроинформатика-99 : науч. конф., 20-22 января 1999 г. : тезисы докл. – М., 1999. – С. 71–79.

29. Гарнаев А. Ю. Microsoft Office 2000 / А. Ю. Гарнаев. – СПб. : БХВ, 2000. – 656 с.

30. Глибовець М. М. Штучний інтелект / М. М. Глибовець, О. В. Олецький. – К. : Києво-Могилян. акад., 2002. – 366 с.

31. Гнатюк С. Кібертероризм: історія розвитку, сучасні тенденції та контрзаходи / С. Гнатюк // Безпека інформації. – 2013. – Том 9, №2. – С. 118 – 129.

32. Головко В. А. Нейронные сети: обучение, организация и применение / В. А. Головко. – М. : ИПРЖР, 2001. – 256 с.

33. Горбань А. Н. Нейронные сети на персональном компьютере / А. Н. Горбань, Д. А. Россиев. – Новосибирск : Наука, 1996. – 276 с.

34. Горбань А. Н. Обучение нейронных сетей / А. Н. Горбань. – М. : ParaGraph, 1990. – 160 с.

35. Горбань А. Н. Визуализация данных методом упругих карт / А. Н. Горбань, А. Ю. Зиновьев, А. А. Питенко // Информационные технологии, – 2000. – № 6. – С. 26–35.

36. Грамматический словарь русского языка: Словоизменение / [сост. Зализняк А. А.] – М. : Рус. яз., 1980. – 880 с.

37. Гришин А.В. Нейросетевые технологии в задачах обнаружения компьютерных атак / А.В. Гришин // Информационные технологии и вычислительные системы – 2011. – №1. – С. 53 -64.

38. Данілов В. О. Розроблення процесу оптимальної модернізації телефонного електрозв'язку : автореф. дис. на здобуття наук. ступеня канд. техн. наук : спец 05.13.06 "Автоматизовані системи управління та прогресивні інформаційні технології" / В. О. Данілов. – К., 2003. – 19 с.

39. Джерри Х. Реестр Microsoft Windows XP / Джерри Х. ; пер. с англ. – М. : Эком, 2006. – 656 с.

40. Довлад О. А. Дослідження та розробка моделі процесу атаки та трафіку локальної мережі / О. А. Довлад // Захист інформації. – 2009. – № 1 – С. 83–86.

41. Додонов А. Г. Живучесть информационных систем / Додонов А. Г., Ландэ Д. В. // К. Наук. думка, 2011. – 256 с.

42. Дорогов А. Ю. Структурный синтез быстрых нейронных сетей / А. Ю. Дорогов // Нейрокомпьютер. – 1999. – № 1 – С. 11–24.

43. Дорогов А. Ю. Структурный синтез двухслойных быстрых нейронных сетей / А. Ю. Дорогов // Кибернетика и системный анализ. – 2000. – № 4. – С. 47–56.

44. Дорогов А. Ю. Быстрые нейронные сети / А. Ю. Дорогов, А. А. Алексеев // Пятьдесят лет развития кибернетики : междунар. научн.-техн. конф., 5-7 окт. 1999 г. : тезисы докл. – СПб., 1999. – С. 120–121.

45. Дорогов А. Ю. Категории ядерных нейронных сетей / А. Ю. Дорогов, А. А. Алексеев // Нейроинформатика-99 : науч. конф., 20-22 января 1999 г. : тезисы докл. – М., 1999. – С. 55–64.

46. Дударь З. В. Реализация нейронов в семантических нейронных сетях / З. В. Дударь, Д. Е. Шуклин // Радиоэлектроника и информатика. – Х., 2000. – № 4. – С. 89–96.

47. Дударь З. В. Семантическая нейронная сеть как формальный язык описания и обработки смысла текстов на естественном языке / З. В. Дударь, Д. Е. Шуклин // Радиоэлектроника и информатика. – Х., 2000. – № 3. – С. 72–76.

48. Дьяконов М.Ю. Нейросетевая система обнаружения аномального поведения вычислительных процессов микроядерных операционных систем: авторефер. дисс. на соискание научн. степени канд. техн. наук : спец. 05.13.19 – Методы и системы защиты информации, информационная безопасность / М. Ю. Дьяконов – Уфа, 2010. – 28 с.

49. Ежов А. А. Нейрокомпьютинг и его применения в экономике и бизнесе / А. А. Ежов, С. А. Шумский. – М. : МИФИ, 1998. – 224 с.

50. Ермаков А. Е. Поиск фактов в тексте / А. Е. Ермаков // Мир ПК. – 2005. – №2. – С. 64–66.

51. Ермаков А. Е. Проблемы полнотекстового поиска и их решение / А.Е. Ермаков // Мир ПК. – 2001. – №5. – С. 64–66.

52. Ермаков А. Е. Тематический анализ текста с выявлением сверхфразовой структуры /А. Е. Ермаков // Информационные технологии. – 2000. – №11. – С. 45–47.

53. Ермаков А. Е. Эксплицирование элементов смысла текста средствами синтаксического анализа-синтеза / А. Е. Ермаков // Компьютерная лингвистика и интеллектуальные технологи : междунар. научн. конф., 12-14 окт. 2003 г. : тезисы докл. – М., 2003. – С. 136–140.

54. Ермаков А. Е. Компьютерная морфология в контексте анализа связного текста / Ермаков А. Е., Плешко В. В. // Компьютерная лингвистика и интеллектуальные технологи : междунар. научн. конф., 16-18 окт. 2004 г. : тезисы докл. – М., 2004. – С. 185–190.

55. Ермаков А. Е. Компьютерный анализ текста при сборе информации к досье из открытых источников / Ермаков А. Е., Плешко В. В. // Конкурентная разведка в металлургии : междунар. научн. конф., 19-20 янв. 2005 г. : тезисы докл. – М., 2005. – С. 124–126.

56. Ермаков А. Е. Синтаксический разбор в системах статистического анализа текста / А. Е. Ермаков, В. В. Плешко // Информационные технологии. – 2002. – № 7. – С. 30–34.

57. Ермаков А. Е. Тематическая навигация в полнотекстовых базах данных / А. Е. Ермаков, В. В. Плешко // Мир ПК. – 2001. – № 8. – С. 52–55.

58. Емельянова Ю. Г. Анализ проблем и перспективы создания интеллектуальной системы обнаружения и предотвращения сетевых атак на облачные вычисления / Ю. Г. Емельянова, В. П. Фраленко // Программные системы: теория и приложения: электрон. научн. журн. – 2011. – № 4(8). – С. 17-31. [Электронный ресурс]. URL: http://psta.psiras.ru/read/psta2011_4_17-31.pdf.

59. 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 приложения. – 2011. – №3(7). – С. 3–15.

60. Жульков Е. Поиск уязвимостей в современных системах IDS / Е. Жульков // Открытые системы. – 2003. – № 7–8. – С. 16–18.

61. Заенцев И.В. Нейронные сети: основные модели / И. В. Заенцев. – Воронеж : Воронежский гос. ун-т, 1999. – 76 с.

62. Зайцев О. Нейросети в системах безопасности/О.Зайцев // IT-Спец. – 2007. – № 6. – С. 54–59.

63. Закер К. Компьютерные сети / Закер К.; : пер. с англ. – СПб. : БХВ-Петербург, 2000. – 1008 с.

64. Захарова М.В. Програмна модель процесу вибору ефективних механізмів захисту інформаційних ресурсів / М.В. Захарова, А.О. Корченко, І.В. Хропата // Захист інформації. – 2011. – №2 (51). – С. 129-134.

65. Замаруева И.В. Математические модели семантики свободных словосочетаний с родовидовыми компонентами и их применение в АИС : дис. ... канд. техн. наук : 05.13.23 / Замаруева Ирина Валерьевна. – Х., 1990. – 134 с.

66. Зима В. М. Безопасность глобальных сетевых технологий / В. М. Зима, А. А. Молдовян, Н. А. Молдовян. – СПб. : БХВ-Петербург, 2000. – 450 с.

67. Зиновьев А. Ю. Визуализация многомерных даннях / А. Ю. Зиновьев. – М. : СК Пресс, 2005. – 180 с.

68. Зиновьев А. Ю. Визуализаци данных методом упругих карт / А. Ю. Зиновьев, А. А. Питенко // Радіоелектроніка. Інформатика. Управління.– 2000. № 1. – С. 76–85.

69. Зуев А. В. Определение оптимальной совокупности контролируемых параметров при косвенном контроле средств защиты информации / А. В. Зуев , Ю. М. Хмелько // Защита информации : Сб. науч. трудов. – К. : НАУ, 2000. – С.70–75.

70. Иванов А.И. Быстрое обучение искусственных нейронных сетей в системах биометрической аутентификации личности : авторефер. дисс. на

соискание научн. степени док. техн. наук : спец. 05. 13. 01 "Управление в технических системах" / А. И. Иванов – Пенза, 2000. – 36 с.

71. Игнатов В. А. Элементы теории оптимального обслуживания технических изделий / В. А. Игнатов, Г. Г. Маньшин, В. В. Костановский. – Минск : Наука и техника, 1974. – 192 с.

72. Игнатов В. А. Статистическая оптимизация качества функционирования электронных систем / В. А. Игнатов, Г. Г. Маньшин, В. А. Трайнев. – М. : Энергия, 1974. – 264 с.

73. Ильницкий С. В. Работа сетевого сервера при самоподобной (self-similar) нагрузке / С. В. Ильницкий // Сб. научн. трудов Рижского техн. ун-та. – Рига : РТУ, 2004 – С. 80–94.

74. Информационная технология. Методы защиты .Менеджмент рисков информационной безопасности : BS ISO/IEC 27005:2008. – К. : 2011. – 70 с.

75. Каллан Р. Основные концепции нейронных сетей / Каллан Р. ; пер. с англ. А. Г. Сивака. – М. : Вильямс, 2003. – 288 с.

76. Карташов А. П. Построение сети нейроподобных элементов с ациклической активностью и экспоненциальным временем затухания / А. П. Карташов, Е. А. Карташова // Автоматика и телемеханика. – 1989. – № 2. – С. 147–157.

77. Касперски К. Техника и философия хакерских атак / К. Касперски. – М. : Солон, 2001. – 256 с.

78. Касперский Е.В. Компьютерные вирусы: что это такое и как с ними бороться / Е. В. Касперский. – М. : СК Пресс, 1998. – 288 с.

79. Кендалл М. Многомерный статистический анализ и временные ряды / Кендалл М., Стюарт А. ; пер. с англ. – М.: Наука, 1976. – 722 с.

80. Кирьянов Д.В. Mathcad 14 / Д. В. Кирьянов. – СПб. : БХВ-Петербург, 2007. – 704 с.

81. Китинг Д. Flash MX. Искусство создания web-сайтов / Китинг Д. ; пер. с англ. – К. : ТИД ДС, 2002. – 848 с.

82. Коваленко М.М. Комп'ютерні віруси і захист інформації / М. М. Коваленко. – К. : Наукова думка, 1999. – 268 с.

83. Кокс Д. Теория очередей / Кокс Д., Смит У. ; пер. с англ. – М. : Мир, 1966. – 452 с.

84. Колганов С. К. Построение в условиях дефицита информации сводных оценок сложных систем / С. К. Колганов, В. В. Корников, П. Г. Попов, Н. В. Хованов. – М. : Радио и связь, 1994. – 80 с.

85. Колисниченко Д.Н. Rootkits под Windows / Д. Н. Колисниченко. – СПб. : Наука и техника, 2006. – 320 с.

86. Комар М.П. Метод построения совокупного классификатора трафика информационно-телекоммуникационных сетей для иерархической классификации компьютерных атак / М.П.Комар // Системи обробки інформації.– 2012. – Випуск 3 (101), том 1 – С.134-138.

