II). Таким образом, на основании формул (13) – (16) случайное кодирование (декодирование) сообщений кодом Препараты B может быть реализовано с линейными временной и емкостной сложностями, составляющими порядка $16N$ элементарных операций и $4N$ бит соответственно.

Литература: ***1.*** *Maurer U.M. Secret key agreement by public discussion from common information // IEEE Trans. on Inform. Theory. – 1993. – Vol. 39. – № 3. – P. 733-742.* ***2.*** *Ahlswede R., Csiszar I. Common randomness in information theory and cryptography – Part 1: Secret sharing // IEEE Trans. Inform. Theory. – 1993. – V. 39. – № 4. – P. 1121-1132.* ***3.*** *Bennet C. H.,Brassard G., Maurer U. M.Generalized privacy amplifications // IEEE Trans. Inform. Theory. – 1995. – V. 41. – № 6. – P. 1915-1923.* ***4.*** *Decatur S.,Goldreich O., Ron D. A probabilistic error-correcting scheme //* http://eprint.iacr.org./1997/005. ***5.*** *Thangarai A., Dihidar S., Calderbank A. R., McLaughlin S., Merolla J.-M.Capacity achiving codes for the wire-tap channel with applications to quantum key distribution //* http://eprint.arXiv.IT/0411003v1.– 2 Nov.2004. ***6.*** *Wyner A. D. The Wire-Tap Channel // Bell System Techn. J. – 1975. – V. 54. – № 8. – P. 1355-1388.* ***7.*** *Коржик В. И., Яковлев В. А. Неасимптотические оценки кодового зашумления одного канала // Проблемы передачи информации. – 1981.– Т. 17.– В. 4.– С. 11-18.* ***8.*** *Алексейчук А. Н. Случайное кодирование в канале связи с аддитивным шумом, распределенным на конечной абелевой группе // Захист інформації. – 2002.– № 3.– С. 7-16.* ***9.*** *Алексейчук А. Н. Оптимальное случайное кодирование равновероятных сообщений в q-ичном симметричном канале // Захист інформації. – 2002. – № 4. – С. 49-58.* ***10.*** *Иванов В. А. О методе случайного кодирования // Дискретная математика.– 1999. – Т. 11. – В. 3. – С. 99-108.* ***11****.Алексейчук А. Н., Сергиенко Ю. В. Оценки стойкости и способ реализации кодовой защиты дискретных сообщений с использованием каскадных кодов // Электронное моделирование. – 2003. – Т. 25. – № 5. – С. 33-44.* ***12.*** *Алексейчук А. Н., Гришаков С.В. Нелинейное случайное кодирование в системах передачи информации по каналу связи с отводом // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. – В. 8. – К.: 2004. – С. 133-140.* ***13.*** *Hammous A. R., Kumar P. V., Calderbank A. R., Sloane N.J.A., Sole P. The Z_4-linearity of Kerdock,Preparata,Goethals and related codes // Bull. Amer. Math. Soc. – 1993. – V. 29. – № 2.– P. 218-222.* ***14****. Нечаев А. А. Код Кердока в циклической форме // Дискретная математика. – 1989. – Т. 1. – Вып. 4. – С. 123-139.* ***15.*** *Кузьмин А. С., Нечаев А. А. Построение помехоустойчивых кодов с использованием линейных рекуррент над кольцами Галуа // Успехи матем. наук – 1992. – Т. 47. – № 5. – С. 183-184.* ***16.*** *Calderbank A. R., Sloane N.J.A. Modular and p-adic ciclic codes // Design, Codes and Cryptography. – 1995. – V. 6. – P. 21-35.* ***17.*** *Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов: Пер. с англ. – М.: Мир, 1979. – 535 с.*

УДК 681.3.06

ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ ПРИ РОЗПІЗНАВАННІ МАКРОВІРУСІВ

Ігор Терейковський
Державний університет інформаційно-комунікаційних технологій

Анотація: **Розглянута можливість використання нейронної мережі типу багатошаровий перспетрон в системах розпізнавання комп'ютерних макровірусів. Сформована архітектура такого перспетрона та показані принципи визначення переліку вхідних параметрів.**
Summary: **The opportunity of use neuronet such as many layers perspetron in systems of recognition computer macroviruses is considered. The architecture such perspetron is generated and the principles of definition of the list of entrance parameters are shown.**
Ключові слова: **Вірус, антивірус, нейронні мережі, перспетрон.**

