

МЕТОДИ ТА ЗАСОБИ ПІДВИЩЕННЯ ДОСТОВІРНОСТІ ПРОЦЕСУ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У статті розглянуто принципи повторного тестування, нейромережну категорійну модель процесу повторного тестування програмного забезпечення (ПЗ) та нейромережний метод процесу повторного тестування ПЗ, а також розроблено метод оцінки достовірності процесу повторного тестування ПЗ. Виконано програмну реалізацію та дослідження штучної нейронної мережі (ШНМ) в пакеті Matlab. Описано структуру та функціонування програмних засобів оцінки достовірності для формування висновку щодо необхідності повторного тестування ПЗ.

In article retesting principles, neuronet categorical model of software retesting process and neuronet technique of software retesting process were viewed and software retesting process reliability evaluation technique was developed. Artificial neuron network (ANN) program realization and research were realized in Matlab. Structure and functioning of reliability evaluation software instruments for inferencing about software retesting necessity were described.

Ключові слова: повторне тестування, нейромережна категорійна модель, нейромережний метод процесу повторного тестування, метод оцінки достовірності.

Вступ

Підвищити достовірність тестування (ймовірність проведення вірного і вичерпного тестування ПЗ, під час якого не припускались помилки) і відповідно якість ПЗ можна не тільки шляхом тестування дефектів на етапах розроблення та налагодження, а й шляхом повторного тестування з метою виявлення прихованих помилок у програмах після основного тестування. Це підтверджується тим, що, як достовірність тестування, так і якість ПЗ залежать від кількості виявлених у ньому помилок, у тому числі і прихованих [1-7].

Для проведення оцінки ПЗ замовником з метою перевірки його відповідності встановленим вимогам та підвищення рівня якості і надійності ПЗ можна використовувати повторне тестування – тестування з метою прогнозування наявності прихованих помилок ПЗ та встановлення їх небезпечності та ступеня впливу на ПЗ, яке здійснюється після розроблення та налагодження ПЗ і є окремим технологічним процесом [8, 9].

Повторне тестування здійснюється на етапі вхідного контролю, який здійснює замовник, тобто допомагає замовнику оцінити якість програмного забезпечення та якість тестування програмного забезпечення, яке приймається, і вказує на наявність в ньому прихованих помилок [10, 11].

Принципи повторного тестування ПЗ

Прихованою помилкою назвемо будь-яку помилку ПЗ, що залишилась у програмному продукті після його діагностування у процесі розроблення та налагодження [8, 9]. Приховані помилки відрізняються від виявлених тим, що вони на певний момент часу після діагностування у процесі розроблення та налагодження ПЗ існують і ще не виявлені. При цьому помилки, зумовлені дефектом, певним чином впливають на систему через програмне забезпечення. Виявлення прихованих помилок проводиться після розроблення і налагодження ПЗ, де тестування програм здійснювалось як часткова технологічна операція під час цих процесів (етапів), під час окремого технологічного процесу повторного тестування.

Отже, головним напрямком дослідження оберемо методи і засоби підвищення достовірності процесу тестування ПЗ за рахунок виявлення прихованих помилок ПЗ на етапі повторного тестування.

Всі приховані помилки розподілимо за їх небезпечністю та ступенем впливу на ПЗ на незначні (НПП), помірні (ППП), серйозні (СПП) та катастрофічні (КПП) приховані помилки. Незначними прихованими помилками (НПП) програмного забезпечення вважатимемо такі, що не впливають на дії користувача, програмний продукт з їх наявністю придатний для використання. Помірними прихованими помилками (ППП) програмного забезпечення вважатимемо такі, що впливають на дії користувача. Програмний продукт з їх наявністю придатний для використання з частковою втратою функційності. Серйозними прихованими помилками (СПП) програмного забезпечення вважатимемо такі, що призводять до помилкових результатів, внаслідок чого програмний продукт непридатний до використання. Катастрофічними прихованими помилками (КПП) програмного забезпечення вважатимемо такі, що призводять до спотворення інформації (даних), внаслідок чого програмний продукт непридатний до використання і намагання його опрацювати призводить до відмови програмної системи. Незначним прихованим помилкам присвоїмо найнижчий рівень категорійності – перший. Помірним прихованим помилкам присвоїмо, відповідно, рівень 2; серйозним – рівень 3. Найвищим рівнем вважатимемо катастрофічний – рівень 4. Таким чином, рівнів прихованих помилок буде чотири.

На основі такої класифікації запропоновано принципи підвищення достовірності тестування ПЗ з виявленням прихованих помилок різних типів шляхом повторного тестування ПЗ з розподілом прихованих помилок на різні категорії і припущенням, що певна кількість помилок попередньої за серйозністю категорії призводить до появи окремих типів помилок наступної категорії, що забезпечило вибір та обґрунтування моделі процесу на базі ШНМ [8, 9].

На основі запропонованого підходу до розподілу прихованих помилок за категорійністю введемо

порог a_i допустимої кількості помилок і важливості помилок різних типів одного виду, при перевищенні якого необхідно здійснювати повторне тестування з метою виявлення прихованих помилок цього виду. Знаходження більшої кількості прихованих помилок під час повторного тестування підвищить, у свою чергу, достовірність процесу тестування взагалі, і відповідно якість програмного продукту.