87. Комар М.П. Нейросетевой подход к обнаружению сетевых атак на компьютерные системы / М.П. Комар, И.О. Палий, Р.П. Шевчук, Т.Б. Федысив // Інформатика та математичні методи в моделюванні – 2011. – Том 1, №2. – С. 156-160.

88. Корченко А.А. Базовая модель параметров для построения систем выявления атак / А.И. Стасюк, А.А. Корченко // Захист інформації. – 2012. – № 2 (55). – С. 47-51.

89. Корченко А.А. Система выявления аномального состояния в компьютерных сетях / А.А. Корченко // Безпека інформації. – 2012. – № 2 (18). – С. 80-84.

90. Корченко А.О. Визначення коефіцієнтів важливості для експертного оцінювання у галузі інформаційної безпеки / Д.А. Горніцька, В.В. Волянська, А.О. Корченко // Захист інформації. – 2012. – №1 (54). – С. 108-121.

91. Корченко О. Г. Верифікація нейромережевих методів розпізнавання кібератак / О. Г. Корченко, І. А. Терейковський, С. В. Казмірчук // Науково-технічний збірник «Управління розвитком складних систем» Київського

національного університету будівництва і архітектури. – 2014. – Випуск 17. – С. 168-172.

92. Корченко О. Г. Метод оцінки нейромережевих засобів щодо можливостей виявлення інтернет-орієнтованих кібератак / О.Г. Корченко, І.А. Терейковський, С.В. Казимірчук // Вісник інженерної академії наук. – 2014. – Вип. 2. – С. 87-93.

93. Корченко О.Г. Системи захисту інформації : Монографія / О.Г. Корченко. – К. : НАУ, 2004. – 264 с.

94. Корченко О. Г. Сучасні нейромережеві методи та моделі оцінки параметрів безпеки ресурсів інформаційних систем / О. Г. Корченко, І. А. Терейковський, А. О. Дзюбаненко // Захист інформації. – 2014. – Т. 16, № 3. – С. 223-232.

95. Корченко А.Г. Построение систем защиты информации на нечетких множествах. Теория и практические решения / Корченко А.Г. – К. : МК-Пресс, 2006. – 320 с.

96. Корченко О. Г. Шкідливі програми та їх класифікація / О. Г. Корченко, К. П. Ануфрієнко // Защита информации : Сб. науч. трудов. – К. : НАУ, 2007. – С.26–32.

97. Котеров Д. PHP 5 / Д. Котеров, А. Костарев. – СПб. : БХВ-Петербург, 2005. – 1120 с.

98. Крамер Г. Математические методы статистики / Крамер Г. ; пер. с англ. А. С. Монина. – М. : Мир, 1976. – 648 с.

99. Красоткин А. Обнаружение сетевых атак / А. Красоткин // Мир ПК. – 2003. – № 6. – С. 24–26.

100. Крыжановский А.В. Применение искусственных нейронных сетей в системах обнаружения атак / А.В. Кржыжановский // Доклады ТУСУРа. – 2008. – № 2 (18), часть 1. – С. 37-41.

101. Круглов В.В. Искусственные нейронные сети / В. В. Круглов, В. В. Борисов. – М. : Горячая линия-Телеком, 2002. – 382 с.

102. Круглов В. В. Нечеткая логика и искусственные нейронные сети / В. В. Круглов, М. И. Дли, Р. Ю. Голунов. – М. : Горячая линия-Телеком, 2004. – 242 с.

103. Кузнецов Г. В. Классификация и анализ систем и методов обнаружения атак / Г. В. Кузнецов, А. М. Иванов // Захист інформації. – 2004. – № 4 – С. 4–11.

104. Кузнецов Г. В. Методы анализа данных для обнаружения атак в компьтерных сетях банковских структур / Г. В. Кузнецов, А. М. Иванов // Защита информации : Сб. науч. трудов. – К. : НАУ, 2004. – С. 45–50.

105. Кузьменко В.Г. VBA 2002 / В. Г. Кузьменко. – М. : БИНОМ, 2002. – 624 с.

106. Куссуль Э. М. Ассоциативные нейроподобные структуры / Э. М. Куссуль. – К. : Наукова думка, 1990, – 160 с.

107. Ли Ц. Оценивание параметров марковских моделей по агрегированным временным рядам / Ли Ц. ; пер.с англ. – М. : Статистика, 1977. – 221 с.

108. Лукацкий А. В. Корреляция на службе безопасности / А. В. Лукацкий // Byte. – 2003. – №10. – С. 10–12.

109. Лукацкий А. В. Обнаружение атак / А. В. Лукацкий. – СПб. : БХВ-Петербург, 2003. – 624 с.

110. Люгер Ф. Искусственный интеллект: стратегии и методы решения сложных проблем, 4-е издание / Люгер Ф. ; пер. с англ. Н. И. Галагана – М. : Вильямс, 2003. – 864 с.

111. Магницкий Ю.Н. Использование бинарной нейронной сети для обнаружения атак на ресурсы распределенных информационных систем / Ю.Н. Магницкий // Динамика неоднородных систем. — 2008. — С. 200-205.

112. Макаренко Н. Г. Лекции по нейроинформатике. Часть 2 / Н. Г. Макаренко. – М. : МИФИ, 2004. – 200 с.

113. Макнамара Д. Секреты компьютерного шпионажа / Макнамара Д. ; пер. с англ. – М. : БИНОМ, 2004. – 536 с.

114. Мелкумян К. В. COM как средство для реализации достоверной вычислительной базы / К. В. Мелкумян // Защита информации : Сб. науч. трудов. – К. : КМУГА, 1999. – С. 104–106.

115. Менаске Д. Производительность Web-служб. Анализ, оценка и планирование / Менаске Д., Виргилио А. ; пер. с англ. – СПб. : ДиаСофтЮп", 2003. – 480 с.

116. Нейман Дж. Теория самовоспроизводящихся автоматов / Нейман Дж. ; пер. с англ. – М. : Мир, 1971. – 384 с.

117. Нейман Дж. Теория игр и экономическое поведение / Нейман Дж., Моргенштерн О. ; пер. с англ. – М. : Наука, 1970. – 326 с.

118. Нижник Е. И. Математическое моделирование производительности файловых систем : автореф. дисс. на соискание научн. степени канд. техн. наук : спец. 05.13.18 "Математическое моделирование, численные методы и комплексы программ" / Е. И. Нижник. – М., 2007. – 24 с.

119. Новак Дж. Как обнаружить вторжение в сеть. Настольная книга специалиста по системному анализу = Network Intrusion Detection. An Analyst's Handbook / Джуди Новак, Стивен Норткатт, Дональд Маклахен ; Перевод И. Дранишникова. – М. : Лори, 2012. – 384 с.

120. Оберг Р. Технология COM+. Основы и программирование / Оберг Р. ; пер. с англ. – М. : Вильямс, 2001. – 480 с.

121. Обнаружение атак [Электронный ресурс] / Maxim Chirkov // Проект OpenNet : Портал по открытому ПО, Linux, BSD и Unix системам. – Электрон. дан. – [РФ], [2010]. – Режим доступа: World Wide Web. – URL: http://www.opennet.ru/prog/sml/85.shtml. – Загл. с экрана.

122. Огарок А. Виртуальные войны. Искусственный интеллект на защите от вирусов и программных закладок / А. Огарок, Д. Комашинский, Д. Школьников // Конфидент. – 2003. – №2 (50). – С. 64–69, 97.

123. Олешко Д. М. Інформаційна технологія прискорення синтезу нейронних мереж для вирішення задач прогнозування при прийнятті рішень : автореф. дис. на здобуття наук. ступеня канд. техн. наук : спец 05.13.06 "Автоматизовані системи управління та прогресивні інформаційні технології"/ Д. М. Олешко. – Одеса, 2005. – 19 с.

124. Орлов А. И. Высокие статистические технологи / А. И. Орлов // Заводская лаборатория. – 2003. – Т. 69, №11. – С. 55-60.

125. Орлов А. И. Прикладная статистика / А.И. Орлов. – М. : Экзамен, 2004. – 656 с.

126. Осовский С. Нейронные сети для обработки информации / С. Осовский. – М. : Финансы и статистика, 2002. – 344 с.

127. Осинский Л. М. Постановка и методы решения задач оптимального планирования мониторинга информационной безопасности вычислительных систем / Л. М. Осинский, А. Н. Мудрак // Защита информации : Сб. науч. трудов. – К. : НАУ, 2001. – С. 136–144.

128. Отнес Р. Прикладной анализ временных рядов / Отнес Р., Эноксон Л. ; пер. с англ. – М. : Мир, 1982. – 547 с.

129. Паркер Т. TCP/IP для профессионалов / Паркер Т., Сиян К. ; пер. с англ. Е. Матвеева. – СПб. : Питер, 2004. – 859 с.

130. Петров А. А. Определение оперативно-технических характеристик систем активной защиты информации / А. А. Петров // Захист інформації. – 2009. – № 1 – С. 73–75.

131. Петров А. С. Обеспечение защиты информации обрабатываемой динамической Web-системой за счет моделирования устойчивой к уязвимостям архитектуры / А. С. Петров, О. А.Талыкин // Вісник СНУ ім. Володимира Даля. – 2007. – №5. – С. 17-21.

132. Петров О. С. Аналіз уязвимости Web-систем / О. С. Петров, О. А. Таликін // Захист інформації. – 2006. – №2. – С. 32-42.

133. Плешко В. В.TopSOM: визуализация информационных массивов с применением самоорганизующихся тематических карт / В. В. Плешко, А. Е. Ермаков , Г. В. Липинский // Информационные технологии. – 2001. – № 8. – С. 8–11.

134. Поликарпов С.В., Дергачёв В.С., Румянцев К.Е., Голубчиков Д.М. Новая модель искусственного нейрона: кибернейрон и области его применения / Електронний ресурс http://arxiv.org/ftp/arxiv/ papers/0907/0907.0229.pdf.

135. Пучков Н. В. Использование искусственных нейронных сетей для контроля корректности информационно - технологического процесса / Н. В. Пучков // Новые промышленные технологи. – 1999. – № 3. – С. 79–84.

136. Рабинович З. Л. Подход к моделированию мыслительных процессов на основе нейроподобных растущих сетей / З. Л. Рабинович, В. А. Ященко // Кибернетика и системный анализ. – 1996. – № 5. – С.3–20.

137. Рассел С., Норвиг П. Искусственный интеллект: современный подход, 2-е изд / Рассел С., Норвиг П. ; пер.с англ. К.А. Птицына. – М. : Вильямс, 2007. – 1408 с.

138. Резник А. М. О природе интеллекта /А. М. Резник// Математические машины и системы. - 2008. - №1. - С.23-45.

139. Розенблат Ф. Аналитические методы изучения нейронных сетей / Розенблат Ф. ; пер.с англ. – М. : Зарубежная радиоэлектроника, 1965. – 150 с.

140. Руденко О.Г. Штучні нейронні мережі. Навч. посіб. / О.Г. Руденко, Є.В. Бодянський. – Харків: ТОВ "Компанія СМІТ", 2006. – 404 с.

141. Свинцов В. И. Смысловой анализ и обработка текста / В. И. Свинцов. – М. : Книга, 1979. – 272 с.

142. Сенешова М. Ю. Погрешности нейронных сетей. Вычисление погрешностей весов синапсов / М. Ю. Сенешова // Методы нейроинформатики : Сб. науч. трудов. – Красноярск : КГТУ. – 1998. – С. 204 – 212.

143. Сигеру О. Нейроуправление и его приложения / Сигеру О., Марзуки Х., Рубия Ю. ; пер. с англ. – М. : ИПРЖР, 2000. – 272 с.

144. Скопа О. О. Конвергенція глобальної інформаційної мережі: питання управління гетерогенними середовищами з урахуванням вимог тріади CIA / О. О. Скопа, Н. Ф. Казакова, Ж. Ю. Зеленцова // Сучасна спеціальна техніка. — 2014. — № 3(38). — С. 28-37.