Значний обсяг успішних вірусних атак на інформаційні ресурси сучасних автоматизованих інформаційних систем визначає важливість розробки ефективних технологій антивірусного захисту. Не зважаючи, на те, що розробці антивірусних систем присвячено достатньо багато досліджень, практичний досвід та висновки [1] вказують на їх відносно низьку ефективність. При цьому потенційно висока небезпека комп'ютерних вірусів зафіксована як у вітчизняних, так і в закордонних нормативних документах в галузі захисту інформації [2, 3].

Багато в чому необхідність вдосконалення антивірусного захисту зумовлена недосконалістю системи діагностики, яка не здатна адекватно реагувати на появу нових вірусів, створених за допомогою сучасних технологій. Завдання ускладнюється тим, що для кожної операційної системи та програмного середовища потрібно використовувати власну методику розпізнавання вірусів. Разом з цим, для більшості

автоматизованих інформаційних систем характерним є організація електронного документообігу при адміністративному обмеженні повноважень користувачів на інсталяцію програмного забезпечення та доступу до файлів. В цьому випадку найбільш ймовірним є зараження офісних документів макровірусами. Ймовірність зараження підвищується завдяки тому, що технологія створення і розповсюдження макровірусів загальнодоступна завдяки мережі Інтернет. При цьому більшість сучасних антивірусних систем не в змозі забезпечити достатній рівень захисту від макровірусів [1, 4]. Це вказує на актуальність проведення досліджень в напрямку застосування нових підходів для розпізнавання макровірусів.

І Недоліки загальнопоширених методів розпізнавання макровірусів

В даній роботі будемо використовувати визначення комп'ютерного вірусу, наведене в вітчизняній нормативній документації в галузі захисту інформації. Комп'ютерний вірус це програма, що розмножується та поширюється самочинно. При цьому комп'ютерний вірус може порушувати цілісність інформації, програмне забезпечення та режим роботи обчислювальної техніки [2, 3]. Макровірус це вірус, який використовує можливості макромов, вбудованих в системи обробки даних [5, 6]. Відзначимо, що можливості сучасних макромов досить широкі. Тому макрос, вбудований в систему обробки даних, який заражає файли типів exe, com або bat можливо вважати макровірусом.

На сьогодні найбільш поширені макровіруси, пристосовані для розповсюдження в середовищі MS Office, написані мовою програмування Visual Basic for Applications. Це пояснюється популярністю як самого пакету MS Office так і мови VBA, яка крім компанії Microsoft використовується великою кількістю інших фірм – виробників програмного забезпечення. Крім того відомі макровіруси адаптовані для функціонування в таких розповсюджених прикладних пакетах як AutoCAD, 1С "Предприятие" та 1С "Бухгалтерия". Незважаючи на середовище розповсюдження макровірус представляє собою макрос, що виконується внаслідок реалізації певної події, наприклад при відкритті документа або при натисненні користувачем певної клавіші на панелі інструментів. В основному для захисту від макровірусів, як і для вірусів у цілому, використовуються антивірусні сканери та блокувальники.

Принцип роботи антивірусних сканерів базується на постійній або періодичній перевірці файлів, секторів дисків та системної пам'яті на предмет виявлення в них відомих та невідомих вірусів. При цьому, найчастіше для виявлення вірусів використовується метод пошуку сигнатур. Сигнатура вірусу представляє собою характерну для цього вірусу послідовність команд програмного коду. Для виявлення відомих вірусів програмний код, що аналізується, співвідноситься з базою даних відомих сигнатур вірусів. Особливістю пошуку сигнатур макровірусів є те що сканер може аналізувати програмний код макросу в текстовому вигляді. В багатьох випадках це спрощує роботу сканера та дозволяє проводити аналіз функціональності макроса.