Нейромережна категорійна модель процесу повторного тестування ПЗ

На основі принципів повторного тестування і розподілу прихованих помилок за категорійністю з врахуванням порогів допустимої кількості помилок і важливості помилок [8-11] розроблено математичну модель процесу повторного тестування, в основі якої лежить штучна нейронна мережа (ШНМ) типу прямонапрявленого перцептрону.

Вибір апарату ШНМ мотивований тим, що штучні нейронні мережі за рахунок можливості апроксимації нелінійних функцій дають можливість враховувати важливість (ваги) кожного типу неприхованих та прихованих помилок, пороги граничної кількості допустимих помилок кожної категорії, взаємний вплив прихованих помилок одних типів на помилки інших типів. Визначення вихідного функціоналу кожного із шарів ШНМ, що відповідають категорійності помилок, дає можливість оцінити сумарний вплив кожної категорії помилок на якість розробленого ПЗ, яке пройшло основне тестування, і зробити висновки щодо необхідності повторного тестування з метою виявлення і усунення помилок тієї чи іншої категорії. Важкоформалізованою задачею повторного тестування є визначення ваг впливу помилок різних типів однієї категорії [8, 10] на помилки іншої категорії, причиною яких є помилки попередніх категорій [8, 10]. Ця задача може бути вирішена за допомогою використання навченої ШНМ.

Зазначений підхід відображено узагальненою складною ШНМ, в якій структура багатозарового перцептрона типу MLP (multi-layer-perceptron) поєднується зі структурою простого перцептрона Розенблатта (одношарова ШНМ, де ваги першого шару незмінні, і зважені компоненти вхідного вектора на вході нейрона другого шару сумуються). Активаційною функцією нейронів вхідних та прихованих (асоціативних) шарів є функція гіперболічного тангенсу. Активаційною функцією нейронів ефекторних шарів є лінійна функція. Результати лінійної активаційної функції нейронів ефекторних шарів лежать в інтервалі [-1; 1]. Для оцінки достовірності з метою формування висновку про необхідність повторного тестування потрібні результати у вигляді 1 або 0 (відповідно є помилка i -го рівня категорійності чи немає), тому здійснюється перетворення (заокруглення) результатів наступним чином:

$$Y_i = \begin{cases} 1, & \text{якщо } Y_i > 0; \\ 0, & \text{якщо } Y_i \leq 0. \end{cases}$$

Структура цієї ШНМ представлена на рис. 1

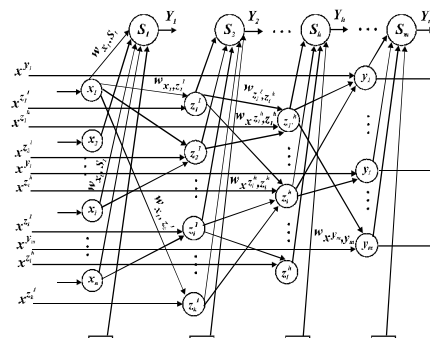


Рис. 1. Нейромережна категорійна модель процесу повторного тестування

Нейромережний метод процесу повторного тестування ПЗ [11, 12].

Початковими даними для реалізації повторного тестування є інформація про типи помилок (множина $P = \{p_k / k = 1..n\}$), виявлених під час основного тестування, та методи (множина $M = \{m_k / k = 1..n\}$) і операції, що були застосовані для їх виявлення (множина $O = \{o_k / k = 1..n\}$). Ця інформація береться із звітів про результати основного тестування. Оскільки основне тестування здійснює тестувальник, то на результати тестування можливий вплив суб'єктивного та людського факторів, що може як позитивно, так і негативно впливати на ефективність повторного тестування. Саме для зменшення зазначеного суб'єктивного фактора враховуються не тільки типи виявлених помилок, а й методи та операції тестування.

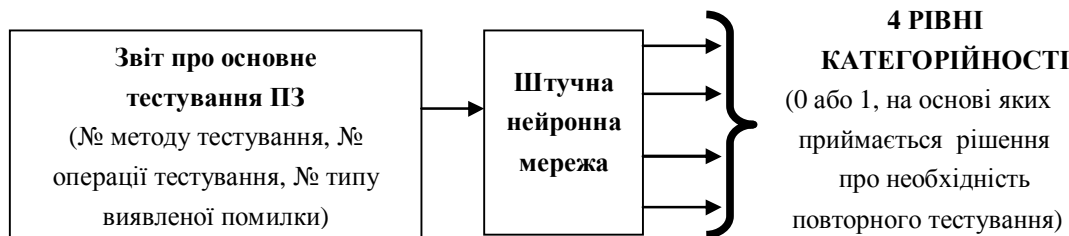


Рис. 2. Принцип застосування ШНМ для процесу повторного тестування

Вхідні дані для реалізації повторного тестування подаються у вигляді матриці $VD = \begin{pmatrix} m_1 & o_1 & p_1 \\ \cdot & \cdot & \cdot \\ m_i & o_i & p_i \\ \cdot & \cdot & \cdot \\ m_n & o_n & p_n \end{pmatrix}$,

