

Твір наукового характеру "Нейромережний метод ідентифікації прихованих помилок програмного забезпечення"

Початковими даними для нейромережного методу ідентифікації прихованих помилок (НМПП) програмного забезпечення (ПЗ) є:

1) типи помилок (множина $P = \{p_k \mid k = 1..n\}$), виявлених під час основного тестування: помилки логічних умов; помилки незалежних маршрутів програми; помилки у гілках True, False для всіх логічних рішень; помилки в циклах у межах їх границь і діапазонів; помилки внутрішніх структур даних; помилки обчислень; помилки порівняння; помилки на граничних умовах; помилки шляхів оброблення помилок; помилки спрягань модуля; функції, описані у специфікації вимог до ПЗ, не відповідають одержаним; не всі елементи конфігурації системи враховані і розроблені правильно; не всі елементи конфігурації системи достатньо деталізовані відповідно до вимог замовника ПЗ; помилки в ситуаціях, коли на верхніх рівнях необхідні результати з нижніх рівнів, які ще не розроблені; некоректна робота "заглушок"; помилки драйверів та їх розробки; помилки об'єднання кластерів в загальну структуру; некоректні чи відсутні функції; помилки інтерфейсу; помилки у внутрішніх структурних даних; помилки характеристик; програма та її функціонування не відповідає специфікації та вимогам;

2) методи (множина $M = \{m_k \mid k = 1..n\}$), застосовані для виявлення помилок: функційне тестування, тестування правильності; тестування елементів; тестування незалежних шляхів (гілок), низхідне тестування; висхідне тестування, тестування спрягань між елементами;

3) операції (множина $O = \{o_k \mid k = 1..n\}$), застосовані для виявлення помилок: перевірка, чи виконує ПЗ очікувані функції; перевірка, чи виконує ПЗ поставлені вимоги; перевірка, чи виконують модулі ПЗ очікувані функції; перевірка, чи виконують модулі ПЗ поставлені вимоги; перевірка виконуваних функцій на основі наперед відомих функцій ПЗ; перевірка коректності сприйняття вхідних даних; перевірка співпадання вихідних результатів з наперед відомими еталонними результатами; перевірка зберігання цілісності інформації; перевірка відповідності системних характеристик ПЗ його специфікації; перевірка коректності кожної гілки програми (графу керування); перевірка правильності гілок True і False для всіх логічних рішень; перевірка виконання всіх циклів; аналіз правильності внутрішніх структур даних; перевірка логічних операторів на коректність, відсутність чи надлишковість; перевірка коректності логічної змінної; перевірка правильності використання логічних дужок; перевірка коректності оператора відношення; перевірка правильності арифметичних виразів; перевірка цілісності зберезуваних даних; перевірка пріоритету арифметичних операцій на правильність і зрозумілість; перевірка форми операцій; перевірка коректності ініціалізації; перевірка представлення точності на узгодженість; перевірка коректності символічного представлення виразів; перевірка, чи не порівнюються дані різних типів; перевірка логічних операцій на коректність; перевірка коректності пріоритетності; перевірка, чи не очікується еквівалентність в умовах, коли помилки точності роблять еквівалентність неможливою; перевірка коректності порівняння змінних; перевірка, чи правильно прикорочено цикл; перевірка, чи не виникла відмова на виході при відхиленні ітерації; перевірка зміни змінних циклу; перевірка правильності опрацювання n -го елемента n -елементного масиву; перевірка виконання m -ї ітерації циклу з m проходками; перевірка задання максимального або мінімального значення; перевірка зрозумілості повідомлення про помилку; перевірка відповідності тексту повідомлення про помилку виявленій помилці; перевірка, чи не відбувається реакція системних засобів реєстрації помилки ще до опрацювання її в модулі; перевірка коректності опрацювання виключної умови; перевірка, чи дозволяє опис помилки визначити її причину; перевірка даних на втрати при проходженні через спрягання; перевірка присутності необхідного посилання в модулі; перевірка, чи немає несприятливого впливу одного модуля на інший; перевірка утворення необхідної функції при об'єднанні під функцій; перевірка, чи не відбувається перехід окремих допустимих неточностей за допустиму межу при інтеграції; перевірка, чи не виникають проблемні ситуації при роботі з глобальними структурами даних; перевірка відповідності функцій, описаних у специфікації вимог до ПЗ, одержаним; перевірка, чи всі елементи конфігурації