В тому випадку, коли фрагмент програмного коду, що аналізується, відповідає певній сигнатурі, це свідчить про зараження файлу певним вірусом (макровірусом). З цієї причини метод сигнатур дозволяє розпізнавати тільки відомі віруси та відкриває шлях для обходу антивірусного захисту поліморфним та стелс вірусам. Характерною ознакою поліморфних вірусів (макровірусів) є зміна свого програмного коду в процесі розповсюдження. Стосовно макровірусів ця зміна може передбачати вставку в макрос нейтрального програмного коду, заміна назв макросів та процедур, заміна програмного коду іншим з аналогічною функціональністю. Таким чином, поліморфному вірусу в базі даних сигнатур має відповідати не один запис, а декілька. Ознакою стелс вірусів є маскування або приховування свого програмного коду від антивірусного сканера. Для цього можуть використовуватись різноманітні методи. Наприклад, частина програмного коду стелс макровірусу може знаходитись в зашифрованому вигляді. Після активізації стелс макровірусу спрацьовує процедура розшифровки коду. Розшифрований код надалі використовується макровірусом в процесі функціонування. Крім того, для приховування програмного коду макровірусу використовується його захист за допомогою паролю або збереження частини макровірусу в тілі документу. Стелс технології можуть використовуватись і при розробці поліморфних вірусів.

Ще одним важливим недоліком методу сигнатур є необхідність постійного оновлення антивірусної бази даних користувачами та постійного функціонування розгалуженої служби підтримки, яка виявляє нові віруси та оновлює базу даних. Крім того, всі антивірусні сканери відзначаються відносно низькою швидкістю пошуку всіх видів вірусів.

Для розпізнавання невідомих вірусів та макровірусів в деяких антивірусних сканерах разом з методом сигнатур використовується так званий евристичний аналізу програмного коду. При цьому в різних антивірусних системах застосовуються різні евристичні методи, реалізація яких практично не документуються. Проте аналіз джерел [1, 4, 5] вказує на те, що в більшості випадків базою цих методів є статистичний аналіз послідовності виконання програмного коду об’єкта, що перевіряється. Відзначимо, що сучасні евристичні методи дозволяють виявити тільки близько 50 % вірусів, сигнатура яких не

представлена в антивірусній базі даних. Крім того, ці методи не дозволяють самонавчатись розпізнавати невідомі макровірусу в процесі функціонування.

Антивірусні блокувальники – це резидентні програми, що аналізують події, які відбуваються в операційній системі на предмет виявлення потенційно небезпечних ситуацій, характерних під час розмноження вірусу. Виявивши таку ситуацію блокувальники сигналізують про неї та (або) забороняють її виконання.

В загальному випадку до небезпечних ситуацій відносяться, наприклад, запис в EXE та COM файли, запис в boot-сектор жорсткого диску, спроба програми зостатись резидентною, спроба програми приєднатись до іншого процесу в операційній системі. Стосовно макровірусу, пристосованого до розмноження в середовищі MS Officce, до таких подій відносяться запис в шаблон документу Normal.dot, спроба зміни рівня безпеки, відключення анти вірусного захисту, створення нової процедури, тощо. Крім того, деякі блокувальники макровірусів забороняють макросам виконувати потенційно небезпечні функції. Наприклад, блокувальник Office Guard, що входить до складу антивірусного комплексу "Антивирус Касперского", дозволяє заборонити макросам виклик функцій API операційної системи та виконувати операції з файлами. До загальновизнаних переваг антивірусних блокувальників відносять можливість розпізнавати та блокувати невідомі віруси. Не зважаючи на це, великий процент хибних спрацювань та великі затрати ресурсів обчислювальної системи сучасних блокувальників вірусів і макровірусів завадять їх широкому практичному застосуванню. Крім того, за великим рахунком, функціональність блокувальників макровірусів багато в чому повторює функції захисту вбудованого в пакет MS Office та має ті ж недоліки. Вибір користувачем високого рівня захисту від макровірусів завадить виконанню корисних макросів, а вибір низького рівня не захищає від макровірусів. При цьому тонка настройка параметрів блокувальника викликає у звичайного користувача значні труднощі та може значно змінюватись залежно від того, з яким документом працює користувач. З цих причин в багатьох випадках ні звичайний користувач, ні адміністратор комп'ютерної системи не в змозі оперативно настроїти параметри сучасних блокувальників макровірусів. Отже, проблема розпізнавання макровірусів як за допомогою сканерів, так і за допомогою блокувальників далека від свого вирішення, що підтверджується незалежним тестуванням антивірусних засобів [4]. Відзначимо, що основні труднощі розв'язання вказаної проблеми полягають в підвищенні достовірності розпізнавання вірусу в результаті аналізу програмного коду макроса або (та) в результаті аналізу подій в програмному середовищі, в якому виконується макрос. При цьому підвищити достовірність розпізнавання можливо за рахунок використання в антивірусних сканерах такої технології штучного інтелекту як нейронні мережі (НМ) [4]. Основною передумовою застосування НМ є апробованість та доведена ефективність в задачах розпізнавання образів на основі неповної та зашумленої інформації. Отже, завданням даної роботи є розробка НМ, призначеної для розпізнавання макровірусів