145. Скопа, О. О. Показники якості та життєві цикли захищених інформаційно-вимірювальних систем / О. О. Скопа, С. Л. Волков, О. В. Грабовський // Вісник Східноукраїнського національного університету імені Володимира Даля. – 2013. – № 15(204). – Ч. 1. – С. 192-198.

146. Слеповичев И.И. Обнаружение DDoS-атак нечеткой нейронной сетью / И. И. Слеповичев, П. В. Ирматов, М. С. Комарова, А. А. Бежин // Известия Саратовского университета. – 2009. – Т. 9, сер. Математика. Механика. Информатика, вып. 3. – С. 84-89.

147. Стаханов А. А. Linux / А. А. Стаханов. – СПб. : БХВ-Петербург, 2004. – 912 с.

148. Талалаев А.А.Разработка нейросетевого модуля мониторинга аномальной сетевой активности / А.А. Талалаев, И.П. Тищенко, В.П.Фраленко, В.М. Хачумов // Нейрокомпьютеры: разработка и применение. — 2011. — № 7. — С. 32-38.

149. Таненбаум Э. Архитектура компьютера / Таненбаум Э. ; пер. с англ. – СПб. : Питер, 2006. – 697 с.

150. Таненбаум Э. Компьютерные сети / Таненбаум Э. ; пер. с англ. А. Леонтьева. – СПб.: Питер, 2002. – 848 с.

151. Таненбаум Э. Современные операционные системы. 2-е изд. / Таненбаум Э. ; пер. с англ. – СПб. : Питер, 2002. – 1036 с.

152. Тарасенко В. П. Метод застосування продукційних правил для подання експертних знань в нейромережевих засобах розпізнавання мережевих атак на комп’ютерні системи / В. П. Тарасенко, О. Г. Корченко, І. А. Терейковський // Безпека інформації. – 2013. – Т. 19, № 3. – С. 168-174.

153. Тейлор Дж. Введение в теорию ошибок / Тейлор Дж. ; пер. с англ. Л. Г. Деденко. – М. : Мир, 1985. – 272 с.

154. Терейковська Л.О. Визначення найбільш ефективної архітектури нейронної мережі, призначеної для розпізнавання голосових сигналів в Moodle / Л.О. Терейковська, І.А. Терейковський // Теорія і практика використання системи управління навчанням Moodle: матер. 2 міжнар. наук.-практ. конф. "MoodleMoot Ukraine-2014", (22-23 травня 2014 р.). – К.: КНУБА, 2014. – С.36.

155. Терейковська Л. О. Проблема голосової взаємодії в дистанційному навчанні вищого навчального закладу / Л. О. Терейковська, І. А. Терейковський // Науково-технічний збірник "Управління розвитком складних систем" Київського національного університету будівництва і архітектури. – 2013. – Випуск 13. –С. 157-161.

156. Терейковский И. А. Безопасность программного обеспечения, созданного с использованием семейства технологий COM, DCOM, COM+ / И. А. Терейковский // Захист інформації. – 2006. – № 1. – С. 55-67.

157. Терейковський І. А. Вдосконалення алгоритму навчання багатошарового персептрону, призначеного для розпізнавання мережевих атак / І. А. Терейковський // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. – 2012. – Випуск 2(24). – С. 65-70.

158. Терейковський І.А. Вдосконалення антивірусного захисту комп'ютерної мережі вищого навчального закладу / І. А. Терейковський // Сучасні тенденції розвитку вищої освіти, трансформація навчального процесу у технологію навчання: міжнар. наук.-метод. конф., 25-26 жовт. 2007 р.: тези допов. – К., 2007. – С. 366-367.

159. Терейковський І. А. Вдосконалення методики захисту інформації в корпоративних мережах, що використовують ресурси Internet / І.А. Терейковський // Вісник національного транспортного університету. – 2003. – № 8 – С. 13-16.

160. Терейковський І. А. Визначення оптимального методу контролю об’єктів захисту комп’ютерних мереж / І. А. Терейковський // Вісник КНУТД. – 2006. – № 5. – С. 39-44.

161. Терейковський І.А. Визначення оптимального типу нейронної мережі, призначеної для використання в програмних засобах захисту інформації / Терейковський І.А. // Сучасні тенденції розвитку технологій в інфокомунікаціях та освіті : матер. VIII наук. конф. (24-25 листопада 2011 р.). – К.: ДУІКТ, 2011. – С. 372-379.

162. Терейковський І. А. Використання нейронної мережі з радіальними базисними функціями в задачах діагностики стану захищеності програмного забезпечення / І. А. Терейковський // Науково-технічний збірник "Управління розвитком складних систем" Київського національного університету будівництва і архітектури. – 2010. – Випуск 3. – С. 111-114.

163. Терейковський І. А. Використання нейронної мережі Кохонена для розпізнавання спаму / І. А. Терейковський // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. – 2007. – Випуск 1(14). – С. 106-114.

164. Терейковський І. А. Використання нейронних мереж при розпізнаванні макровірусів / І. А. Терейковський // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. – 2006. – Випуск 2(13). – С. 176-183.

165. Терейковський І. А. Використання семантичної нейронної мережі в задачах моніторингу текстової інформації / І. А. Терейковський // Вісник ДУІКТ. – 2012.– Т.10, № 1.– С. 36-41.

166. Терейковський І.А. Використання експертних знань в процесі навчання нейронних мереж / І.А. Терейковський // Стратегії розвитку інформаційного культурно-освітнього та економічного простору України: Всеукр. наук.-практ. конф., 20-21 травня 2014 р.: тези допов. – К., 2014. – С.134-136.

167. Терейковський І.А. Використання семантичної нейронної мережі в задачах моніторингу текстової інформації / І.А. Терейковський // Сучасні інформаційно-комунікаційні технологі. COMINFO'2011: матер. VII міжнар. наук.-техн. конф. (10-14 жовтня 2011 р.). – К.: ДУІКТ, 2011. – С. 218-220.

168. Терейковський І. А. Дослідження ефективності функціонування веб-серверу / І. А. Терейковський // Комп'ютерне моделювання та інформаційні технології в науці, економіці та освіті: зб. наук. праць КЕІ КНЕУ. – Кривий Ріг, 2005. – С. 216-217.

169. Терейковський І. А. Дослідження стійкості серверних технологій Java від атак на відмову / І. А. Терейковський // Захист інформації. – 2004. – № 4. – С. 34-42.

170. Терейковський І. А. Использование возможностей Microsoft Word при создании Web-ориентированных вирусов / І. А. Терейковський // Защита информации: сб. науч. трудов НАУ. – 2004. – Выпуск 11. – С. 87-96.

171. Терейковський І. Захист Web-сайтів корпоративних інформаційних систем від атак на відмову / І. Терейковський // Зб. наук. праць ВІТІ НТУ України "КПІ". – 2004. – № 4 – С. 201-208.

172. Терейковський І. А. Захищеність Web-серверів Apache та IIS / І. А. Терейковський // Проблеми програмування. – 2005. – № 2. – С. 42-51.

173. Терейковський І. А. Концепція атаки Web-орієнтованих пошукових систем / І. А. Терейковський // Вісник ДУІКТ. – 2006. – Т. 3, № 3-4. – С. 67-71.

174. Терейковський І. А. Концепція визначення оптимального режиму контролю захищеності програмного забезпечення комп'ютерних систем / І. А. Терейковський // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. – 2006. – Випуск 1(12). – С. 88-96.

175. Терейковський І.А. Концепція використання марківських процесів для контролю атак на програмне забезпечення комп'ютерних систем та мереж / І.А. Терейковський // Захист інформації. – 2005. – № 3. – С. 4-12.

176. Терейковский И. А. Концепция защиты программного обеспечения Internet-сервера с использованием активной составляющей / И. А. Терейковский // Захист інформації. – 2005. – Спец. випуск. – С. 6-11.

177. Терейковський І. А. Методологія класифікації листів електронної пошти з використанням нейронних мереж / І. А. Терейковський // Захист інформації. – 2013. – Т. 15, № 2. – С. 115-121.

178. Терейковський І. А. Методи коннективізму та захист в них / І.А. Терейковський // Захист інформації. – 2009. – № 1 – С. 59-70.

179. Терейковський І. А. Методи обробки статистики при формуванні шаблонів нормальної поведінки Інтернет-серверів / І. А. Терейковський, Л. О. Терейковська // Інформаційна безпека: наук.-практ. конф., 26-27 березня 2009 р. : зб. текстів виступів. – К., 2009. – С. 56-60.

180. Терейковский И.А. Моделирование профилей нормального поведения компьютерных систем / И. А. Терейковский // Защита информации: сб. науч. трудов НАУ. – 2006. – Выпуск 13. – С. 103-108.

181. Терейковський І.А. Моделювання експлуатаційних параметрів веб-серверу системи дистанційного навчання / Терейковський І.А. // Сучасні комп'ютерні системи та мережі: розробка та використання ACSN’2011 : матер. 5-ої міжнар. наук.-техн. конф. (29 вересня – 01 жовтня 2011). – Львів: ЛПНУ, 2011. – С. 93-96.

182. Терейковський І. А. Негомогенна марківська модель прогнозування параметрів захисту веб-орієнтованих комп’ютерних систем / І. А. Терейковський, Л. О. Терейковська // Проблеми впровадження інформаційних технологій в економіці : матер. VIII міжнар. наук.-практ. інтернет-конф. (23.01.2012–30.03.2012). – Ірпінь: НУДПСУ, 2012. – С. 320-325.

183. Терейковський І. А. Нейромережева методологія розпізнавання інтернет-орієнтованого шкідливого програмного забезпечення / І. А. Терейковський // Безпека інформації. – 2013. – Т. 19, № 1. – С. 24-28.

184. Терейковський І. Нейронні мережі в засобах захисту комп’ютерної інформації: монографія / І. Терейковський. – К. : ПоліграфКонсалтинг. – 2007. – 209 с.

185. Терейковський І. А. Нейромережевий поведінковий аналізатор антивірусної системи / І. А. Терейковський // Захист інформації. – 2012. – № 2. – С. 67-70.

186. Терейковський І. А. Оцінка документованих можливостей Flash Macromedia для здійснення несанкціонованого доступу до інформації клієнтів Інтернет / І. А. Терейковський // Проблеми програмування. – 2004. – № 4 – С.112-118.

187. Терейковський І.А. Оптимізація архітектури нейронної мережі, призначеної для діагностики стану комп’ютерної мереж / І. А. Терейковський // Науково-технічний збірник "Управління розвитком складних систем" Київського національного університету будівництва і архітектури. – 2011. – Випуск 6. – С. 155-158.

188. Терейковський І. Оптимізація захисту відкритих корпоративних мереж / І. Терейковський, Л. Терейковська // Вісник КНТЕУ. – 2004. – № 1. – С. 103-112.

189. Терейковський І. А. Оптимізація захисту Web-орієнтованих інформаційних систем органів державної влади / І. А. Терейковський // Державне управління і право: зб. наук. праць Київського національного університету культури і мистецтв.– 2006. – Вип. 1, Ч. 2. – С. 97–105.

190. Терейковський І. А. Оптимізація структури двохшарового персептрону, призначеного для розпізнавання аномальних величин експлуатаційних параметрів комп’ютерної мережі / І. А. Терейковський // Науково-технічний збірник "Управління розвитком складних систем" Київського національного університету будівництва і архітектури. – 2011. – Випуск 5. – С. 128-131.

191. Терейковський І.А. Оптимізація структури та змісту корпоративних Web–сайтів / І.А. Терейковський // Вісник КНТЕУ. – 2004. – № 3. – С. 95-104.