системи враховані і розроблені правильно; перевірка, чи всі елементи конфігурації системи достатньо деталізовані відповідно до вимог замовника; перевірка коректності роботи “заглушок”; перевірка коректності роботи в ситуаціях, коли на верхніх рівнях необхідні результати з нижніх рівнів; перевірка правильності розробки та функціонування драйверів; перевірка коректності об’єднання кластерів в загальну структуру.

Ця інформація береться із звітів про результати основного тестування, які надаються тестувальником у вигляді журналу “Метод тестування – Операція, яка виконується під час тестування – Результат операції (тип виявленої помилки)”. Оскільки основне тестування здійснює тестувальник, то на результати тестування можливий вплив суб’єктивного та людського факторів (врахування «почерку» тестувальника), що може як позитивно, так і негативно впливати на ефективність повторного тестування. Саме для зменшення зазначеного суб’єктивного фактора враховуються не тільки кількість і типи виявлених помилок, а й методи та операції тестування.

Результатами опрацювання вхідних множин є:

1) прогноз наявності прихованих помилок (прихованою помилкою назвемо будь-яку помилку ПЗ, що залишилась у програмному продукті після його діагностування у процесі розроблення та налагодження);

2) рівень категорійності прихованих помилок (4 рівні категорійності - незначні приховані помилки, помірні приховані помилки, серйозні приховані помилки, катастрофічні приховані помилки), які залишились у програмному забезпеченні.

На основі цієї інформації робиться висновок про необхідність повторного тестування ПЗ (повторним тестуванням вважатимемо тестування з метою виявлення прихованих помилок, яке здійснюється після розроблення та налагодження ПЗ і є окремим технологічним процесом).

Отже, НМПП складається з наступних етапів:

1) підготовка множин типів виявлених помилок, методів та операцій основного тестування для подання їх на вхід штучної нейронної мережі (ШНМ);

2) перевірка, чи не виходять одержані значення типів помилок, методів та операцій тестування за межі діапазонів значень входів ШНМ;

3) представлення значень множин типів помилок, методів та операцій тестування у кількісному вигляді;

4) формування набору вхідних векторів для ШНМ;

5) опрацювання значень множин типів помилок, методів та операцій тестування штучною нейронною мережею;

6) перетворення вихідних векторів ШНМ у лінгвістичну форму;

7) аналіз результатів функціонування ШНМ;

8) формування висновку про необхідність повторного тестування.

Для формування прогнозу наявності прихованих помилок у ПЗ та ідентифікації їх рівня категорійності на основі звіту основного тестування ПЗ слід вирішити задачу визначення взаємозв'язків між значеннями множин типів помилок, методів і операцій тестування та прихованими помилками і їх рівнями категорійності. Одним із засобів, який дозволяє узагальнити інформацію та виявити залежності між вхідними і результуючими даними, є штучні нейронні мережі. Принцип застосування ШНМ у НМПП відображено на рис. 1.

