

УДК 004.891.3: 004.3

Т.О. ГОВОРУЩЕНКО

Хмельницький національний університет, Україна

ПРОБЛЕМИ РЕАЛІЗАЦІЇ МЕТОДУ ІДЕНТИФІКАЦІЇ ПРИХОВАНИХ ПОМИЛОК ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ НЕЙРОМЕРЕЖНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

В статті висвітлюється суть категорійної моделі процесу повторного тестування програмного забезпечення і методу ідентифікації прихованих помилок програмного забезпечення на основі нейромережних інформаційних технологій, реалізація системи ідентифікації прихованих помилок програмного забезпечення, а також проблеми, які виникають під час реалізації методу.

метод ідентифікації прихованих помилок програмного забезпечення, категорійна модель процесу повторного тестування програмного забезпечення, проблеми реалізації методу ідентифікації прихованих помилок програмного забезпечення

Вступ

Діагностування програмного забезпечення (ПЗ) на різних етапах його життєвого циклу є одним з основних засобів, за допомогою котрого досягається висока надійність правильного функціонування обчислювальної техніки та вірність розв'язку прикладних задач.

Проведення аналізу програмного забезпечення як об'єкта діагностування дало можливість зробити наступні висновки:

1) розвиток і впровадження нових архітектур та апаратна складність комп'ютерів і систем, побудованих на їх базі, на сьогодні випереджає розроблення методів і засобів діагностування системного та прикладного програмного забезпечення цих систем;

2) існуючі діагностичні програми не завжди повністю враховують зростаючі вимоги до розроблення програм, за причини постійного ускладнення ПЗ та намагання користувачів розв'язувати за його допомогою важкоформалізовані та неформалізовані задачі;

3) низька якість окремих діагностичних програм знижує ефективність і достовірність використання існуючого ПЗ комп'ютерних систем.

Однією з основних складових діагностування ПЗ

є тестування як процес виявлення помилок у програмах. Його роль зростає в зв'язку з тим, що ПЗ сучасних комп'ютерних систем є досить складним і не може бути бездефектним. Причиною невиявлення помилок у розроблюваному ПЗ швидше за все слід вважати недосконалість тестів, а не бездефектність програми.

З аналізу методів тестування ПЗ стає зрозумілим, що жоден з них не є універсальним і має певні недоліки.

Виявлення недоліків тестування ПЗ, у тому числі ідентифікація прихованих помилок програмних продуктів, є актуальною задачею розвитку методів тестування ПЗ, що підвищує його достовірність.

Для вирішення проблеми підвищення достовірності процесу тестування програмного забезпечення розроблено концепцію підвищення достовірності тестування ПЗ, категорійну модель процесу повторного тестування ПЗ на базі нейромережних інформаційних технологій (НІТ), метод ідентифікації прихованих помилок ПЗ на основі ШНМ та систему ідентифікації прихованих помилок програмного забезпечення [1 – 3].

Розвинута концептуальна модель підвищення достовірності тестування ПЗ за рахунок виявлення прихованих помилок різних типів шляхом повтор-

ного 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, забезпечила вибір та обґрунтування категорійної моделі процесу повторного тестування на базі ШНМ.

1. Категорійна модель процесу повторного тестування програмного забезпечення

На основі запропонованої концепції підвищення достовірності тестування розроблено математичну модель процесу повторного тестування, поклавши в її основу штучну нейронну мережу (ШНМ) типу прямонапрявленого перцептрон [1].

Вибір апарату ШНМ мотивований тим, що штучні нейронні мережі 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.

Задачею повторного тестування є визначення ваг впливу помилок різних типів однієї категорії на помилки іншої категорії, причиною яких є помилки попередніх категорій. Ця задача може бути вирішена методом навчання ШНМ.

Відобразимо зазначений підхід узагальненою складною ШНМ, в якій структура багат шарового перцептрон типу MLP (multi-layer-perceptron) поєднується зі структурою простого перцептрон Розенблатта. Структура цієї ШНМ представлена на рис.1.