192. Терейковский И.А. Парольная защита офисного электронного документооборота / И. А. Терейковский // Вісник ДУІКТ. – 2006. – Т. 4, № 2 – С. 109-115.

193. Терейковський І. А. Підвищення ефективності функціонування корпоративних web – сайтів / І .А. Терейковський // Вісник КНУТД. – 2004. – № 4. – С. 41-46.

194. Терейковський И. А. Применение семантического анализа содержимого электронных писем в системах распознавания спама / И. А. Терейковский // Захист інформації. – 2006. – № 4. – С. 49-60.

195. Терейковський І.А. Про використання вейвлет-перетворень та нейронних мереж для розпізнавання аномального стану комп'ютерної мережі / І. А. Терейковський // Вісник Університету "Україна". – 2011. – № 2. – С.60-65.

196. Терейковський І. А. Розпізнавання скриптових вірусів за допомогою нейронної мережі з радіальними базисними функціями / І. А. Терейковський // Науково-технічний збірник "Управління розвитком складних систем" Київського національного університету будівництва і архітектури. – 2010. – Випуск 4. – С. 104-108.

197. Терейковський І. А. Розпізнавання скриптових вірусів за допомогою багатошарового персептрону / І. А. Терейковський // Защита информации: сб. науч. трудов НАУ. – 2007. – Выпуск 14. – С. 206-212.

198. Термінологія в галузі захисту інформації в комп'ютерних системах від несанкціонованого доступу [Текст] : НД ТЗІ 1.1-003 – 1999. – Чин. 1999. 04.28. – К. : ДСТСЗІ СБ України, 1999. – 12 с.

199. Тимофеев А. Исследование и моделирование нейросетевого метода обнаружения и классификации сетевіїх атак / А.Тимофеев,

А.Браницкий // International Journal Information Technologies & Knowledge. – 2012. – Vol.6, Number 3. – P. 257-265

200. Тимченко А. А. Модель самоорганизации нейронной сети на примере задачи оценки уровня пожарной безопасности объекта / Тимченко А. А., Джулай А.Н. // Нейросетевые технологии и их применение : междунар. науч. конф., 4-5 дек. 2002 г. : тезисы докл. – Краматорск, 2002. – С.144–146.

201. Тихонов Э. Е. Методы и алгоритмы прогнозирования экономических показателей на базе нейронных сетей и модулярной арифметики / Э. Е. Тихонов, В. А. Кузьмищев. – Невинномысск: НИЭУП, 2004. – 166 с.

202. Ткаченко Р. О. Нейронні мережі прямого поширення з неітераційним навчанням : автореф. дис. на здобуття наук. ступеня д-ра. техн. наук : спец 05.13.06 "Автоматизовані системи управління та прогресивні інформаційні технології" / Р. О. Ткаченко. – Л., 2000. – 32 с.

203. Тэрано Т. Прикладные нечеткие системы / Тэрано Т., Асаи К., Сугэно М. ; пер. с японского Ю. Н. Чернышева – М. : Мир, 1993. – 364 с.

204. Тынкевич М.А. Экономико-математические методы / М. А. Тынкевич. – Кемерово, Кузбасский гос. техн. ун-т, 2000. – 177 с.

205. Уэйнпрат П. Apache для профессионалов / Уэйнпрат П. ; пер. с англ. И. Дранишников. – М. : Лори, 2001. – 473 с.

206. Уэнстром М. Организация защиты сетей Cisco / Уэнстром М. ; пер. с англ. – М. : Вильяме, 2005. – 768 с.

207. Уссермен Ф. Нейрокомпьютерная техника / Уссермен Ф. ; пер. с англ. – М. : Мир, 1992. – 284 с.

208. Федотов Е. В. Механизмы возможных атак в сети Internet / Е. В. Федотов // Защита информации : Сб. науч. трудов. – К. : НАУ, 2001. – С. 30–42.

209. Фленов М. Е. PHP глазами хакера / М. Е. Фленов. – СПб. : БХВ-Петербург, 2005. – 304 с

210. Хайкин С. Нейронные сети: полный курс, 2-е изд., испр. / Хайкин С. ; пер. с англ. Н. Н. Куссуль – М. : Вильямс, 2006. – 1104 с.

211. Харченко В.С. Новые информационные технологии и безопасность информационно–управляющих систем АЭС / В.С.Харченко, М.А. Ястребенецкий, В.В. Скляр // Ядерная и радиационная безопасность. – 2003. – Т. 6, № 2. – С. 19–28.

212. Хафизов А.Ф. Нейросетевая система обнаружения атак на WWW-сервер: дис. ... канд. техн. наук : 05.13.11 / А.Ф. Хафизов– Уфа, 2004–172 с.

213. Хеннан Э. Многомерные временные ряды / Хеннан Э. ; пер. с англ. А. С. Холева. – М. : Мир, 1974. – 576 с.

214. Хильер С. Создание приложений COM+ в среде Visual Basic. Руководство разработчика / Хильер С. ; пер. с англ.– М. : Вильямс, 2001. – 416 с.

215. Хмелёв Д. В. Распознавание автора текста с использованием цепей Маркова / Д. В. Хмелёв // Вестник МГУ, сер.9: Филология. – 2000. – № 2. – С. 115–126.

216. Хоглунд Г. Руткиты: внедрение в ядро Windows / Хоглунд Г., Батлер Дж. ; пер. с англ. – СПб. : Питер, 2007. – 285 с.

217. Хорошко В. О. Використання багатошарового персептрону для розпізнавання поштових скриптових вірусів / В.О. Хорошко, І.А. Терейковський // Сучасні інформаціно-комунікаційні технології: міжнар. наук.-техн. конф., 8-14 жовт. 2006 р. : тези допов. – К. : 2006. – С. 103-104.

218. Хорошко В. А. Использование искусственных нейронных сетей в задачах распознавания атак на компьютерные системы / В. А. Хорошко, И. А. Терейковский // Захист інформації. – 2006. – № 3. – С. 57-65.

219. Хорошко В. О. Концепція визначення оптимального режиму контролю Web-серверу системи дистанційного навчання / В. О. Хорошко,

Д. В. Чирков, І. А. Терейковський // Болонський процес: трансформація навчального процесу у технологію навчання: міжнар. наук.-метод. конф., 26-27 жовт. 2006 р. : тези допов. – К., 2006. – С. 224-225.

220. Хорошко В. О. Методичний підхід до формалізації задачі оцінювання ефективності системи захисту інформаційної системи ОВС України / В. О. Хорошко, В. А. Кудінов // Захист інформації. – 2004. – №4. – С. 11–18.

221. Хорошко В. О. Основи інформаційної безпеки / В. О. Хорошко, В. С. Чередниченко. – К. : ДУІКТ, 2008. – 186 с.

222. Хорошко В.О. Термінологічний довідник з питань технічного захисту інформації / В. О. Хорошко , С. Р. Коженевський , Д. В. Чирков. – К. : ДУІКТ, 2007. – 365 с.

223. Царегородцев В.Г. Извлечение явных знаний из таблиц данных при помощи обучаемых и упрощаемых искусственных нейронных сетей / В. Г. Царегородцев // Материалы XII Междунар. конф. по нейрокибернетике ["Проблемы нейрокибернетики"]. – Ростов-на-Дону : СКНЦ ВШ, 1999. – С. 245–249.

224. Царегородцев В. Г. Редукция размеров нейросети не приводит к повышению обобщающих способностей / В. Г. Царегородцев // Материалы XII Всеросс. семинара ["Нейроинофрматика и ее приложения"]. – Красноярск : КГТУ, 2004. – С. 163–165.

225. Царегородцев В. Г. Упрощение нейронных сетей – цели , идеи и методы / В. Г. Царегородцев // Нейрокомпьютеры: разработка , применение . – 2002. – № 4. – С. 5–13.

226. Цуриков О. М. Исследование конструктивно – эксплуатационных факторов в задаче оптимизации режима контроля / О. М. Цуриков, И. А. Терейковский // Техническая диагностика и неразрушающий контроль. – 1999. – №3. – С. 51 – 56.

227. Цуриков О. М. Исследование режима контроля и промывки фильтров жидкостных функциональных систем воздушных судов / О. М. Цуриков, И. А. Терейковский // Техническая диагностика и неразрушающий контроль. – 1999. – № 1. – С. 75-85.

228. Цуриков О. М. Оптимизация режима многопараметрического контроля на примере многопараметрического контроля / О. М. Цуриков, И. А. Терейковский // Вісник КМУЦА. 2-видання : зб. наук. праць. – К. : КМУЦА, 1999. – С. 217-221.

229. Цуриков О. М. Оптимизация режима однопараметрического контроля и связанных с ним профилактических работ агрегатов функциональных систем воздушных судов / О.М. Цуриков, И.А. Терейковский // Вісник КМУЦА. 1-видання : зб. наук. пр. – К. : КМУЦА, 1999. – С. 7-15.

230. Цюцюра С. В. Застосування нейронних мереж для розпізнавання «ідеального співрозмовника» серед користувачів соціальних мереж / С. В. Цюцюра, І. А.Терейковський, С. В. Палій // Системи навігації та управління. – 2013. – Випуск 4(28). – С.123-127.

231. Цюцюра С. В. Модифікація класичної нейронної мережі ймовірнісного типу для розпізнавання "ідеального співрозмовника" серед користувачів соціальних мереж / С. В. Цюцюра, І. А.Терейковський, С. В. Палій // Науково-технічний збірник "Управління розвитком складних систем" Київського національного університету будівництва і архітектури, 2014, Випуск 19. – С 118-123.

232. Цыбаков Б. С. Модель телетрафика на основе самоподобного случайного процесса / Б. С. Цыбаков // Радиотехника. – 1999. – № 5. – С. 34–38.

233. Червяков Н.И. Методы и принципы построения модулярных нейрокомпьютеров / Н. И. Червяков. – Ставрополь : Ставропольский военный ин-т сязи ракетных войск, 2000. – 212 с.

234. Шампандар А. Искусственный интеллект в компьютерных играх: как обучить виртуальные персонажи реагировать на внешние воздействия / Шампандар А. ; пер. с. англ. К. А. Птицына. – М. : Вильямс, 2007. – 786 с.

235. Широчин В.П. Біт-орієнтовані оцінки стійкості криптографічних алгоритмів. (Мухін В.Є., Широчин С.С.) Ж. Сучасна спеціальна техніка. № 1, 2010, с. 54 -59.

236. Широчин В.П. Мінімізація об'єму управляючого списку для планування подій в Petri-Nets моделях. (Шилов Ю. Н.) - К: Вісник Національного технічного університету України «КПІ», Інформатика, управління та обчислювальна техніка. № 56, 2012. с. 8-12.

237. Широчин В.П. Основи безпеки комп'ютерних систем. (Широчин С.В., Мухін В.Є.) - Київ: "Корнійчук", 2009. - 286 с.

238. Шохін Б. П. Вдосконалення контролю за станом захищеності ком'ютерної мережі на основі адаптивного моніторингу / Б. П. Шохін, О. М. Юдін, О. Є. Мазулевський // Зб. наук. праць військового інституту телекомунікацій та інформатизації національного технічного університету України "КПІ". – К. : КНТУ, 2004. – № 4. – С. 208–217.

239. Шуклин Д. Е. Модели семантических нейронных сетей и их применение в системах искусственного интеллекта : дис. … кан. техн. наук : 05.13.23 / Шуклин Дмитрий Евгеньевич. – Х., 2003. – 196 с.

240. Шумский С. А. Самоорганизующиеся карты финансовых индикаторов 200 крупнейших российских предприятий / С. А. Шумский, А. Н. Кочкин // Нейроинформатика-99 : науч. конф., 20-22 января 1999 г. : тезисы докл. – М., 1999. – С. 122–127.

241. Щеглов А. Ю. Защита компьютерной информации от несанкционированного доступа / А. Ю. Щеглов. – СПб. : Наука и техника, 2004. – 384 с.