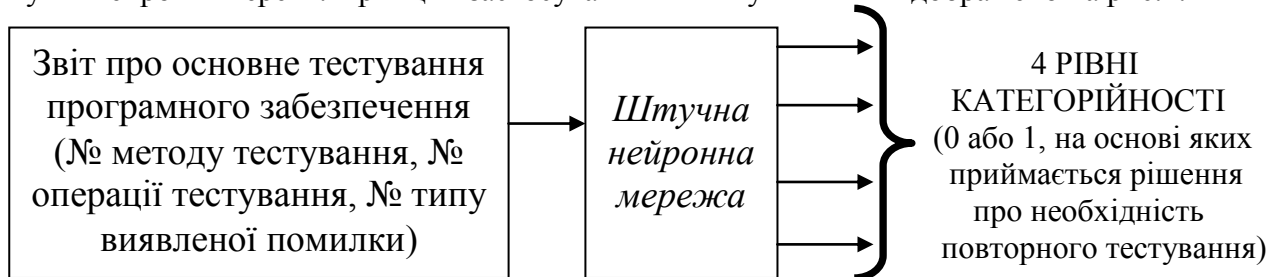


Рис.1. Принцип застосування ШНМ

Вхідні дані для реалізації повторного тестування подаються у вигляді матриці

$$VD = \begin{pmatrix} m_1 & o_1 & p_1 \\ \cdot & \cdot & \cdot \\ m_i & o_i & p_i \\ \cdot & \cdot & \cdot \\ m_n & o_n & p_n \end{pmatrix}, \text{ де } m_i, o_i, p_i - \text{ елементи множин } M, O, P \text{ відповідно.}$$

Кожен елемент матриці VD , представлений у вигляді тексту, піддається перетворенню

для представлення його у кількісному вигляді. Використовуючи матриці $MN = \begin{pmatrix} I & m_1 \\ \cdot & \cdot \\ i & m_i \\ \cdot & \cdot \\ s & m_s \end{pmatrix}$, де

$$m_i - \text{ елемент множини } M, \quad ON = \begin{pmatrix} I & o_1 \\ \cdot & \cdot \\ i & o_i \\ \cdot & \cdot \\ v & o_v \end{pmatrix}, \text{ де } o_i - \text{ елемент множини } O, \quad PN = \begin{pmatrix} I & p_1 \\ \cdot & \cdot \\ i & p_i \\ \cdot & \cdot \\ z & p_z \end{pmatrix}, \text{ де } p_i -$$

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 заноситься в $[i,1]$ -й елемент

матриці $VDM = \begin{pmatrix} mn_1 & on_1 & pn_1 \\ \cdot & \cdot & \cdot \\ mn_i & on_i & pn_i \\ \cdot & \cdot & \cdot \\ mn_n & on_n & pn_n \end{pmatrix}$, де mn_i, on_i, pn_i - 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 $NMRK = \begin{pmatrix} nm_1 & rk_1 \\ nm_2 & rk_2 \\ nm_3 & rk_3 \\ nm_4 & rk_4 \end{pmatrix}$, де nm - номер методу основного

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

містить i -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 прихованих помилок $RK = \begin{vmatrix} 1 & rk_1 \\ 2 & rk_2 \\ 3 & rk_3 \\ 4 & rk_4 \end{vmatrix}$, де rk_i - 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. Для опису правил формування висновку про необхідність повторного тестування введемо поріг $a_i \in A$ ($A = \{a_h / h = 1..4\}$, де a_h - поріг допустимої кількості помилок і важливості помилок різних типів одного виду, при перевищенні якого необхідно здійснювати повторне тестування з метою виявлення прихованих помилок цього виду, h - кількість типів порогів, що змінюється від 1 до 4, s - кількість рівнів категорійності прихованих помилок (тобто $s = 4$)), при перевищенні якого необхідно здійснювати повторне тестування з метою виявлення прихованих помилок цього виду.

Правила для формування висновку про необхідність повторного тестування та метод повторного тестування мають вигляд:

1) Якщо відношення сумарного значення помилок i -го рівня категорійності до загальної кількості виявлених під час основного тестування помилок перевищує поріг a_i , то повторне тестування здійснювати необхідно;

2) Якщо висновок про необхідність повторного тестування прийнятий по перевищенню порога кількістю помилок j -го рівня категорійності, то повторне тестування слід здійснювати методами і операціями, які виявляють помилки типу j .

На основі результатів роботи ШНМ формуємо множину $K = \{k_i / i = 1..4\}$, де k_i - сумарні значення помилок кожного з рівнів категорійності. На основі множини K та множини рівнів категорійності прихованих помилок $R = \{rk_k / k = 1..n\}$ формуємо множину $KR = \{kr_i / i = 1..4\}$ відношень $kr_i = \frac{k_i}{n}$.