Визначення вагових матриць, функціоналів та активізаційні функції ШНМ описано в [1, 2]. Одержана категорійна модель процесу повторного тестування ПЗ на базі ШНМ відрізняється від відомих тим, що в ній враховується вплив прихованих помилок різних типів попередньої категорії на виникнення помилок наступної категорії, що дає можливість оцінити сумарний вплив помилок цієї категорії на

якість ПЗ і зробити висновок щодо необхідності повторного тестування ПЗ з метою усунення помилок попередніх, наступних чи розглядуваної категорій.

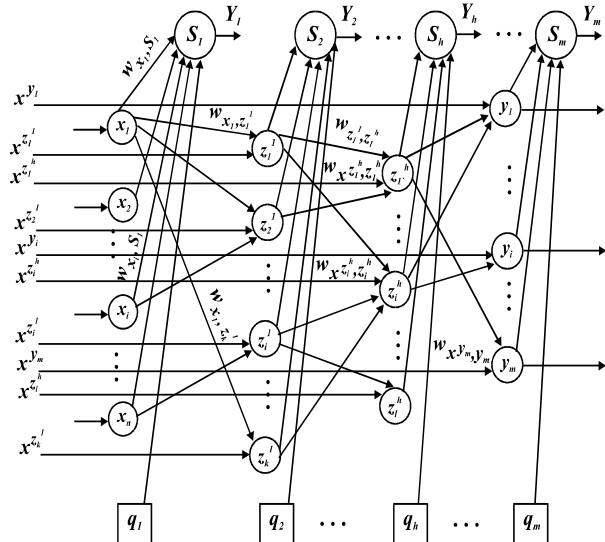


Рис. 1. Категорійна модель на базі ШНМ, що відображає зв'язок помилок ПЗ різних категорій

Було розроблено імітаційну модель та програмну реалізацію моделі ШНМ в пакеті Matlab [2].

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. Результати такого дослідження приведені в [2].

Аналізуючи графіки навчання і тестування ШНМ [2], зроблено висновок щодо найкращого алгоритму для навчання мережі (алгоритм навчання СGB на основі метода спряженого градієнта з оберненим поширенням і рестартами в модифікації Пауела-Біеле та його модифікації) та критерію оцінки якості навчання (комбінований).

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 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).

2. Метод ідентифікації прихованих помилок програмного забезпечення

Вхідними даними для реалізації повторного тестування є інформація про типи помилок, виявлених під час основного тестування, та методи і операції, що були застосовані для їх виявлення [2, 3]. 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)”. За допомогою матриць присвоєння номерів методам тестування, операціям тестування та типам виявлених помилок відбувається перетворення вхідного звіту про основне тестування з лінгвістичної в кількісну форму. Відбувається формування матриць вхідних даних у лінгвістичній та кількісній формах. Згідно методики застосування ШНМ матриця вхідних даних у кількісній формі подається на входи ШНМ. З вихідних даних ШНМ формується матриця вихідних векторів у кількісній формі. На виходах ШНМ є чотири рівні категорійності, значення 0 на яких означає відсутність помилки такого рівня категорійності, а значення 1 – наявність помилки такого рівня категорійності. Матриця вихідних векторів ШНМ перетворюється в лінгвістичну форму за допомогою матриці присвоєння рівнів категорійності типам прихованих помилок.

Суть одержаного методу ідентифікації прихованих помилок ПЗ на основі ШНМ полягає у виявленні множини типів прихованих помилок різних рівнів категорійності та аналізу цієї множини на предмет необхідності повторного тестування. Метод відрізняється від відомих тим, що вхідна інформація про результати основного тестування опрацьовується штучною нейронною мережею, яка відповідає моделі процесу повторного тестування. Запропонований метод ідентифікації прихованих помилок ПЗ на базі

ШНМ доцільно застосовувати на етапі вхідного контролю, який здійснює замовник, тобто цей метод допомагає оцінити замовнику якість тестування ПЗ, яке приймається, і вкаже на наявність в цьому ПЗ прихованих помилок. Цей метод застосовний, коли є звіт про результати основного тестування. Використання цього методу вирішує ключову практичну задачу, пов’язану з оцінкою замовником якості розроблення і тестування ПЗ, яке приймається, і ключову наукову задачу – підвищення достовірності процесу тестування програмних продуктів

3. Система ідентифікації прихованих помилок програмного забезпечення

Структурна схема системи ідентифікації прихованих помилок програмного забезпечення наведена в [3].