де m_i, o_i, p_i – елементи множин M, O, P відповідно. Кожен елемент матриці VD , представлений у вигляді тексту, піддається перетворенню для представлення його у кількісному вигляді. Використовуючи

$$\text{матриці } MN = \begin{bmatrix} 1 & m_1 \\ \cdot & \cdot \\ i & m_i \\ \cdot & \cdot \\ s & m_s \end{bmatrix}, \text{ де } m_i - \text{ елемент множини } M, \quad ON = \begin{bmatrix} 1 & o_1 \\ \cdot & \cdot \\ i & o_i \\ \cdot & \cdot \\ v & o_v \end{bmatrix}, \text{ де } o_i - \text{ елемент множини } O, \quad PN = \begin{bmatrix} 1 & p_1 \\ \cdot & \cdot \\ i & p_i \\ \cdot & \cdot \\ z & p_z \end{bmatrix},$$

де p_i - елемент множини P , які представляють собою присвоєння номерів методам тестування, операціям тестування та типам виявлених помилок відповідно, $[i,1]$ -й елемент матриці VD , представлений у вигляді тексту, піддається перетворенню для представлення його у кількісній формі. Відбувається пошук елемента в другому стовпці матриці MN , одержується порядковий номер j рядка елемента. $[j,1]$ -й елемент матриці

$$MN \text{ заноситься в } [i,1]\text{-й елемент матриці } VDM = \begin{bmatrix} mn_1 & on_1 & pn_1 \\ \cdot & \cdot & \cdot \\ mn_i & on_i & pn_i \\ \cdot & \cdot & \cdot \\ mn_n & on_n & pn_n \end{bmatrix}, \text{ де } mn_i, on_i, pn_i - \text{ кількісне представлення}$$

значень елементів множин M, O, P відповідно.

Далі піддається перетворенню в кількісне представлення $[i,2]$ -й елемент матриці VD . Відбувається пошук елемента в другому стовпці матриці ON , одержується порядковий номер j рядка елемента. $[j,1]$ -й елемент матриці ON заноситься в $[i,2]$ -й елемент матриці VDM . Останнім піддається перетворенню в кількісне представлення $[i,3]$ -й елемент матриці VD . Відбувається пошук елемента в другому стовпці матриці PN , одержується порядковий номер j рядка елемента. $[j,1]$ -й елемент матриці PN заноситься в $[i,3]$ -й елемент матриці VDM .

Після одержання кількісного представлення значень кожного елемента матриці VD формується набір вхідних векторів для ШНМ. На вхід q_i подається 1, якщо використовувався відповідний для i -го рівня категорійності метод основного тестування (дана відповідність наведена в матриці

$$NMRK = \begin{bmatrix} nm_1 & rk_1 \\ nm_2 & rk_2 \\ nm_3 & rk_3 \\ nm_4 & rk_4 \end{bmatrix}, \text{ де } nm - \text{ номер методу основного тестування, } rk - \text{ рівень категорійності). На вхід } x'_i$$

подається номер i -ї операції основного тестування on_i ($[i,2]$ -й елемент матриці VDM), на вхід x_i подається номер i -го типу виявленої під час основного тестування помилки pn_i ($[i,3]$ -й елемент матриці VDM). На всі інші входи подається «0».

ШНМ опрацьовує набір вхідних векторів згідно методу вирішення задачі повторного тестування та

$$\text{видає матрицю вихідних векторів } VV = \begin{bmatrix} rk_{11} & rk_{12} & rk_{13} & rk_{14} \\ \cdot & \cdot & \cdot & \cdot \\ rk_{i1} & rk_{i2} & rk_{i3} & rk_{i4} \\ \cdot & \cdot & \cdot & \cdot \\ rk_{n1} & rk_{n2} & rk_{n3} & rk_{n4} \end{bmatrix}, \text{ де } i\text{-й рядок містить } i\text{-й вихідний вектор,}$$

елемент rk_{i1} містить значення «нуль» або «одиниця» для рівня категорійності з номером 1 i -го вихідного вектора, елемент rk_{i2} містить значення «нуль» або «одиниця» для рівня категорійності з номером 2 i -го вихідного вектора, елемент rk_{i3} містить значення «нуль» або «одиниця» для рівня категорійності з номером 3 i -го вихідного вектора, rk_{i4} містить значення «нуль» або «одиниця» для рівня категорійності з номером 4 i -го вихідного вектора. Вихідні вектори потрібно піддати перетворенню для одержання результатів у лінгвістичній формі. Для цього використовується матриця присвоєння рівнів категорійності типам

$$\text{прихованих помилок } RK = \begin{bmatrix} 1 & rk_1 \\ 2 & rk_2 \\ 3 & rk_3 \\ 4 & rk_4 \end{bmatrix}, \text{ де } rk_i - \text{ тип прихованих помилок. Перетворенню з кількісної в}$$

лінгвістичну форму піддається окремо кожен вихідний вектор, тобто окремо кожен рядок матриці VV . Для

перетворення i -го рядка в ньому відбувається пошук "одиниці", запам'ятовується номер стовпця h та знаходиться $[h,2]$ -й елемент матриці RK . Знайдений елемент rk_h є лінгвістичним представленням одержаного результату. Цей елемент заноситься в множину результатів $R = \{rk_k | k = 1..n\}$. На основі аналізу складу множини R робиться висновок про необхідність та тип повторного тестування.

Оцінка достовірності ідентифікації прихованих помилок програмного забезпечення

Із запропонованої моделі випливає, що при $Y_h > 0$ у програмі є помилки категорії, якій відповідає Y_h . І припустимо, що вони впливають на появу додаткових помилок наступного рівня категорійності. Нехай без врахування впливу на початку у програмі було p_x помилок першого рівня категорійності, p_{z^1} – помилок другого рівня категорійності, ..., $p_{z^{h-1}}$ – помилок h -го рівня категорійності. З врахуванням впливу помилок попереднього рівня категорійності на наступний помилку стало: p_x – першого рівня категорійності, $p_{z^1} + p_{z_i^{h-1}}$ – h -го рівня категорійності.