II Загальні принципи функціонування НМ типу – багатошаровий перспетрон

НМ представляє собою сукупність довільним чином об'єднаних елементів (штучних нейронів). НМ оброблює вхідну інформацію і в процесі зміни свого стану формує сукупність вихідних сигналів. Величини вихідних сигналів є відповіддю НМ на поставлену перед нею задачу. При цьому нейрони представляють собою прості процесори, обчислювальні можливості яких полягають у комбінуванні вхідних сигналів та розрахунку вихідного сигналу на основі деякої функції активації. Розрізняють три типи нейронів: вхідні – сприймають сигнал із зовнішнього середовища, вихідні – віддають сигнал у зовнішнє середовище та сховані – безпосередньо не сприймають сигнал від зовнішнього середовища і не відсилають дані в зовнішнє середовище. Вихідний сигнал нейрона надсилається іншим нейронам зваженими зв'язками. Зв'язки, якими сигнали поширюються в напрямку від входу НМ до її виходу, називаються прямими. Зв'язки, якими сигнали поширюються в протилежному напрямку, називаються зворотними. Кожному зв'язку між нейронами відповідає певний ваговий коефіцієнт (вага зв'язку). Вхідний сигнал нейрону (*net*) розраховується таким чином:

$$net = w_0 + \sum_{i=1}^{n} x_i w_i \ , \qquad (1)$$

де w_0 – фіксована величина, n – кількість вхідних зв'язків, x_i – величина i-го зв'язку, w_i – вага і-го зв'язку.

Функція активації представляє правило розрахунку вихідного значення нейрону, яке передається іншим нейронам або в зовнішнє середовище. Як функцію активації досить часто використовують лінійну (2), лінійну з погашенням від'ємних імпульсів (3), порогову (4) та сигмоїдальну функції (5):

$$f(net) = net \ , \qquad (2)$$

$$f(net) = \begin{cases} net, \exists net > z \\ 0, \exists net \le z \end{cases}, \tag{3}$$

$$f(net) = \begin{cases} 1, \exists net \ge z \\ 0, \exists net < z \end{cases}, \tag{4}$$

$$f(net) = \frac{1}{1 + e^{-a \times net}}, \tag{5}$$

де z – деяке пирогове значення, a – визначений коефіцієнт.

Вихідний сигнал нейрона відповідно до функції активації може бути більшим або меншим нуля. При цьому зв'язки, якими до нейрону поступають сигнали більші за нуль, називаються збуджуючими. Зв'язки, якими поступають від'ємні сигнали, називаються гальмуючими. НМ потребують навчання, суть якого полягає в розрахунку таких параметрів НМ, при яких вона найкраще вирішує поставлену проблему, наприклад, класифікує образ залежно від величин характерних вхідних параметрів. При визначеній архітектурі навчання полягає в розрахунку вагових коефіцієнтів. Відзначимо, що архітектура НМ залежить від кількості нейронів різних типів, структури та видів зв'язків між нейронами.

Аналіз [7, 8] дозволяє сформулювати висновок: на сьогодні для кожного класу прикладних задач використовуються різні типи НМ. При цьому однією із найбільш досліджених та апробованих в задачах розпізнавання образів, що базуються на аналізі множини взаємокорельованих дискретних параметрів, є НМ з архітектурою типу – багатошаровий перспетрон.

В загальному випадку багатошаровий перспетрон, показаний на рис. 1, – це НМ, яка складається із декількох послідовно з'єднаних між собою шарів штучних нейронів.