На блок підключення даних подається файл користувача з результатами основного тестування, представленими у вигляді журналу “Метод тестування – Операція тестування – Тип виявленої помилки”. Дані цього файлу передаються на кодувальник. Кодувальник здійснює перетворення вхідних даних з лінгвістичної форми в кількісну форму, заповнення бази знань вхідними даними та формування вхідних векторів вирішувача. База знань містить таблиці з вхідними даними системи, допоміжні таблиці, таблиці з правилами для формування висновку про необхідність та метод(и) повторного тестування. В якості вирішувача використовується штучна нейронна мережа (ШНМ) На входи ШНМ подається інформація про методи і операції основного тестування та типи виявлених під час основного тестування помилок, а на виході одержується рівень категорійності прихованих помилок. Вихідні дані вирішувача подаються на кодувальник, який здійснює заповнення бази знань результуючими даними, перетворення результуючих векторів вирішувача з кількісної в лінгвістичну форму та передачу їх у модуль опрацювання результатів роботи вирішувача

ча. Модуль опрацювання результатів роботи вирішувача на основі правил [3] генерує висновок про необхідність та метод повторного тестування, який передається через діалоговий компонент користувачу. Динамічний довідник надає користувачу під час роботи системи довідки про формат вхідного файлу, про відомі системі методи і операції основного тес-

тування ПЗ, типи виявлених під час основного тестування помилок ПЗ, а також передає користувачу всі повідомлення будь-якого з компонентів системи. Результатом роботи системи є висновок про необхідність повторного тестування та рекомендовані для повторного тестування метод(и) тестування ПЗ.

ВВЕДІТЬ РЯДОК ВАШОГО ЗВІТУ:

Оберіть метод тестування

- Тестування елементів
- Функційне тестування, тестування правильності
- Тестування елементів
- Тестування незалежних шляхів (гілок), низхідне тестування
- Висхідне тестування, тестування спрагань між елементами

Оберіть операцію(ї) тестування (не більше чотирьох)

- 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

Оберіть тип виявленої(их) помилки(ок) (не більше чотирьох)

- Помилки внутрішніх структур даних
- Помилки обчислень
- Помилки порівняння
- Помилки на граничних умовах
- Помилки шляхів оброблення помилок

Для закінчення введення звіту і формування вибірки для вирішувача (ШНМ) натисніть кнопку:

Закінчити введення звіту і передати введений звіт на вирішувач

Інформацію про зв'язок методів тестування, операцій тестування та типів виявлених помилок можна переглянути за допомогою кнопки "ДОВІДКА"

Для запису поточного рядка Вашого звіту в таблицю кількісного представлення вхідних даних натисніть кнопку "Запам'ятати обрані дані рядка"

Для переходу до введення наступного рядка натисніть кнопку "Наступний рядок"

ДОВІДКА **Запам'ятати обрані дані рядка** **Наступний рядок**

Рис. 2. Введення рядка звіту

Запропонована система ідентифікації прихованих помилок програмного забезпечення дозволяє користувачу, на основі звіту про результати основного тестування, одержати висновок про необхідність повторного тестування, а саме: про наявність у програмному забезпеченні прихованих помилок та про метод, яким рекомендується здійснювати повторне тестування.

В даній системі кодувальник виступає в якості інтерфейсу між користувачем і системою, а динамічний довідник – у якості інтерфейсу між користувачем та кодувальником. Систему ідентифікації прихованих помилок програмного забезпечення було реалізовано в Borland C++ Builder 6.0. Діалогові вікна даної системи відображено на рис. 2.