Припустимо, що ідентифікуються як помилки, що відповідають кожному рівню категорійності, так і помилки, що виникають внаслідок впливу кожного попереднього рівня на наступний.

За критерій достовірності процесу ідентифікації помилок ПЗ приймемо кількість виявлених помилок згідно запропонованої моделі. Тоді загальна кількість помилок N визначиться як сума всіх виявлених помилок:

$$N = p_x + (p_{z^1} + p_{x_i}) + \dots + (p_{z^h} + p_{z_i^{h-1}}) + \dots [10].$$

Достовірність D процесу ідентифікації прихованих помилок ПЗ шляхом повторного тестування дорівнюватиме [3]:

$$D = kn_1 p_x + kn_2 \frac{p_{z^1} + p_{x_i}}{p_{z^1}} + \dots + kn_h \frac{p_{z^h} + p_{z_i^{h-1}}}{p_{z^h}} + \dots,$$

де $KN = \{kn_h\}$ – множина коефіцієнтів нормування категорійності прихованих помилок.

Підвищення достовірності процесу ідентифікації прихованих помилок дорівнюватиме $\Delta D = 1 - \frac{D'}{D}$,

де D' – достовірність процесу ідентифікації прихованих помилок ПЗ без врахування впливу прихованих помилок кожного попереднього рівня категорійності на помилки наступного рівня категорійності [10].

Визначимо достовірність процесу виявлення прихованих помилок при повторному тестуванні для згаданих раніше чотирьох рівнів категорійності- НПП, ППП, СПП, КПП:

$$D = kn_1 \cdot p_x + kn_2 \cdot \frac{p_{z^1} + p_{x_i}}{p_{z^1}} + kn_3 \cdot \frac{p_{z^2} + p_{x_i}^{z^1}}{p_{z^2}} + kn_4 \cdot \frac{p_{z^3} + p_{x_i}^{z^2}}{p_{z^3}}.$$

Експертним шляхом (за результатами роботи 9 експертів Хмельницької філії софтверної організації Sitronics Telecom Solutions) присвоєно наступні значення коефіцієнтам нормування категорійності прихованих помилок: $kn_1 = 0,08$; $kn_2 = 0,22$; $kn_3 = 1,7$; $kn_4 = 8$.

Наприклад, з j -ї вибірки одержано наступні значення величин для визначення достовірності процесу повторного тестування (2-8-й стовпці табл. 1). Тоді за вище наведеними формулами обчислимо достовірність процесу виявлення прихованих помилок при повторному тестуванні (9-й стовпець табл. 1), достовірність процесу виявлення прихованих помилок при неврахуванні впливу попередніх рівнів категорійності (10-й стовпець табл. 1) та підвищення достовірності процесу ідентифікації прихованих помилок (11-й стовпець табл. 1)

Таблиця 1

Визначення достовірності повторного тестування

№ екс n.	p_x	p_{z^1}	p_{z^2}	p_{z^3}	p_{x_i}	$p_{x_i}^{z^1}$	$p_{x_i}^{z^2}$	D	D'	ΔD
1	28	16	10	4	6	4	2	16,92	12,16	0,28
2	32	20	10	6	8	6	2	16,26	12,48	0,23
3	46	28	14	8	10	6	4	18,40	13,60	0,26
4	50	30	18	10	12	8	2	16,36	13,92	0,15

Очевидно, що врахування впливу помилок попередніх рівнів категорійності підвищило достовірність процесу виявлення прихованих помилок на 15-28%. Графік підвищення достовірності відображений на рис. 3.

На цій гістограмі ряд 1 (чорний колір) показує достовірність D процесу ідентифікації прихованих помилок при повторному тестуванні, а ряд 2 (сірий колір) – відображає достовірність D' процесу ідентифікації прихованих помилок ПЗ без врахування впливу прихованих помилок кожного попереднього рівня категорійності на помилки наступного рівня категорійності.

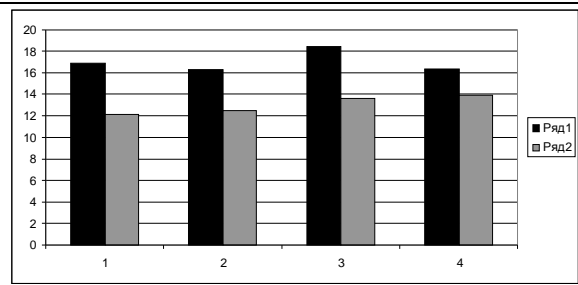


Рис. 3. Гістограма підвищення достовірності

Запропонована категорійна модель виявлення прихованих помилок підвищує якість програмного забезпечення і дає можливість оцінити достовірність процесу виявлення прихованих помилок.

Програмна реалізація та дослідження ШНМ в пакеті Matlab

В пакеті Matlab було виконано програмну реалізацію моделі ШНМ. Структурну схему імітаційної моделі ШНМ в пакеті Matlab представлено на рис. 4 [13].