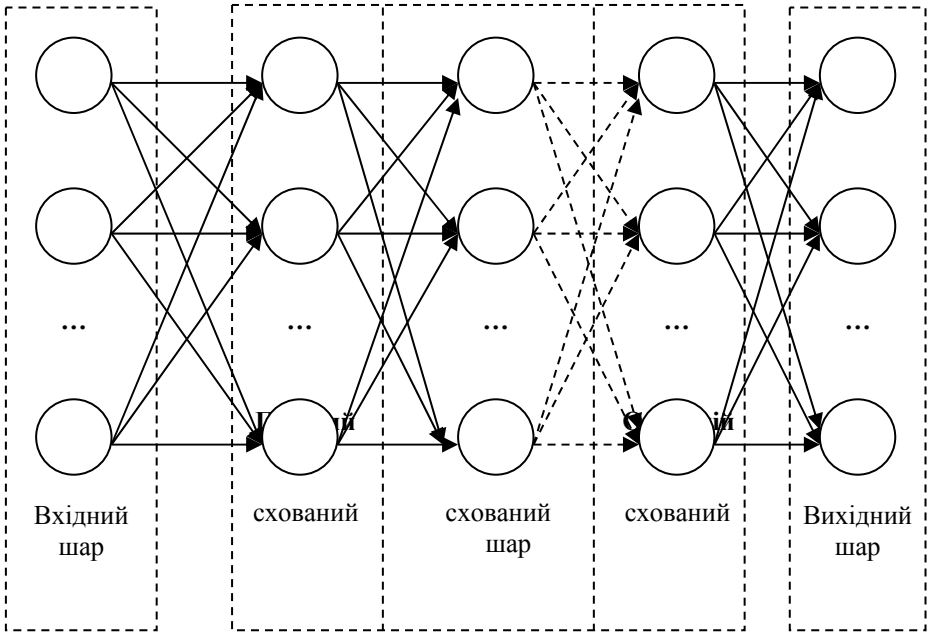

Рисунок 1 – Структура багатошарового перспетрону

Відзначимо, що при підрахунку кількості шарів вхідний шар не враховують. Тобто: в двохшаровий перспетрон складається із вхідного, одного схованого та вихідного шару.

Як правило, кожен нейрон схованого шару приймає всі вихідні сигнали попереднього шару, а його вихідний сигнал надсилається всім нейронам наступного шару. Особливістю багатошарового перспетрону є наявність тільки прямих гальмуючих або збуджуючих зв'язків між сусідніми шарами нейронів. При цьому кожен нейрон в схованому шарі характеризується унікальним вектором вагових коефіцієнтів. Для вхідних нейронів досить часто використовується лінійна (2), лінійна з погашенням від'ємних імпульсів (3) та порогова функції активації (4). Для схованих нейронів як правило використовують порогову (4) та сигмоїдальну функцію активації (5). Теоретично доведено [7, 8], що одного схованого шару нейронів з сигмоїдальною функцією активації достатньо для апроксимації будь-якої функції із наскільки завгодно

високою точністю. В більшості випадків вихідні елементи перспетрона виконують тільки розрахунок власних вхідних сигналів (1), тому функція активації для них не потрібна. Розраховані величини називають величинами вихідних елементів.

Максимальна кількість образів (P), яку може запам'ятати двохшаровий перспетрон з пороговою функцією активації схованого шару виду (4), можна оцінити таким чином [7]:

$$\frac{L_w}{m} < P < \frac{L_w}{m} log\left(\frac{L_w}{m}\right), \tag{6}$$

де L_w – кількість зв'язків між нейронами схованого шару та вихідними нейронами, m – кількість вихідних нейронів.

Місткість двохшарового перспетрона з сигмоїдальною функцією активації виду (5) дещо більша [7, 8]. Місткість перспетрона з кількістю шарів більше двох теоретично не визначена, хоча вважається дещо вищою місткості двохшарового перспетрона з тими ж показниками L_w та m [7, 8]. Приблизну кількість нейронів в схованих шарах (L) можливо оцінити за допомогою (7) або (8) [7, 8]:

$$L \approx \frac{L_w}{n+m}, \tag{7}$$

$$\frac{N}{10} - n - m \le L \le \frac{N}{2} - n - m, \tag{8}$$

де N – кількість елементів навчальної вибірки, n – розмірність вхідного сигналу.

Як правило, розмірність вхідного сигналу відповідає кількості елементів вхідного шару. При відомій кількості вихідних нейронів кількість зв'язків між нейронами схованого шару та вихідними нейронами можливо визначити за допомогою (6).