242. Щеглов И. Н. Алгоритм формирования репрезентативной обучающей выборки искусственной нейронной сети / Щеглов И. Н.,

Демченко С. А., Подлесских А. А. // Нейроинформатика-99 : науч. конф., 20-22 янв. 1999 г. : тезисы докл. – М., 1999. – С. 405–407.

243. Яремчук Ю. Є. Вирішення проблеми доступності до мережі Інтернет шляхом динамічного балансування пропускної здатності каналу web-трафіку / Ю. Є. Яремчук, Д. О. Кец, Т. М. Жевега, К. В. Безпалий // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. – Вип. 1(23), 2012. – С. 58–64.

244. Яремчук Ю.Є. Моделювання вибору оптимального методу протидії загрозам інформаційній безпеці / Ю. Є. Яремчук, А. А. Шиян, Л. О. Нікіфорова // Реєстрація, зберігання і обробка даних. – Т. 16, №4, 2014. – С. 28–33.

245. Яремчук Ю.Є. Підхід до формування ієрархічних класифікацій методів захисту телекомунікаційних мереж від негативного впливу / Ю. Є. Яремчук, А. А. Шиян // Вимірювальна та обчислювальна техніка в технологічних процесах. – №4, 2014. – С. 226–230.

246. Яремчук Ю.Є. Вирішення проблеми доступності однотипних об’єктів мережі за доменним ім’ям в протоколі трансляції мережевих адрес / Ю. Є. Яремчук, Д. О. Кец, Є. С. Ніколаєв, Д. О. Іванішина // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. – Вип. 2(21), 2010. – С. 65–69.

247. Bezobrazov S., Golovko V. Neural Networks for Artificial Immune Systems: LVQ for Detectors Construction // International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. – Dortmund, 2007. – P. 180-184.

248. Bivens A., Palagiri C., Smith R., Szymansky B., Embrechts M. Network–Based Intrusion Detection Using Neural Networks // Proc. Intelligent Engineering Systems through Artificial Neural Networks ANNIE–2002, St. Louis, MO, Volume 12. – New York: ASME Press, 2002. P. 579–584.

249. Chen Y., Narayanan A. , Shaoning Pang , Ban Tao. Multiple sequence alignment and artificial neuralnetworks for malicious software detection // Natural Computation, 2012, P. 261 – 265.

250. Du Toit T., Kruger H. Filtering spam e-mail with Generalized AdditiveNeural Networks // Information Security for South Africa. 2012., P.1-8.

251. Hnatiuk S. Cyberterrorism: History of current trends and countermeasures. / S. Hnatiuk // Privacy Notice . - 2013 . - Volume 9 , № 2. - S. 118 - 129.

252. Skaruz J., Seredynski F. Recurrent neural networks towards detection of SQL attacks // Parallel and Distributed Processing Symposium, 2007.

253. Koch R. Attack Trends in Present Computer Networks / R. Koch, B. Stelte, M. Golling // 4th International Conference on Cyber Conflict [CYCON 2012], (Tallinn, Estonia, 5–8 June 2012). – 2012. – P. 225–236.

254. Mantel H. A Uniform Framework for the Formal Specification and Verification of Information Flow Security: Diss. … Doctor der Ingenieurwissenschaften / Heiko Mantel. – Saarbrücken, 2003. – 275 p.

255. AirSnare [Electronic resource] : [Intrusion Detection Software for Windows] / [AirSnare Project]. – Electronic data. – [USA], [2011]. – Mode of access: World Wide Web. – URL: http://home.comcast.net/~jay.deboer/airsnare/. – Language: English. – Description based on home page (viewed on Oct. 07, 2012).

256. Anderson J. Computer security threat monitoring and surveillance [Electronic resource] / J. Anderson // Computer Security Resource Center of National Institute of Standards and Technology / Computer Security Laboratory Department of Computer Science University of California at Davis. – Electronic data. – Gaithersburg, MD, USA : NIST, 1980. – Mode of access: World Wide Web. – URL: http://csrc.nist.gov/publications/history/ande80.pdf. – Language: English. – Description based on home page (viewed on Oct. 20, 2011).

257. Callegari C. A new statistical approach to network anomaly detection / C. Callegari, S. Vaton, M. Pagano // Proc. of Performance Evaluation of Computer and Tele-communication Systems (SPECTS). – 2008. – P. 441-447.

258. Denning D. An Intrusion Detection Model. // IEEE Transactions on Software Engineering, v. SE-13, № I, 1987, pp. 222-232.

259. Forrester S., Hofmeyr S., Longstaff T. A sense of self for Unix process // Proceedings of the 1996 IEEE Symposium on Security and Privacy. P/120-128? Los Alamos. CA. 1996. IEEE Computer Society Press.

260. Gavrilis D. Real-time detection of distributed denial-of-service attacks using RBF networks and statistical features / D. Gavrilis, E. Dermatas // Computer Networks. – 2005. – № 48. – P. 235-245.

261. IBM Proventia Network Anomaly Detection System [Electronic resource] / IBM // IBM. – Electronic data. – [Armonk, New York, USA]: IBM, [2011]. – Mode of access: World Wide Web. – URL: http://www.ibm.com/ru/services/iss/proventia_network_ anomaly_detection_system.html. – Language: English. – Description based on home page (viewed on Oct. 20, 2012).

262. IBM RealSecure Network [Electronic resource] / IBM // IBM. – Electronic data. – [Armonk, New York, USA]: IBM, [2010]. – Mode of access: World Wide Web. – URL: http://www.ibm.com/ru/services/iss/realsecure_network.html. – Language: English. – Description based on home page (viewed on Mar. 08, 2012).

263. Kazakova, Nadia. Model that Solve the Information Recovery Problems [Текст] / Nadia Kazakova, Oleksandr Skopa, Mikołaj Karpiński // Jornal of Telecommunications and Information Technology. — 2014. — №4. — P. 116-121. – ISSN 1509-4553 (SCOPUS).

264. Kharchenko V. Dependability of Safety–Critical Computer Systems through Component–Based Evolution / V. Kharchenko, V. Sklyar, A. Siora // Proceeding of International Conference on Dependability of Computer systems “DepCoS – RELCOMEX 2009”, 30 June – 02 July 2009. – Poland, Brunow, 2009. – P. 42–49.

265. Koch R. Attack Trends in Present Computer Networks / R. Koch, B. Stelte, M. Golling // 4th International Conference on Cyber Conflict [CYCON 2012], (Tallinn, Estonia, 5–8 June 2012). – 2012. – P. 225–236.

266. Korchenko O.G. Modern methods and neural network model parameter estimation of information systems security / O.G. Korchenko, I.A. Terejkowski// Aviation in the XXI-st century. Safety in Aviation And Space.

267. Kotov V. Detection of web server attacks using principles of immunocomputing / V. Kotov, V. Vasilyev // Proc. of 2nd World Congress on Nature and Biologically Inspired Computing. – 2010. – P. 25-30.

268. Kotov V. Detection of web server attacks using principles of immunocomputing / V. Kotov, V. Vasilyev // Proc. of 2nd World Congress on Nature and Biologically Inspired Computing. – 2010. – P. 25-30.

269. Planquart J.-P. Application of neural networks to intrusion detection [Electronic resource] / Jean-Philippe Planquart // SANS Information Security Reading Room. – Electronic data. – [USA] : SANS Institute, 2001. – Mode of access: World Wide Web. – URL: http://www.sans.org/reading_room/whitepapers/detection/application-neural-networks-intrusion-detection_336. – Language: English. – Description based on home page (viewed on May. 10, 2010).

270. Peng T. Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems / T. Peng, C. Leckie, K. Ramamohanarao // ACM Computing Surveys. – 2007. – Vol. 39, N 1. – 42 p.

271. Prelude-IDS [Electronic resource] : [Universal Open-Source Security Information & Event Management system] / The Prelude Team. – Electronic data. – [USA] : CS Systèmes d'Information, 2012. – Mode of access: World Wide Web. – URL: https://www.prelude-ids.org/. – Language: English. – Description based on home page (viewed on Oct. 20, 2012).

272. Recommended Security Controls for Federal Information Systems and Organizations. NIST Special Publication 800–53. Revision 3˜ National Institute of Standards and Technology, 2009. – 237 p.

273. Reznik A.M "Non-Iterative Learning for Neural Networks" Proceedings International Joint Conference on Neural Networks, Washington DC, July 10-16, 1999, №548.

274. Self-nonself fiscrimination in a computer / S. Forrest [et al.] // Proc. of 1994 IEEE Symp. on Research in Security and Privacy. – 1994. – P. 202-212.

275. Shyrochin V. Adaptive security mechanisms for the computer networks based on risk analysis. (Mukhin V.) // Journal of Qafqaz University: AZN. Mathematics and Computer Science. – Num. 1, Vol. 1, 2013. – pp. 11 – 16.

276. The Bro Network Security Monitor [Electronic resource] / The Bro Project. – Electronic data. – [USA] : The Bro Project, 2011. – Mode of access: World Wide Web. – URL: http://www.bro-ids.org/. – Language: English. – Description based on home page (viewed on Oct. 20, 2012).

277. The White House, Cyber space policy review. Assuring a Trusted and Resilient Information and Communications Infrastructure, 2010. – 76p. – [Электронный ресурс] режим доступа: http://msisac.cisecurity.org/awareness/documents/Cyberspace_Policy_Review_final.pdf.

278. Tripwire [Electronic resource] / [Tripwire, Inc.]. – Electronic data. – [Portland, OR, USA] : [Tripwire, Inc.], 2011. – Mode of access: World Wide Web. – URL: http://www.tripwire.org/. – Language: English. – Description based on home page (viewed on Oct. 20, 2012).

ДОДАТОК А

ОПИС WINDOWS-ЗАСТОСУНКУ ДЛЯ РОЗПІЗНАВАННЯ СКРИПТОВОГО ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Розроблений застосунок призначений для розпізнавання скриптового ШПЗ, написаного на мові програмування JavaScript, що розміщується на веб-сайтах. Застосунок базується на результатах п. 5.3 і реалізує ДШП пристосований для аналізу потенційно небезпечних функцій JavaScript. Він пристований для інтеграції в систему розпізнавання шкідливого програмного забезпечення та класифікації листів електронної пошти. Крім того цей застосунок може використовуватись самостійно.

Головне вікно застосунку проказане на рис. А.1.