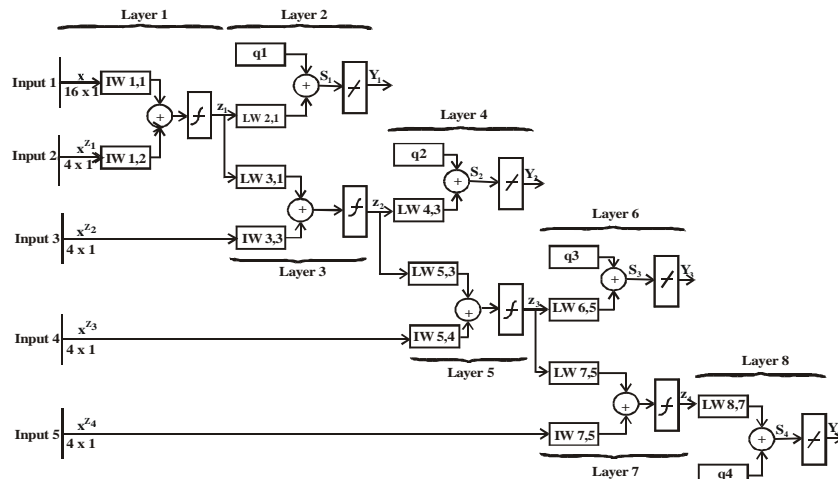


Рис. 4. Структурна схема імітаційної моделі ШНМ в пакеті Matlab

На кожен з входів $q_1 - q_4$ потрібно подати “одиницю”, тому що тестування здійснюється одним з методів тестування, які утворюються внаслідок об’єднання двох методів тестування під одним номером, що відображено в матриці присвоєння номерів методам тестування.

За статистикою [14] тестувальник тестує програму не більш як чотирма операціями одного методу тестування, тому на кожен з входів Input2 – Input5 можна подати не більше чотирьох номерів операцій тестування. На входи Input2 (x^{Z_1}), Input3 (x^{Z_2}), Input4 (x^{Z_3}), Input5 (x^{Z_4}) подаються номери операцій основного тестування ПЗ.

На вхід Input1 (x) подаються номери результатів операцій основного тестування ПЗ, тобто номери типів виявлених під час основного тестування помилок. Оскільки за статистикою [14] в програмі буває максимум 14-15 помилок, то на даний вхід можна подати не більше 16 типів помилок.

Кожен з виходів Y_i відповідає за один з чотирьох рівнів категорійності (Y_1 – перший рівень категорійності, Y_2 – другий рівень категорійності, Y_3 – третій рівень категорійності, Y_4 – четвертий рівень категорійності) і приймає значення „1”, якщо штучною нейронною мережею спрогнозовано наявність в програмі помилок i -го рівня категорійності, в протилежному випадку значення виходу Y_i становить „0”.

Для вибору алгоритму навчання ШНМ та критерію оцінки якості навчання ШНМ досліджувалась при навчанні вибіркою з 2250 векторів різними алгоритмами з використанням різних критеріїв оцінки якості навчання. В результаті проведеного аналізу було зроблено висновок, що похибка навчання змодельованої ШНМ залежить від критерію оцінки якості навчання та від форми представлення вхідних даних. Тому надалі використовується комбінований критерій якості навчання і масштабована навчальна вибірка. Мінімальна похибка, яку було досягнуто при використанні комбінованого критерію якості навчання та масштабованої навчальної вибірки з 2250 векторів, становить 0.448359. Меншої похибки навчання досягати неможливо і не потрібно, оскільки виходи мережі, які знаходяться в інтервалі [-1; 1] перетворюються для представлення цілими значеннями 1 або 0 (є чи немає помилки i -го рівня категорійності відповідно): $Y_i = \begin{cases} 1, & \text{якщо } Y_i > 0; \\ 0, & \text{якщо } Y_i \leq 0; \end{cases}$

В результаті аналізу імітаційної моделі ШНМ за часовим показником та за показником “кількість епох” найкращими є: алгоритм навчання CGB на основі метода спряженого градієнта з оберненим поширенням і рестартами в модифікації Пауела-Біеле, алгоритм навчання SCG, алгоритм навчання Флетчера-Рівса, алгоритм навчання Полака-Рібейри, пороговий алгоритм оберненого поширення помилки

Ргор. Оскільки алгоритм навчання CGB на основі метода спряженого градієнта з оберненим поширенням і рестартами в модифікації Пауела-Біеле, алгоритм навчання Флетчера-Рівса та алгоритм навчання Полака-Рібейри є модифікаціями метода спряженого градієнта, то для навчання обрано один з них – алгоритм навчання CGB на основі метода спряженого градієнта з оберненим поширенням і рестартами в модифікації Пауела-Біеле. Для тестування ШНМ було побудовано тестову вибірку з 200 векторів, яка також підлягала масштабуванню.

Програмні засоби для реалізації процесу повторного тестування ПЗ

На основі неймережного методу процесу повторного тестування розроблено структурну схему системи ідентифікації прихованих помилок ПЗ (рис. 5) [15].