Навчання перспетрона виконується методом "навчання з вчителем" та полягає в визначенні таких вагових коефіцієнтів зв'язків нейронів схованого шару, які дозволяють найкраще вирішувати поставлену задачу. Процес навчання починається з ініціалізації вказаних вагових коефіцієнтів випадковими величинами. Після цього на вхід НМ подаються параметри, що відповідають відомим образам. Відзначимо, що з точки зору перспетрона відомий образ означає відомий набір значень вихідних параметрів. Якщо реальні вихідні параметри відрізняються від цих значень, то вагові коефіцієнти нейронів схованого шару уточнюються за допомогою алгоритму оберненого розповсюдження помилок [7, 8]. Алгоритм базується на мінімізації функції помилки перспетрона всій множині навчальної вибірки. Пошук мінімуму помилки реалізується методом градієнтного спуску. Вказаний алгоритм навчання достатньо ефективний, але накладає обмеження на використання тільки гладких активізаційних функцій в схованих шарах нейронів. Після навчання перспетрон може розпізнавати вхідні дані, або нести інше змістовне навантаження. Інформація про отриманий в процесі навчання досвід зберігається у вигляді вагових коефіцієнтів зв'язків схованих нейронів.

Аналіз [7, 8] дозволив сформувати алгоритм розробки перспетрона для вирішення конкретної задачі.

- Визначити номенклатуру та допустимі величини вхідних параметрів.
- Підготувати тестову та навчальну вибірку.
- Визначити максимальну та мінімальну межу загальної кількості схованих нейронів.
- В межах допустимої області вибрати загальну кількість схованих нейронів.
- Вибрати кількість схованих шарів та кількість нейронів в кожному схованому шарі.
- Вибрати вид та параметри функцій активації для всіх типів нейронів.
- Провести навчання.
- Провести тестування.
- Якщо результати тестування не задовільні – змінюємо параметри перспетрона. Для цього повторити п. 4 – 8.

Таким чином, для розв'язання практичної задачі необхідно сформувати множину вхідних параметрів (п. 1), розробити архітектуру перспетрона (п. 2 – 6) та провести його навчання (п. 2, 7 – 9).

III Розробка архітектури перспетрона для розпізнавання макровірусів

В номенклатурі вхідних параметрів перспетрона насамперед слід врахувати здатність макровірусу до саморозповсюдження. Очевидно, що шляхи саморозповсюдження багато в чому залежать від програмних засобів створення макровірусу. В даній роботі акцент ставиться на макровірусах, призначених для зараження документів MS Office та написаних з використанням мови програмування VBA. При цьому VBA є практично повноцінною об'єктно-орієнтованою мовою програмування, яка дозволяє працювати з

файловою системою, встановлювати мережеві з'єднання, маніпулювати процесами і потоками, здійснювати виклик функцій API операційної системи та запускати на виконання зовнішні програми. Таким чином, шляхи розповсюдження макровірусів MS Office не обмежені програмним середовищем зараженого документу. При цьому можливості VBA дозволяють макровірусам використовувати різноманітні засоби розповсюдження. Наприклад, макровірус MS Word може заражати документи (креслення) AutoCAD або командні файли. Засобами розповсюдження макровірусу в цьому випадку можуть бути об'єкти відповідних бібліотек, функції API операційної системи та програмні додатки. Можливе середовище, засоби та ознаки розповсюдження, показані на рис. 2.

Рисунок 2 – Ознаки розповсюдження макровірусів

Крім ознак розповсюдження номенклатура вхідних параметрів перспетрона повинна враховувати характерні ознаки макровірусів. Серед вказаних ознак можливо виділити групи: автоматизації запуску, ігнорування помилок, маскування макровірусу, деструктивних функцій. Приблизний перелік ознак представлено в табл. 1.