Визначаємо порядок перегляду і застосування правил на основі отриманих результатів (метод пошуку). Процедура вибору зводиться до визначення напрямку пошуку і способу його здійснення. В даному дослідженні використовується метод здійснення пошуку в ширину в прямому напрямку, тобто спочатку аналізуються всі правила формування висновку про необхідність повторного тестування і за відомими фактами (елементи вектора KR) відшукуються заключення, яке з цих фактів слідує, і лише потім, якщо буде сформовано висновок про необхідність повторного тестування, аналізу піддаватимуться правила формування висновку про метод повторного тестування і за відомими фактами відшукається

висновок, який з цих фактів слідує.

Аналіз правил формування висновку про необхідність повторного тестування відбувається наступним чином. В множині правил типу “якщо-то” $PR = \{pr_h | h = 1..m\}$ шукаємо правило для кожного з елементів множини KR . Якщо значення елемента множини задовольняє умові лівої частини правила, то це правило заноситься в множину обраних правил $OPR = \{opr_y | y = 1..g\}$. Критерій вибору єдиного правила для множини OPR не актуальний, тому що у всіх правил однакова права частина (результат), в якій робиться висновок про необхідність повторного тестування. Отже, якщо кількість відібраних правил $g > 0$, то робиться висновок про те, що повторне тестування необхідне.

Після формування висновку про необхідність повторного тестування аналізуємо, поріг кількості помилок якого рівня категорійності був перевищений. Тоді, за правилом 2, повторне тестування рекомендовано проводити методами, які виявляють помилки саме цього рівня категорійності.

Структурна схема нейромережної складової методу ідентифікації прихованих помилок ПЗ представлена на рис.2.

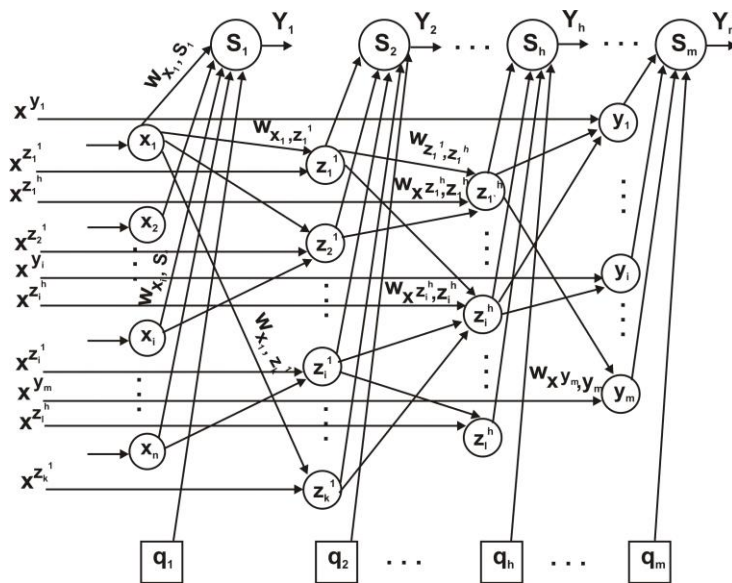


Рис.2. Нейромережна складова методу ідентифікації прихованих помилок ПЗ

На кожен з входів $q_1 - q_4$ потрібно подати “одиницю”, тому що тестування здійснюється одним з методів тестування.

На входи $x^{Z_1}, x^{Z_2}, x^{Z_3}, x^{Z_4}$ подаються номери операцій основного тестування ПЗ. За статистикою тестувальник тестує програму не більш як чотирма операціями одного методу тестування, тому на кожен з входів $x^{Z_1}, x^{Z_2}, x^{Z_3}, x^{Z_4}$ можна подати не більше чотирьох номерів операцій тестування.

На вхід x подаються номери результатів операцій основного тестування ПЗ, тобто номери типів виявлених під час основного тестування помилок. Оскільки за статистикою 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 (незначні помилки), Y_2 - другий рівень категорійності (помірні помилки), Y_3 - третій рівень категорійності (серйозні помилки), Y_4 - четвертий рівень категорійності (катастрофічні помилки). Вихід 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.

Розроблений нейромережний метод ідентифікації прихованих помилок програмного забезпечення дозволяє ідентифікувати наявність і рівень категорійності прихованих помилок та виявити необхідність повторного тестування і методи, якими слід його здійснювати.