Система ідентифікації прихованих помилок ПЗ складається з наступних компонентів [15]:

1) блок збору – передачі даних – підключає наданий користувачем файл з результатами основного тестування, представленими у вигляді журналу “Метод тестування– Операція тестування– Тип виявленої помилки”;

2) кодувальник – виконує перетворення вхідних даних з лінгвістичної форми представлення в кількісну форму за допомогою відповідних таблиць бази знань та формування вхідних векторів для модуля вирішувача. Здійснює заповнення бази знань вихідними даними, перетворення результуючих векторів вирішувача з кількісної в лінгвістичну форму за допомогою відповідної таблиці бази знань та передачу їх модулю опрацювання результатів роботи вирішувача;

3) база знань – містить таблиці присвоєння номерів методам і операціям основного тестування, типам виявлених помилок та присвоєння номерів рівням категорійності прихованих помилок; таблицю кількісного представлення вхідних даних, в якій містяться вхідні дані, перетворені кодувальником в кількісну форму; таблицю текстового представлення результуючих векторів вирішувача (ШНМ), в якій представлені результуючі вектори, перетворені кодувальником в лінгвістичну форму; таблиці відповідності методу основного тестування, операцій основного тестування, типів виявлених під час основного тестування помилок, відповідності між номером методів тестування ПЗ та рівнем категорійності прихованих помилок ПЗ, відповідності між операціями тестування ПЗ та рівнем категорійності прихованих помилок, на основі яких система формує висновок про метод, яким рекомендується проводити повторне тестування прикладного ПЗ, а також правила для формування висновку про необхідність повторного тестування;

4) вирішувач – штучна нейронна мережа, на входи якої подається інформація про методи і операції основного тестування та типи виявлених під час основного тестування помилок, а на виході одержується рівень категорійності прихованих помилок;

5) модуль опрацювання результатів роботи вирішувача – на основі правил та таблиці результатів роботи вирішувача, взятих з бази знань, генерує висновок про необхідність чи непотрібність повторного тестування, який передається користувачу через кодувальник, динамічний довідник та діалоговий компонент;

6) динамічний довідник – надає користувачу довідку про формат вхідного файлу, про відомі системі методи і операції основного тестування ПЗ, типи виявлених під час основного тестування помилки ПЗ, а також представляє в наглядній формі всі повідомлення будь-якого з компонентів системи;

7) діалоговий компонент – візуалізує повідомлення динамічного довідника та видає їх користувачу в зрозумілій для сприйняття формі.

Запропонована система ідентифікації прихованих помилок програмного забезпечення дозволяє користувачу, на основі звіту про результати основного тестування, одержати висновок про необхідність повторного тестування, а також про наявність у ПЗ прихованих помилок та їх рівень категорійності. Систему ідентифікації прихованих помилок ПЗ було реалізовано в Borland C++ Builder 6.0 із застосуванням системи управління базами даних Paradox7.

Діалогові вікна системи ідентифікації прихованих помилок ПЗ показано на рис. 6-8:

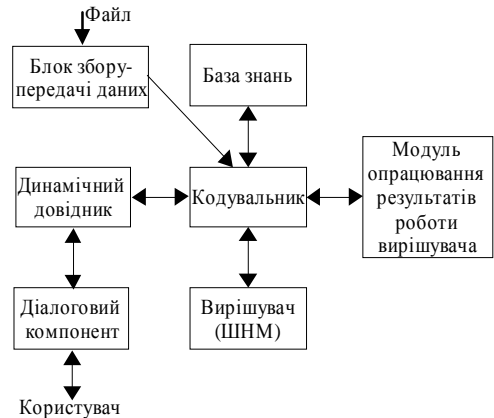


Рис. 5. Структурна схема системи ідентифікації прихованих помилок ПЗ

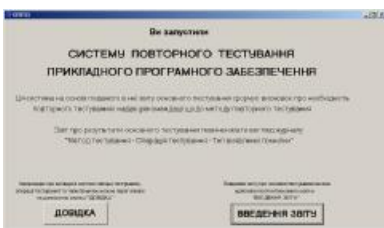


Рис. 6. Діалогове вікно системи повторного тестування прикладного програмного забезпечення



Рис. 7. Вікно довідника

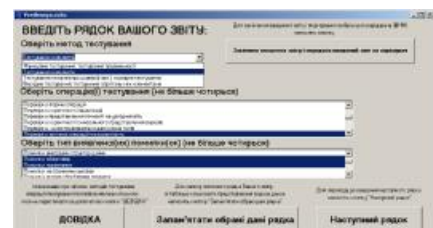


Рис. 8. Введення рядка звіту

При натисканні в діалоговому вікні (рис. 6) кнопки "ДОВІДКА" користувач може викликати довідник (рис. 7), а при натисканні кнопки "ВВЕДЕННЯ ЗВІТУ" – користувач має змогу подати по рядку звіт в систему ідентифікації прихованих помилок програмного забезпечення (рис. 8). Діалогове вікно введення рядка звіту (рис. 8) передбачає можливості: виклику довідки (рис. 7), запам'ятовування обраних даних рядка, переходу до введення наступного рядка звіту, а також закінчення введення звіту і передачі його на вирішувач.

Для формування висновку про необхідність повторного тестування модулем опрацювання результатів роботи вирішувача введемо величини порогових значень кількостей помилок кожного рівня категорійності, оскільки принципи підвищення достовірності тестування ПЗ з виявленням прихованих помилок різних типів шляхом повторного тестування ПЗ з розподілом прихованих помилок на різні категорії і припущенням, що певна кількість помилок попередньої за серйозністю категорії призводить до появи окремих типів помилок наступної категорії, є авторською і розроблялась саме в рамках цього дослідження. Для встановлення цих порогових значень проведемо дослідження кількості помилок програмного забезпечення, яке складалось з різної кількості операторів, з врахуванням і без врахування впливу помилок одного типу (категорії) на виникнення помилок наступного типу (категорії). Результати такого дослідження відображені в таблиці 2.