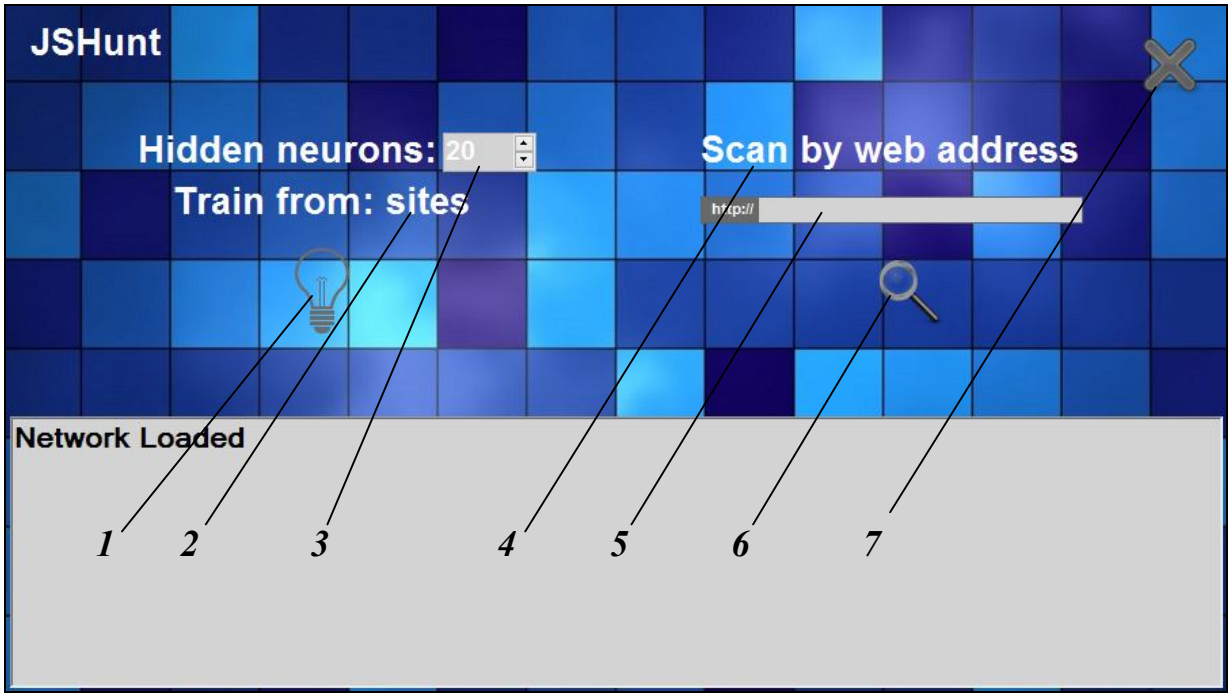


Рис. А.1. Головне вікно застосунку

Цифрами на рис. А.1 позначені наступні управляючі елементи:

– кнопка ініціалізації навчання ДШП;

– кнопка вибору джерела наачальних даних;

– поле вибору кількості елементів в СШН;

– кнопка вибору цілі сканування;

– поле для визначення адреси веб-сайту, який перевіряється на предмет наявності ШПЗ;

– кнопка ініціалізації розпізнавання;

– кнопка закриття вікна застосунку.

Відповідно до типової методики використання ДШП застосунок має два режими функціонування: навчання та сканування.

Навчання ДШП відбувається наступним чином. За замовчуванням вибраний режим навчання на прикладах програмного забезпечення веб-сайтів. Адреси цих сайтів повинні бути записані в файлах black.txt та white.txt, розташованих в поточній директорії. В файлі black.txt повині бути записані адреси сайтів з скриптовим ШПЗ, а в файлі white.txt – адреси сайтів без скриптовго ШПЗ.

Також передбачено можливість навчання ДШП на прикладах файлів, розташованих на локальному комп’ютері. Для цього треба натиснути на надпис режиму тренування (позначено цифрою 2). Після цього за стосунок автоматично перевіряє у поточній директорії наявність директорій black та white, а також прикладів у них. Алгоритм навчання показано на рис. А.2. Якщо джерела навчальних прикладів задані правильно, то починається навчання. В протилежному випадку генерується повідомлення про помилку.

Сканування також має два режими: сканування однієї веб-адреси чи сканування усіх сайтів з пошукової вибірки. Вибір режиму змінюється при натисненні на надпис режиму сканування над полем вводу адреси чи пошукового запиту відповідно. Тобто можна ввести веб-адресу та перевірити її на наявність небезпечного коду, або вибрати режим пошуку та ввести пошуковий запит. В цьому випадку після сканування за стосунок відкриває сторінку пошукової системи Google з результатами пошуку та результатом

сканування кожного сайту біля нього.

Алгоритм сканування показано на рис. А.3.

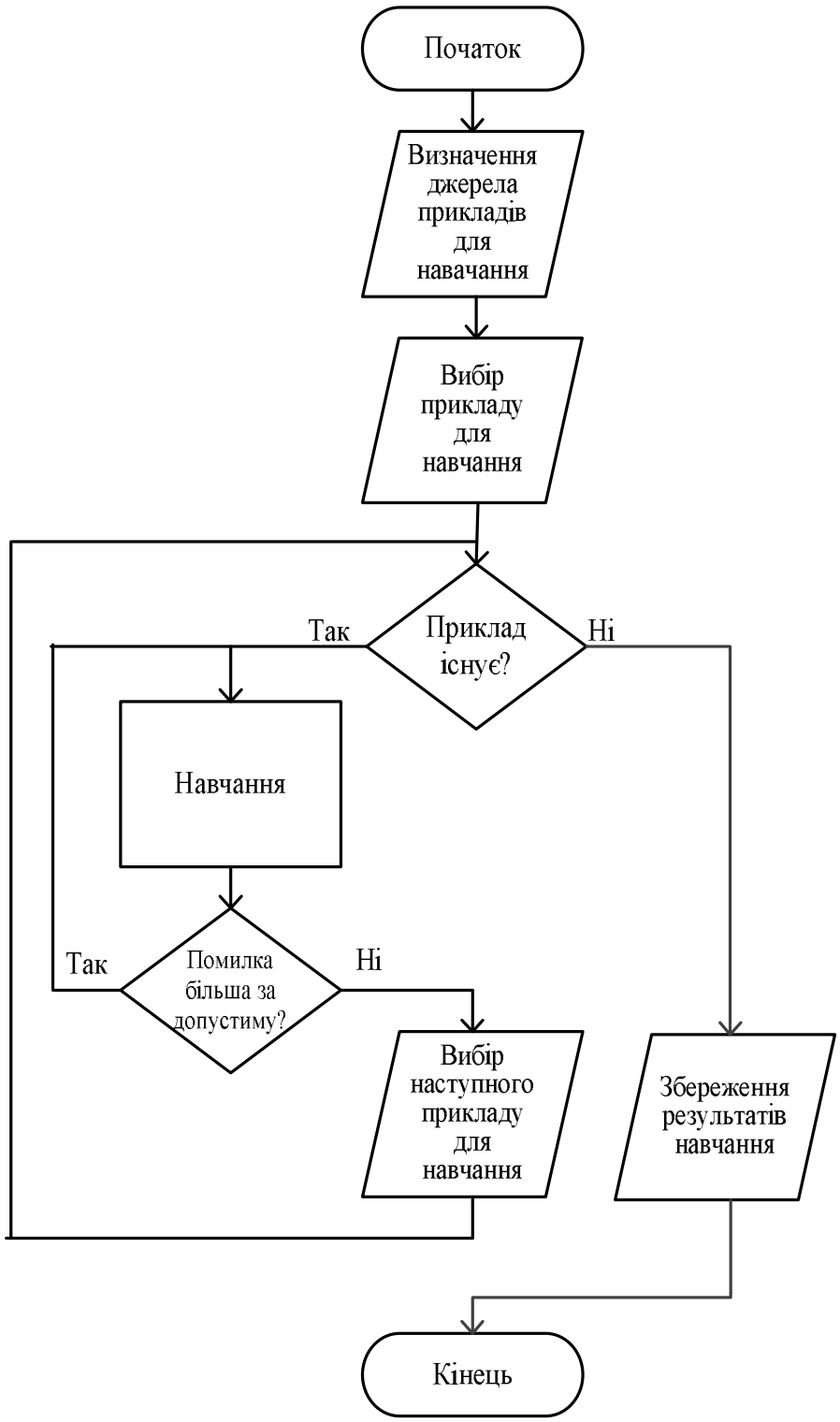


Рис. А.2. Алгоритм навчання

У нижній частині програми знаходиться текстовий елемент інтерфейсу, що відображає поточний стан програми. Туди виводяться повідомлення про успішне чи неуспішне навчання системи, імена поточних файлів чи сайтів, що оброблюються, а також результат розпізнавання.

Рис. А.3. Алгоритм сканування

Якщо вибраний режим сканування за пошуком, то замість вікно виводу

там з’являється вікно браузера.

В правій верхній частині вікна знаходиться кнопка 7, що закриває головне вікно програми. При цьому програма залишається у системному треї для швидкого доступу до неї за потребою.

Завершити виконання програму можна з контекстного меню.

Програмний продукт розроблено на мові C# у Visual Studio 2013 Express з використанням Windows Forms.

Вимоги до апаратного забезпечення:

- комп’ютер з тактовою частотою не нижче 1,2 ГГц;
- вільна оперативна пам’ять не менше ніж 2 Гб;
- вільний простір на жорсткому диску не менше ніж 10 Мб;
- SVGA монітор;
- SVGA відеоадаптер;
- мережева карта.

Вимоги до програмного забезпечення:

- операційна система Windows версії XP та вище;
- мережеве підключення до Інтернет;
- браузер, що підтримує стандарт HTML 4;
- підтримка протоколу передачі даних TCP/IP.