Таблиця 1 – Характерні ознаки макровірусів

Назва групи	Перелік ознак
автоматизації запуску	використання автомакросів
ігнорування помилок	використання операторів ігнорування помилок та переходу на певний рядок програмного коду після виникнення помилок

маскування макровірусу	шифрування та дешифрування макросу, захист макросу паролем, відключення захисту від макровірусів, блокування натиску клавіш, перевизначення кодів клавіш, зміна шрифту макросів, запис інформації в буфер обміну та зчитування інформації із буферу, відключення редактору VBA, знищення панелі інструментів для роботи з макросами та шаблонами, ігнорування повідомлень програмного середовища, поліморфізм макровіруса, використання API функцій для порушення функціонування антивіруса, знищення файлу з вірусом, знищення модулю з вірусом, знищення (модифікація) процедури (функції) з вірусом
деструктивні функції	форматування жорстких дисків, реалізація записів в файли, знищення файлів, встановлення паролів на файли, встановлення мережевих з'єднань, доступ до поштових клієнтів

Безпосередньо вхідними параметрами перспетрона будуть фрагменти програмного коду (назви процедур, параметрів, об'єктів, бібліотек, методів та властивостей об'єктів), що відповідають ознакам процесу розмноження та характерним ознакам макровірусів. Наприклад, однією з характерних ознак макровірусу є використання автомакросів. Відповідними параметрами будуть AutoOpen, AutoClose та інші аналогічні назви макросів. Очевидно, що вхідні параметри можуть мати тільки два значення, 1 – якщо ознака присутня та 0 в протилежному випадку. Кількість вхідних параметрів (N) буде дорівнювати кількості відповідних фрагментів. В першому наближенні визначимо, що кількість схованих шарів дорівнює 1. При цьому приблизну кількість схованих нейронів визначимо за допомогою (6) – (8), вважаючи, що кількість образів, яку повинен запам'ятати перспетрон, в 1,5 – 2 рази перевищує кількість елементів навчальної вибірки. Для збільшення гнучкості системи розпізнавання, визначимо в вихідному шарі три елементи, які будуть відповідати трьом можливим ситуаціям: макровірус відсутній, макровірус знайдено, підозра на макровірус. До підозрілих слід віднести макроси, в яких знайдено тільки окремі ознаки макровірусів, наприклад зашифрований програмний код або функції пошуку та функції встановлення паролю на документи MS Word. Таким чином, перспетрон буде класифікувати всі макроси на три класи: безпечні, макровіруси та підозрілі. Для вхідних елементів виберемо лінійну функцію активації (2), для схованих елементів сигмоїдальну (5) з параметром a=0,1. Кількість схованих елементів та величину параметру a можливо уточнити відповідно до пункту 9 загального алгоритму розробки перспетрона. При цьому навчальну та тестову вибірки можливо сформувати на основі аналізу сигнатур макровірусів, що входять до складу баз даних антивірусних програм. Крім того як до навчальної, так і до тестової вибірки слід включити макроси без ознак макровірусів та макроси, що мають певні ознаки макровірусів. Наприклад, макрос з назвою AutoOpen та оператором ігнорування помилок On Error Resume Next. На практиці можливим результатом роботи перспетрона будуть ненульові величини всіх трьох вихідних елементів. Так перспетрон буде сигналізувати про ймовірність належності макросу до кожного із визначених класів. Остаточну класифікацію можна провести так: макрос належить до класу, величина вихідного елементу якого найбільша. При цьому слід ввести певні обмеження. Наприклад, всі макроси, для яких величина вихідного параметру, що відповідає за безпечний клас, менша ніж 0,5, будуть вважатись макровірусами.

IV Висновки

Підвищити достовірність розпізнавання макровірусів можливо за рахунок використання в антивірусних сканерах блоку розпізнавання на базі багатошарового перспетрону. Розроблена архітектура такого перспетрону. Запропоновано підхід до визначення його вхідних параметрів.

Перспективним шляхом підвищення рівня захисту програмного забезпечення комп'ютерних систем є використання НМ в антивірусних блокувальниках та вдосконалення запропонованого перспетрона для розпізнавання макровірусів типу троянський кінь.