Таблиця 2

Кількість помилок програмного забезпечення з врахуванням і без врахування впливу помилок одного типу на виникнення помилок наступного типу

Кількість операторів в програмі	Кількість виявлених помилок без врахування взаємовпливу помилок					Реальна кількість помилок				
	100	500	1000	5000	10000	100	500	1000	5000	10000
Загальна кількість помилок	10	18	25	42	68	16	24	36	42	85
Незначні	8	14	19	31	51	8	14	19	31	51
Помірні	2	4	5	11	16	5	9	14	11	33
Серйозні	0	0	1	0	1	2	1	2	0	1
Катастрофічні	0	0	0	0	0	1	0	1	0	0

Після аналізу одержаних результатів дослідження зроблено евристичні оцінки, на основі яких введено наступні порогові значення, по перевищенню яких виникають приховані помилки більш високих рівнів категорійності:

1) якщо кількість помилок 4-го рівня категорійності (катастрофічних) перевищує 1, то повторне тестування необхідне за причини можливості відмови програмної системи;

2) якщо кількість помилок 3-го рівня категорійності (серйозних) перевищує 2, то повторне тестування необхідне за причини виникнення помилок вищого рівня категорійності;

3) якщо кількість помилок 2-го рівня категорійності (помірні) дорівнює або перевищує 50 % від загальної кількості виявлених під час основного тестування помилок, то повторне тестування необхідне за причини виникнення помилок вищих рівнів категорійності;

4) якщо кількість помилок 1-го рівня категорійності (незначні) дорівнює або перевищує 75 % від загальної кількості виявлених під час основного тестування помилок, то повторне тестування необхідне за причини виникнення помилок вищих рівнів категорійності.

Подамо на вхід системи звіт 1 про результати основного тестування (таблиця 3).

Таблиця 3

Звіт 1 про результати основного тестування

Метод тестування	Операція тестування	Тип виявленої помилки
Тестування елементів	Перевірка коректності кожної гілки програми (графу керування)	Помилки незалежних маршрутів програми
Тестування незалежних шляхів (гілок)	Перевірка форми операцій	Помилки обчислень
Вихідне тестування	Перевірка правильності розробки та функціонування драйверів	Помилки драйверів та їх розробки

Результат роботи системи визначення необхідності повторного тестування ПЗ виводиться у лінгвістичній формі, зрозумілій користувачу, програмісту і тестувальнику. Після аналізу звіту 1 система видає висновок: "Повторне тестування не потрібне" (рис. 9), оскільки жодне порогове значення не було перевищене.

Наступним на вхід системи подамо звіт 2 про результати основного тестування (таблиця 4).

Звіт 2 про результати основного тестування

Метод тестування	Операція тестування	Тип виявленої помилки
Тестування елементів	Перевірка коректності кожної гілки програми (графу керування)	Помилки незалежних маршрутів програми
Тестування елементів	Перевірка цілісності збережуваних даних	Помилки внутрішніх структур даних
Тестування незалежних шляхів (гілок)	Перевірка форми операцій	Помилки обчислень
Вихідне тестування	Перевірка правильності розробки та функціонування драйверів	Помилки драйверів та їх розробки

Після аналізу звіту 2 система видає висновок: "Повторне тестування потрібне" (рис. 10). Проводити його система рекомендує методом тестування елементів, оскільки висновок про необхідність повторного тестування зроблено по перевищенню порогового значення кількістю помилок 3-го рівня категорійності, а помилки 3-го рівня категорійності виявляє метод тестування елементів (ця відповідність наведена у таблиці відповідності між номером методів тестування ПЗ та рівнем категорійності прихованих помилок ПЗ бази знань).

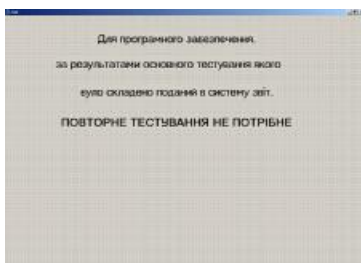


Рис. 9. Висновок системи повторного тестування після аналізу звіту 1



Рис. 10. Висновок системи повторного тестування після аналізу звіту 2

Висновки

Для оцінки небезпечності та ступеня впливу прихованих помилок на ПЗ введено 4 рівні категорійності. На основі запропонованого принципу повторного тестування розроблено нейромережну категорійну модель процесу повторного тестування ПЗ і нейромережний метод процесу повторного тестування ПЗ, дослідження яких дало можливість зробити висновок про підвищення достовірності процесу повторного тестування з врахуванням впливу помилок попередніх рівнів категорійності на 15-28%. На основі нейромережного методу процесу повторного тестування розроблено структуру та виконано програмну реалізацію системи визначення необхідності повторного тестування, яка на основі звіту про основне тестування ПЗ дає висновок про необхідність повторного тестування ПЗ на основі прогнозу наявності прихованих помилок в аналізованому ПЗ, тобто дає можливість замовнику оцінити якість ПЗ.