Програмний код розробленого Windows-застосунку:

```
using System;
using System.IO;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
```

```
using System.Diagnostics;
using System.Net;
using System.Threading;

namespace JSHunt
{
   public partial class JSHunt : Form
   {
      private int _hiddenDims = 150;
      private int _inputDims = 17;
      private int _iteration;
      private int _restartAfter = 10000000;
      private Layer _hidden;
      private Layer _inputs;
      private List<Pattern> _patterns;
      private Neuron _output;
      private Random _rnd = new Random();

      WebClient wclient = new WebClient();
      bool trained = false, websearch = false, holdon = false, usesites = true;
      string curaddr = "", websearchcode = "";
      int bl, wl, addr;

      public void Network()
      {
         LoadPatterns();
         Initialise();
         Train();
      }
```

```
        private void Train()
        {
            double error, mine = 1000, maxe = 0;
            do
            {
                error = 0;
                foreach (Pattern pattern in _patterns)
                {
                    double delta = pattern.Output - Activate(pattern);
                    AdjustWeights(delta);
                    error += Math.Pow(delta, 2);
                }
                status.Text = "Iteration {0}\tError {1:0.000}", _iteration, error;
                _iteration++;
                //if (_iteration > _restartAfter) Initialise();
                if (error < mine) mine = error;
                if (error > maxe) maxe = error;
                wstatus.Invoke((MethodInvoker)delegate { wstatus.Clear(); });
                wstatus.Invoke((MethodInvoker)delegate { wstatus.Text +=
"Status: Mine = " + mine + "\n"; });
                wstatus.Invoke((MethodInvoker)delegate { wstatus.Text +=
"Status: Error = " + error + "\n"; });
                wstatus.Invoke((MethodInvoker)delegate { wstatus.Text +=
"Status: Maxe = " + maxe + "\n"; });
            } while (error > 0.1);
            trained = true;
            wstatus.Invoke((MethodInvoker)delegate { nns.Clear(); });
            for (int i = 0; i < Neuron._weights.Count; i++)
                wstatus.Invoke((MethodInvoker)delegate { nns.Text +=
Neuron._weights[i].Value + "\n" ; });
```

```
        wstatus.Invoke((MethodInvoker)delegate { nns.SaveFile("nns.dat",
RichTextBoxStreamType.PlainText); });
        wstatus.Invoke((MethodInvoker)delegate { wstatus.Text += "Status:
Successfully trained\n"; });
    }

    private void Test()
    {
        StreamReader file = File.OpenText("input.txt");
        string values = "";
        while (!file.EndOfStream)
        {
            values = file.ReadLine();
        }
        file.Close();
        double a = Activate(new Pattern(values, _inputDims));
        if (websearch)
        {
            if (a < 0.3)
                websearchtl.Text += Math.Round(a, 2) * 100 + "%\n";
            else
            if (a < 0.5)
                websearchtl.Text += Math.Round(a, 2) * 100 + "%\n";
            else
            if (a <= 1)
                websearchtl.Text += Math.Round(a, 2) * 100 + "%\n";
        }
        else
        {
            if (a < 0.3)
```

```
                wstatus.Text += "Low threat possibility (" + Math.Round(a, 2) *
100 + "%)\n";
            else
            if (a < 0.5)
                wstatus.Text += "Medium threat possibility (" + Math.Round(a,
2) * 100 + "%)\n";
            else
            if (a <= 1)
                wstatus.Text += "High threat possibility (" + Math.Round(a, 2)
* 100 + "%)\n";
        }
    }

    private double Activate(Pattern pattern)
    {
        for (int i = 0; i < pattern.Inputs.Length; i++)
        {
            _inputs[i].Output = pattern.Inputs[i];
        }
        foreach (Neuron neuron in _hidden)
        {
            neuron.Activate();
        }
        _output.Activate();
        return _output.Output;
    }

    private void AdjustWeights(double delta)
    {
        _output.AdjustWeights(delta);
```

```
            foreach (Neuron neuron in _hidden)
            {
                neuron.AdjustWeights(_output.ErrorFeedback(neuron));
            }
        }

        private void Initialise()
        {
            _inputs = new Layer(_inputDims);
            _hidden = new Layer(_hiddenDims, _inputs, _rnd);
            _output = new Neuron(_hidden, _rnd);
            _iteration = 0;
            wstatus.Invoke((MethodInvoker)delegate { wstatus.Text += "Status:
Network Initialised\n"; });
        }

        private void LoadPatterns()
        {
            _patterns = new List<Pattern>();
            StreamReader file = File.OpenText("input.txt");
            while (!file.EndOfStream)
            {
                string line = file.ReadLine();
                _patterns.Add(new Pattern(line, _inputDims));
            }
            file.Close();
            //for (int i=0; i<input.Lines.Count(); i++)
            //    _patterns.Add(new Pattern(input.Lines[i], _inputDims));
        }
```

```
public JSHunt()
{
   if (PriorProcess() != null)
   {
      tray.Visible = false;
      Close();
   }

   InitializeComponent();
}

public static Process PriorProcess()
{
   Process curr = Process.GetCurrentProcess();
   Process[] procs = Process.GetProcessesByName(curr.ProcessName);
   foreach (Process p in procs)
   {
      if ((p.Id != curr.Id) && (p.MainModule.FileName ==
curr.MainModule.FileName))
         return p;
   }
   return null;
}

private void button1_Click(object sender, EventArgs e)
{
   this.WindowState = FormWindowState.Minimized;
   this.ShowInTaskbar = false;
}
```

```
private void close_MouseEnter(object sender, EventArgs e)
{
   close.Image = Properties.Resources.close;
}

private void close_MouseLeave(object sender, EventArgs e)
{
   close.Image = Properties.Resources.close_act;
}

private void scan_MouseEnter(object sender, EventArgs e)
{
   scan.Image = Properties.Resources.scan;
   if (websearch)
      scanlab.Text = "Search";
   else
      scanlab.Text = "Scan";
   scanlab.Visible = true;
}

private void scan_MouseLeave(object sender, EventArgs e)
{
   scan.Image = Properties.Resources.scan_act;
   scanlab.Visible = false;
}

private void train_MouseEnter(object sender, EventArgs e)
{
   train.Image = Properties.Resources.train;
   trainlab.Visible = true;
```

```
}

private void train_MouseLeave(object sender, EventArgs e)
{
   train.Image = Properties.Resources.train_act;
   trainlab.Visible = false;
}

private void tray_MouseDoubleClick(object sender, MouseEventArgs e)
{
   this.WindowState = FormWindowState.Normal;
   this.ShowInTaskbar = true;
}

private void exitToolStripMenuItem_Click(object sender, EventArgs e)
{
   Close();
}

private void showToolStripMenuItem_Click(object sender, EventArgs e)
{
   this.WindowState = FormWindowState.Normal;
   this.ShowInTaskbar = true;
}

private void jextract(int err)
{
   primitive example
   needs improvements
   string jtmp = "", tmp = "";
```

```
            rtb.Invoke((MethodInvoker)delegate { tmp = rtb.Text; });
            while (tmp.Contains("<script "))
            {
                try
                {
                    jtmp = jtmp + tmp.Substring(tmp.IndexOf("<script "),
tmp.IndexOf("</script>") - tmp.IndexOf("<script ") + 9);
                    tmp = tmp.Substring(tmp.IndexOf("</script>") + 8, 0);
                }
                catch
                {
                    wstatus.Invoke((MethodInvoker)delegate { wstatus.Text +=
"Status: error reading js " + err + "\n"; });
                    return;
                }
            }
            rtb.Invoke((MethodInvoker)delegate { rtb.Text = jtmp; });
        }

        private void open_Click(object sender, EventArgs e)
        {
            Close();
        }

        private void scansite()
        {
            bool black = false, white = false;
            if (trained == false)
            {
                wstatus.Text += "Status: you should train network first\n";
```

```
            return;
        }

        for (int i = 1; i < blacklist.Lines.Length; i++)
            if (websiteadr.Text == blacklist.Lines[i]) black = true;

        for (int i = 1; i < whitelist.Lines.Length; i++)
            if (websiteadr.Text == whitelist.Lines[i]) white = true;

        if (black)
        {
            wstatus.Text += "Status: site is in a blacklist\n";
            return;
        }

        if (white)
        {
            wstatus.Text += "Status: site is in a whitelist\n";
            return;
        }

        try
        {
            rtb.Text = wclient.DownloadString("http://" + curaddr);
        }
        catch
        {
            rtb.Clear();
            wstatus.Text += "Status: site http://" + curaddr + " is not
availiable\n";
```

```
            if (websearch) websearchtl.Text += "-1%\n";
            return;
        }
        jextract(0);
        wstatus.Text += "Status: site http://" + curaddr + " loaded for scan\n";

        if (rtb.Lines.Length == 0)
        {
            wstatus.Text += "Status: no input site\n";
            return;
        }
        Parse(0, 0);
        input.SaveFile("input.txt", RichTextBoxStreamType.PlainText);
        Test();
    }

    private void scan_Click(object sender, EventArgs e)
    {
        curaddr = websiteadr.Text;
        if (websearch)
        {
            webBrowser2.Navigate("https://www.google.com.ua/search?q=" + websiteadr.Text, null, null, "User-Agent: Opera/9.80 (S60; SymbOS; Opera Mobi/499; U; ru) Presto/2.4.18 Version/10.00");
        }
        else
            scansite();
    }
```

```
        private void Parse(int gb, int tr)
        {
            string inp = "", ln = "", rt = "";
            if (tr == 0) input.Invoke((MethodInvoker)delegate { input.Clear(); });
            for (int i = 0; i < _inputDims; i++)
            {
                vbd.Invoke((MethodInvoker)delegate { ln = vbd.Lines[i]; });
                rtb.Invoke((MethodInvoker)delegate { rt = rtb.Text; });
                if (rt.Contains(ln))
                    inp += "1, ";
                else
                    inp += "0, ";
            }
            inp += gb + "\n";
            input.Invoke((MethodInvoker)delegate { input.Text += inp; });
        }

        void StartTrainBySites()
        {
            bool err = false;
            string whitesite = "", blacksite = "", htmp = "";
            for (int i = 0; i < bl; i++)
            {
                try
                {
                    blacklist.Invoke((MethodInvoker)delegate { blacksite =
blacklist.Lines[i]; });
                    htmp = wclient.DownloadString("http://" + blacksite);
                    rtb.Invoke((MethodInvoker)delegate { rtb.Text = htmp; });
                    jextract(i);
```

```
            }
            catch
            {
               err = true;
            }
            if (err) err = false;
            else Parse(1, 1);
         }
         for (int i = 0; i < wl; i++)
         {
            try
            {
               whitelist.Invoke((MethodInvoker)delegate { whitesite =
whitelist.Lines[i]; });
               htmp = wclient.DownloadString("http://" + whitesite);
               rtb.Invoke((MethodInvoker)delegate { rtb.Text = htmp; });
               jextract(i);
            }
            catch
            {
               err = true;
            }
            if (err) err = false;
            else Parse(0, 1);
         }
         rtb.Invoke((MethodInvoker)delegate { rtb.Clear(); });
         input.Invoke((MethodInvoker)delegate { input.SaveFile("input.txt",
RichTextBoxStreamType.PlainText); });
         Network();
      }
```

```
        void StartTrainByFiles()
        {
            DirectoryInfo bdir = new
DirectoryInfo(Directory.GetCurrentDirectory() + "\\black");
            DirectoryInfo wdir = new
DirectoryInfo(Directory.GetCurrentDirectory() + "\\white");
            if (wdir.Exists && bdir.Exists && (wdir.GetFiles().Length != 0 ||
bdir.GetFiles().Length != 0))
            {
                wstatus.Invoke((MethodInvoker)delegate { wstatus.Text +=
"Status: training...\n"; });
                foreach (var item in bdir.GetFiles())
                {
                    rtb.Invoke((MethodInvoker)delegate {
rtb.LoadFile(Directory.GetCurrentDirectory() + "\\black\\" + item.Name,
RichTextBoxStreamType.PlainText); });
                    Parse(1, 1);
                }
                foreach (var item in wdir.GetFiles())
                {
                    rtb.Invoke((MethodInvoker)delegate {
rtb.LoadFile(Directory.GetCurrentDirectory() + "\\white\\" + item.Name,
RichTextBoxStreamType.PlainText); });
                    Parse(0, 1);
                }
                rtb.Invoke((MethodInvoker)delegate { rtb.Clear(); });
                input.Invoke((MethodInvoker)delegate {
input.SaveFile("input.txt", RichTextBoxStreamType.PlainText); });
                Network();
```

```
        }
        else wstatus.Invoke((MethodInvoker)delegate { wstatus.Text += "No
input files\n"; });
    }

    private void train_Click(object sender, EventArgs e)
    {
        _hiddenDims = Convert.ToInt32(nud_hn.Value);
        if (usesites)
        {
            if (File.Exists("black.txt") || File.Exists("white.txt"))
            {
                if     (File.Exists("black.txt"))     blacklist.LoadFile("black.txt",
RichTextBoxStreamType.PlainText);
                bl = blacklist.Lines.Length;
                if     (File.Exists("white.txt"))     whitelist.LoadFile("white.txt",
RichTextBoxStreamType.PlainText);
                wl = whitelist.Lines.Length;
                wstatus.Text += "Status: training...\n";
                new Thread(StartTrainBySites).Start();
            }
            else wstatus.Text += "No input site lists";
        }
        else new Thread(StartTrainByFiles).Start();
    }

    private void LoadNet()
    {
        nns.LoadFile("nns.dat", RichTextBoxStreamType.PlainText);
        _hiddenDims = nns.Lines.Length - 1;
```

```
            nud_hn.Value =  nns.Lines.Length - 1;
            LoadPatterns();
            Initialise();
            for (int i = 0; i < nns.Lines.Length - 1; i++)
                Neuron._weights[i].Value = Convert.ToDouble(nns.Lines[i]);
            wstatus.Text += "Network Loaded\n";
            trained = true;
        }

        private void Retrain()
        {
            Network();
        }

        private void JSHunt_Load(object sender, EventArgs e)
        {
            if (File.Exists("nns.dat"))
                LoadNet();
            else
                if (File.Exists("input.txt"))
                {
                    _hiddenDims = 20;
                    nud_hn.Value = 20;
                    new Thread(Retrain).Start();
                }
        }

        private void webBrowser1_DocumentCompleted(object sender,
WebBrowserDocumentCompletedEventArgs e)
        {
```

```
            websearchcode = webBrowser1.DocumentText;
            string tmp = websearchcode;
            while (tmp.IndexOf("<cite>") > 0)
            {
                tmp = tmp.Substring(tmp.IndexOf("<cite>"));
                websearchrep.Text += tmp.Substring(6, tmp.IndexOf("</cite>") -
6) + "\n";
                tmp = tmp.Substring(tmp.IndexOf("</cite>"));
            }
            websearchreptmp.Text = websearchrep.Text;
            websearchrep.Text = "";
            websearchrep.Clear();
            for (int i = 0; i < websearchreptmp.Lines.Length; i++)
                if (websearchreptmp.Lines[i] != "www.youtube.com")
                    websearchrep.Text += websearchreptmp.Lines[i] + "\n";
            tmp = websearchcode;
            while (tmp.IndexOf("<h3 class") > 0)
            {
                tmp = tmp.Substring(tmp.IndexOf("<h3 class"));
                websearchsites.Text += tmp.Substring(30, tmp.IndexOf("&amp")
- 30).Replace("https://", "").Replace("http://", "") + "\n";
                tmp = tmp.Substring(tmp.IndexOf("</cite>"));
            }
        }

        private void websearchsites_TextChanged(object sender, EventArgs e)
        {
            curaddr = websearchsites.Lines[addr];
            addr++;
            scansite();
```

```
    }

    private void websearchtl_TextChanged(object sender, EventArgs e)
    {
      holdon = true;
      for (int i = 0; i < websearchtl.Lines.Length; i++)
      {
        if (websearchrep.Lines[websearchtl.Lines.Length - 1] != "www.youtube.com")
          websearchcode = websearchcode.Replace("<cite>" + websearchrep.Lines[i] + "</cite></div>", "<cite>" + websearchrep.Lines[i] + "</cite>" + websearchtl.Lines[i] + "</div>");
      }
      webBrowser2.DocumentText = websearchcode;
    }

    private void webBrowser2_DocumentCompleted(object sender, WebBrowserDocumentCompletedEventArgs e)
    {
      if (holdon == false)
        webBrowser1.Navigate("https://www.google.com.ua/search?q=" + websiteadr.Text, null, null, "User-Agent: Opera/9.80 (S60; SymbOS; Opera Mobi/499; U; ru) Presto/2.4.18 Version/10.00");
    }

    private void label4_Click(object sender, EventArgs e)
    {
      if (websearch)
      {
        websearch = false;
```

```
            textBox1.BackColor = Color.DimGray;
            textBox1.Text = "http://";
            webBrowser2.Visible = false;
            wstatus.Visible = true;
            labser.Text = "Scan by web address";
        }
        else
        {
            websearch = true;
            textBox1.BackColor = Color.Green;
            textBox1.Text = "Google";
            webBrowser2.Visible = true;
            wstatus.Visible = false;
            labser.Text = "Scan by search phrase";
        }
    }

    private void label3_Click(object sender, EventArgs e)
    {
        if (usesites)
        {
            usesites = false;
            tflab.Text = "Train from: folders";
        }
        else
        {
            usesites = true;
            tflab.Text = "Train from: sites";
        }
    }
```

```
}

public class Layer : List<Neuron>
{
   public Layer(int size)
   {
      for (int i = 0; i < size; i++)
         base.Add(new Neuron());
   }

   public Layer(int size, Layer layer, Random rnd)
   {
      for (int i = 0; i < size; i++)
         base.Add(new Neuron(layer, rnd));
   }
}

public class Neuron
{
   private double _bias;
   private double _error;
   private double _input;
   private double _lambda = 6;
   private double _learnRate = 0.5;
   private double _output = double.MinValue;
   public static List<Weight> _weights;

   public Neuron() { }

   public Neuron(Layer inputs, Random rnd)
```

```
        {
            _weights = new List<Weight>();
            foreach (Neuron input in inputs)
            {
                Weight w = new Weight();
                w.Input = input;
                w.Value = rnd.NextDouble() * 2 - 1;
                _weights.Add(w);
            }
        }

        public void Activate()
        {
            _input = 0;
            foreach (Weight w in _weights)
            {
                _input += w.Value * w.Input.Output;
            }
        }

        public double ErrorFeedback(Neuron input)
        {
            Weight w = _weights.Find(delegate(Weight t) { return t.Input ==
input; });
            return _error * Derivative * w.Value;
        }

        public void AdjustWeights(double value)
        {
            _error = value;
```

```
            for (int i = 0; i < _weights.Count; i++)
            {
                _weights[i].Value += _error * Derivative * _learnRate *
_weights[i].Input.Output;
            }
            _bias += _error * Derivative * _learnRate;
        }

        private double Derivative
        {
            get
            {
                double activation = Output;
                return activation * (1 - activation);
            }
        }

        public double Output
        {
            get
            {
                if (_output != double.MinValue)
                {
                    return _output;
                }
                return 1 / (1 + Math.Exp(-_lambda * (_input + _bias)));
            }
            set
            {    _output = value;    }
        }
```

```
    }
    public class Pattern
    {
        private double[] _inputs;
        private double _output;
        public Pattern(string value, int inputSize)
        {
            string[] line = value.Split(',');
            if (line.Length - 1 != inputSize)
                throw new Exception("Bad network configuration");
                _inputs = new double[inputSize];
            for (int i = 0; i < inputSize; i++)
            {    _inputs[i] = double.Parse(line[i]);        }
            _output = double.Parse(line[inputSize]);
        }
        public double[] Inputs
        {         get { return _inputs; }        }
        public double Output
        {         get { return _output; }          }
    }
    public class Weight
    {
        public Neuron Input;
        public double Value;
    }
}
```