Література: ***1.*** *А. Щеглов, К. Щеглов Антивирусное противодействие механизмами зашиты информации от несанкционированного доступа. Требования к реализации // Daily.Sec.Ru 01. 11. 2005 http://daily.sec.ru/dailypblshow.cfm?rid=9&pid=13821.* ***2.*** *ДСТУ 3396. 2-97. Захист інформації. Технічний захист інформації. Терміни і визначення./ К.: Держстандарт України, 1998. – 16 с.* ***3.*** *Богуш В. М., Кривуца В. Г., Кудін А. М. Інформаційна безпека. Термінологічний навчальний довідник. / К.: Издатель ООО "Д. В. К.", 2002. – 508 с.* ***4.*** *А. Огарок, Д. Комашинский, Д. Школьников Виртуальные войны. Искусственный интеллект на защите от вирусов и программных закладок / Конфидент № 2 (50) 2003 г. с. 64 – 69, 97.* ***5.*** *Касперский Е. В. Компьютерные вирусы: что это такое и как с ними бороться / М.: СК Пресс, 1998. – 288 с.* ***6.*** *Терейковский И. А. Использование возможностей Microsoft Word при создании Веб – ориентированных*

вирусов / Защита информации Сб. н. т. НАУ - 2004 Выпуск 11, с. 87 – 96. ***7.*** *Круглов В. В., Борисов В. В. Искусственные нейронные сети. / М.: Горячая линия-Телеком, 2002. – 382 с.* ***8.*** *Каллан Р. Основные концепции нейронных сетей. // М.: Вильямс, 2003. – 288 с.*

УДК 621.391

ОЦЕНКА СЛОЖНОСТИ РЕАЛИЗАЦИИ АЛГОРИТМОВ ДЕКОДИРОВАНИЯ ТУРБОКОДОВ ПРИ ДЕКОДИРОВАНИИ БИТА ИНФОРМАЦИИ НА ЦИФРОВЫХ СИГНАЛЬНЫХ ПРОЦЕССОРАХ

Сергей Зайцев, Сергей Ливенцев, Борис Горлинский
Специальный факультет СБ Украины ВИТИ НТУУ "КПИ"

Анотація: **Проведено оцінку складності декодування біта інформації при апаратно-програмній реалізації турбокодів на цифрових сигнальних процесорах. Розглянуто Map, Max Log Map та Log Map алгоритми декодування турбоходів.**
Summary: **In the article the analysis of complexity of decoding the information bit is made at hardware-software realization turbocodes on digital signal processors. Are considered Map, Max Log Map and Log Map algorithms of decoding turbocodes.**
Ключові слова: **Турбокоди, цифрові сигнальні процесори, алгоритми декодування.**

I Введение

Турбокоды (ТК) обладают высокими корректирующими свойствами при низких отношениях сигнал-шум в канале связи, могут быть реализованы программным, аппаратным или программно-аппаратным способом. Как правило, практически ТК реализуются программно-аппаратным способом на цифровых сигнальных процессорах (ЦСП) [1, 2]. В качестве ЦСП широко используется высокопроизводительный 32-разрядный процессор ADSP-2106x семейства ADSP-21000 с плавающей точкой, который используется для обработки речи, графики, звука и др. Процессоры семейства ADSP-21000 выполняют все команды за один цикл. Они поддерживают высокую тактовую частоту, а также полный набор арифметических операций, включающий помимо традиционных умножения, сложения, вычитания и комбинированного умножения/сложения, примитивы деления ($1/x$ и $\sqrt{10}$), сравнения, определения абсолютного значения, операции min, max, Shift (арифметический сдвиг), Rotate (циклический сдвиг) и др. [3, 4]. Практически все они являются общими и для других семейств высокопроизводительных процессоров. Все вычислительные операции выполняются в арифметико-логическом устройстве, умножителе и устройстве сдвига вычислительного устройства ЦСП.

Наиболее сложным элементом в структуре ТК является декодер, который использует для декодирования специальные алгоритмы. За оценку сложности алгоритмов декодирования ТК в работе принято количество элементарных операций, необходимых для декодирования одного бита информации ЦСП.

II Постановка задачи

Анализ работ [5 – 8] показывает, что при оценке сложности декодирования одного бита информации не полностью учитывалось количество проверочных символов с выхода рекурсивного систематического сверточного кодера (РССК), обязательное использование нормализации, а основные алгебраические операции не были полностью представлены как элементарные. В связи с этим возникает задача анализа сложности алгоритмов декодирования ТК с учетом данных особенностей.

Таким образом, **целью работы** является аналитическое описание и сравнительный анализ сложности реализации декодирования бита информации при использовании *Map, Max Log Map* и *Log Map* алгоритмов декодирования ТК по показателю количества элементарных операций.

Данная статья является продолжением работы [8], в которой получена оценка количества