Література

1. Харченко В.С., Скляр В.В., Гордеев А.А. Верификация программного обеспечения. – Харьков: НАУ "ХАИ", 2006. – 132 с.
2. Локазюк В.М. Надійність, помилки і тестування програмного забезпечення комп'ютерних пристроїв та систем: Навчальний посібник. – Хмельницький: ТУП, 2003. – 74с
3. Липаев В.В. Качество программного обеспечения. – М.: Финансы и статистика, 1983. – 263 с.
4. Липаев В.В. Отладка сложных программ: Методы, средства, технология. – М.: "Энергоатомиздат", 1993. – 384 с.
5. Липаев В.В. Тестирование программ. – М.: "Радио и связь", 1986. – 411 с.
6. Myers G.J. The Art of Software Testing. – New York: John Wiley and Sons, 1979. – 312.
7. Сэм Канер, Джек Фолк, Енг Кек Нгуен. Тестирование программного обеспечения: Пер. с англ. – К.: Издательство "ДиаСофт", 2001. – 544 с.
8. Локазюк В.М., Пантелеєва (Говорущенко) Т.О. Категорійна модель процесу повторного тестування дефектів програмного забезпечення // Вісник Технологічного університету Поділля – Хмельницький: ТУП, 2004. – ч.1, т.1, с. 53 – 58
9. Lokazyuk V.M., Govoruschenko T.O. Category Model of Process of Repeated Software Testing // Proceedings of the Third IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. – Sofia, Bulgaria, 2005. – p. 241-245
10. Говорущенко Т.О. Підвищення достовірності тестування програмного забезпечення // Вісник Національного університету "Львівська політехніка" "Комп'ютерні науки та інформаційні технології" –

Львів: Видавництво Національного університету "Львівська політехніка", 2007 – с.186-196

11. V.Lokasyuk, O.Pomogova, T.Govorushchenko. Neural Nets Method for Estimation of the Software Retesting Necessity // Proceedings of the 2008 international workshop on Software Engineering in east and south Europe – Germany, Leipzig, 2008. – pp. 9-14. ISBN 978-1-60558-076-0. (<http://doi.acm.org/10/1145/1370868.1370871>)

12. Говорущенко Т.О. Система повторного тестування програмного забезпечення // Радіоелектронні і комп'ютерні системи – Харків: НАУ "ХАІ", 2005. – № 4, с.120-126

13. Говорущенко Т.О. Дослідження моделі вирішувача системи повторного тестування прикладного програмного забезпечення // Вісник Хмельницького національного університету – Хмельницький: ХНУ, 2007 – № 3, т.1, с.236-244

14. Калбертсон Р., Браун К., Кобб Г. Быстрое тестирование: Пер. с англ. – М.: Издательский дом "Вильямс", 2002. – 384 с.

15. Говорущенко Т.О. Реалізація та функціонування системи повторного тестування прикладного програмного забезпечення // Вісник Хмельницького національного університету – Хмельницький: ХНУ, 2007 – № 2, т.2, с. 113-120

Надійшла 9.12.2009 р.

УДК 389:638.011.54

В.Т. КОНДРАТОВ

Институт кибернетики им. В.М.Глушкова НАН Украины

ДОКАЗАТЕЛЬСТВА ФУНДАМЕНТАЛЬНОСТИ ФИЗИЧЕСКОЙ ТЕОРИИ ИЗБЫТОЧНЫХ ИЗМЕРЕНИЙ

В работе представлены доказательства фундаментальности физической теории избыточных измерений. Показано соответствие созданной теории всем требованиям, предъявляемым к физическим теориям со стороны ученых-физиков и ученых-философов.

In-process adduced solidity of physical theory of the surplus measuring. Accordance the created theory is rotined to all of requirements, to produced to the physical theories from the side of scientists-physicists and scientists-philosophers.

Ключевые слова: фундаментальность, избыточные измерения.

Введение

Вопросы фундаментальности физической теории волнуют многих ученых. Для исключения субъективных мнений и необоснованных предположений, нами было проанализировано отношение философов и физиков к данной проблеме. В результате было предложено и объединено в единую систему четырнадцать правил и признаков фундаментальности физических теорий [1]:

- 1) использование общенаучной методологии системного подхода¹ при создании новых теорий;
- 2) следование основным требованиям философии (науки всех наук) к теориям в части формулирования и использования основных структурных элементов теории, – общих и частных законов, явлений, категорий, принципов, методов, методологий, научных положений, основных правил, руководящих идей, парадигм, точек зрения и т.д., отличающих данную теорию от других;
- 3) глубокая проработка теории, использование структур и структурных элементов общих со структурами и структурными элементами фундаментальных физических теорий;
- 4) следование принципу соответствия Н.Бора или принципу преемственности знаний, их использование, обновление и обогащение с помощью новой теории;
- 5) инвариантные проявления сущности ФС или возможность установления инвариантов известных групп преобразований или группы преобразований, известные элементы которой неизменны;
- 6) наличие не одной, а совокупности целей (системы стратегических целей);
- 7) наличие доступного для понимания математического аппарата, описывающего основные законы, закономерности, состояния ФС, свойства, связи и воздействия разной физической природы и т.д.;
- 8) получение системного эффекта от использования данной теории;
- 9) возможность системной и целенаправленной (закономерной) структуризации данных с целью получения новых знаний;
- 10) возможность получения достоверной информации о типовых состояниях ФС и прогнозирование критических состояний;
- 11) обладание предсказательной силой;
- 12) получение новых положительных системных результатов исследований состояний статических или динамических систем;
- 13) развитие прикладных направлений научных исследований и становление новых дисциплин;

¹ В XXI-м веке методологию системного анализа целесообразно использовать только на начальных этапах развития теории.