ДОДАТОК Б

АКТИ ВПРОВАДЖЕНЬ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОГО ДОСЛІДЖЕННЯ

ЗАТВЕРДЖУЮ:
Проректор з наукової роботи
Національного авіаційного
університету
_____ В.Харченко
"__02__" _____ 2015 р.

АКТ

впровадження результатів дисертаційної роботи Терейковського Ігоря Анатолійовича на тему «Нейромережеві моделі, методи і засоби оцінювання параметрів безпеки Інтернет-орієнтованих інформаційних систем» на здобуття наукового ступеня доктора технічних наук у навчальний процес Національного авіаційного університету.

Комісія у складі: голова – завідувач кафедрою безпеки інформаційних технологій (БІТ) Корченко О.Г., доцент кафедри БІТ Гнатюк С.О., доцент кафедри БІТ Щербина В.П. склали даний акт про те, що результати дисертаційної роботи Терейковського Ігоря Анатолійовича " Нейромережеві моделі, методи і засоби оцінювання параметрів безпеки Інтернет-орієнтованих інформаційних систем" впроваджені у навчальний процес та використовуються на кафедрі БІТ у 2014-2015 навчальному році при викладанні наступних дисциплін: "Комп'ютерні мережі", "Безпека інформаційних і комунікаційних систем"

№ з/п	Назва роботи, що впроваджується	Форма впровадження	Ефективність від впровадження
	1	2	3
1.	Розпізнавання кібератак на основі оцінки параметрів безпеки	Лекція	Ознайомлення студентів з підходами до розпізнавання різнотипних кібератак за допомогою оцінки параметрів безпеки
2.	Нейромережеві засоби оцінки параметрів безпеки	Лекція	Ознайомлення студентів з сучасними нейромережевими моделями, методами та системами оцінки параметрів безпеки
3.	Оптимізація нейромережевої моделі оцінки параметрів безпеки	Лекція	Ознайомлення студентів з методикою оптимізації виду та параметрів нейромережевої моделі оцінювання параметрів безпеки для розпізнавання різнотипних кібератак на ресурси інформаційних систем

Голова комісії,
завідувач кафедрою безпеки
інформаційних технологій О. Корченко

Члени комісії:
доцент кафедри безпеки
інформаційних технологій С. Гнатюк

доцент кафедри безпеки
інформаційних технологій В. Щербина

«ЗАТВЕРДЖУЮ»

Проректор
Київського національного університету
будівництва і архітектури
доктор економічних наук, професор
_____ Лагутін Г.В.
«___» _____2015 року

АКТ ВПРОВАДЖЕННЯ
результатів дисертаційних досліджень
здобувача Терейковського Ігоря Анатолійовича

Цим актом підтверджується, що результати наукових досліджень у рамках дисертаційної роботи «Нейромережеві моделі, методи і засоби оцінювання параметрів безпеки Інтернет-орієнтованих інформаційних систем» щодо розробки нейромережевих моделей, призначених для класифікації листів електронної пошти з метою розпізнавання спаму та витоків текстової інформації, використані фахівцями кафедри інформаційних технологій Київського національного університету будівництва і архітектури при створенні засобів захисту інформації системи дистанційного навчання.

Використання означених матеріалів дозволило на 19-23% зменшити похибку класифікації листів електронної пошти.

Завідуючий кафедрою
інформаційних технологій,
д.т.н., професор Білощицький А.О.

«ЗАТВЕРДЖУЮ»
Директор Інституту проблем моделювання в енергетиці ім. Г.Є. Пухова НАНУ
доктор технічних наук, професор
_____ Євдокімов В.Ф.
«12» січня 2015 року

АКТ ВПРОВАДЖЕННЯ
результатів дисертаційної досліджень
здобувача Терейковського Ігоря Анатолійовича

Цим актом підтверджується, що результати наукових досліджень у рамках дисертаційної роботи «Нейромережеві моделі, методи і засоби оцінювання параметрів безпеки Інтернет-орієнтованих інформаційних систем» щодо розробки нейромережевих моделей призначених для використання в якості управляючого елементу в системах розпізнавання кібератак на об'єкти захисту інформаційних систем використані фахівцями Інституту проблем моделювання в енергетиці ім. Г.Є. Пухова НАН України при виконанні НДР "Дослідження та розробка методів оцінювання захищеності інформації в розподілених високопродуктивних інформаційних системах при вирішенні задач енергетики" (шифр МОД-Д), що виконується за відомчим замовленням Відділення фізико-технічних проблем енергетики НАН України.

Повнота, достовірність та актуальність результатів отриманих Терейковським І.А. дозволяє підвищити оперативність створення алгоритмів функціонування апаратних засобів захисту інформації.

Науковий керівник НДР,
зав. відділом «Теорії моделювання»,
к. т. н., с.н.с. Давиденко А.М.

ЗАТВЕРДЖУЮ
Перший проректор Національного
технічного університету України
«Київський політехнічний інститут»

_____ Якименко Ю.І.
_____ 2015 р.

А К Т

про впровадження результатів дисертаційного дослідження докторанта кафедри безпеки інформаційних технологій Національного авіаційного університету Терейковського Ігоря Анатолійовича на тему «НЕЙРОМЕРЕЖЕВІ МОДЕЛІ, МЕТОДИ І ЗАСОБИ ОЦІНЮВАННЯ ПАРАМЕТРІВ БЕЗПЕКИ ІНТЕРНЕТ-ОРІЄНТОВАНИХ ІНФОРМАЦІЙНИХ СИСТЕМ» на здобуття наукового ступеня доктора технічних наук.

Комісія у складі: голова – завідувач кафедри СПіСКС НТУУ «КПІ», д.т.н., проф. Тарасенко В.П.; члени комісії – професор кафедри СПіСКС НТУУ «КПІ», д.т.н., Зайцев В.Г., доцент кафедри СПіСКС НТУУ «КПІ», к.т.н., доц. Тесленко А.К. цим Актом засвідчує, що результати дисертаційного дослідження Терейковського Ігоря Анатолійовича використані співробітниками кафедри СПіСКС НТУУ «КПІ» при підготовці і викладанні курсу лекцій і лабораторного практикума навчальних дисциплін: «Захист інформації в комп'ютерних системах», «Технологія проектування спеціалізованих операційних систем».

Зокрема впроваджено: новий метод метод подання експертних знань для нейромережевих засобів оцінки параметрів безпеки, в якому за рахунок продукційних правил представлення навчальних прикладів, забезпечується оперативність розпізнавання та розширення множини видів кібератак, характеристики яких не представлені в статистичних даних; новий метод визначення часових характеристик використання нейромережевих засобів, в якому завдяки розробленим аналітичним залежностям для визначення терміну розробки, отримана можливість визначення доцільності застосування нейромережевих засобів для виявлення кібератак на заданий об'єкт захисту.

Голова комісії
д.т.н., проф. Тарасенко В.П.

Члени комісії
д.т.н., проф. Зайцев В.Г.
к.т.н., доц. Тесленко А.К.