Оскільки уточнення цього підходу щодо опису прихованих помилок з введенням концепції катего-

рійності помилок проведено в дисертаційній роботі, то порогові значення кількостей помилок кожного рівня категорійності, по перевищенню яких приймається висновок про необхідність повторного тестування, в відомих літературних джерелах не описані. Для встановлення цих порогових значень проводились дослідження кількості помилок програмного забезпечення, яке складалось з різної кількості операторів, з врахуванням і без врахування впливу помилок одного типу (рівня категорійності) на виникнення помилок наступного типу (рівня категорійності). Після аналізу результатів досліджень було зроблено висновки щодо порогових значень кількостей помилок для всіх чотирьох рівнів категорійності, по перевищенню яких приймається висновок про необхідність повторного тестування. Ці висновки представлені в [3].

4. Проблеми реалізації методу ідентифікації прихованих помилок програмного забезпечення

До переваг запропонованого рішення в порівнянні з існуючими методами тестування програмного забезпечення слід віднести:

- немає обмежень на тип та розмір програмного забезпечення, а також на мову, якою написано програмне забезпечення; єдиною вимогою є наявність звіту про основне тестування досліджуваного програмного забезпечення;

- врахування взаємовпливів помилок однієї категорії на виникнення помилок наступних за серйозністю категорій призводить до підвищення достовірності на 15-28% [4];

- можливість проведення донавчання штучної нейронної мережі при недостатності врахованих при навчанні або при виникненні нових методів-операцій-типів помилок в процесі експлуатації системи ідентифікації прихованих помилок програмного забезпечення.

Разом з тим, під час реалізації методу виникло і ряд проблем, які вимагають внесення певних змін та часткового доопрацювання методу ідентифікації прихованих помилок програмного забезпечення.

Першою й найважливішою проблемою є вірна побудова навчальної вибірки для ШНМ, яка б враховувала всі ймовірні правильні комбінації методів-операцій-типів помилок, а також містила б невірні комбінації та граничні комбінації. Навчальна вибірка для ШНМ з 2250 векторів була побудована при обробці даних, наданих 9 експертами, в ролі яких виступали провідні програмісти ТОВ “СТУ-Електронікс” (м.Хмельницький). Найскладнішим завданням для експертів виявилось визначення рівнів категорійності тієї чи іншої помилки, які є вихідними даними ШНМ і які потрібно було оцінити для побудови навчальної вибірки ШНМ. Другим за складністю завданням для експертів стала розробка невірних та граничних комбінацій методів-операцій-типів помилок. Тому навчальна вибірка містить, в більшій мірі, правильні комбінації, а неправильні та граничні присутні в ній в дуже малій кількості. Для вирішення таких проблем бажано

було б брати дані від експертів з різних фірм, які розробляють програмне забезпечення різних типів, тоді навчальна вибірка буде бідше насиченою та більш вірною. Але знаходження великої кількості таких експертів теж є серйозною проблемою. До того ж, внести знання експертів у повному обсязі через навчальну вибірку практично неможливо.

Другою проблемою використання запропонованого методу, як показав досвід, стала проблема, що більшість фірм-розробників програмного забезпечення не складають звітів про проведення основного тестування розробленого програмного забезпечення, саме на основі аналізу яких система ідентифікації прихованих помилок пропонує висновок про необхідність та метод повторного тестування, тобто про наявність в програмному забезпеченні прихованих помилок.

Колективи, які й складають звіти, не завжди використовують класичні методи-операції-типи помилок, на основі яких побудована розглянута вище система. Тому при використанні звітів, які базуються на неklasичних методах-операціях-типах помилок, потрібно буде донавчати ШНМ на обробку саме нововведених колективом тестувальників певної фірми методів-операцій-типів помилок.

Третьою проблемою є проблема небажання розробником програмного забезпечення використовувати запроповану систему, оскільки її доцільно застосовувати на етапі вхідного контролю, який здійснює замовник, тобто цей метод допомагає оцінити замовнику якість розроблення і тестування ПЗ, яке приймається, і вказує на наявність в цьому ПЗ прихованих помилок. Раніше замовники намагались якомога краще, якісніше писати програмне забезпечення. Але ринок диктує інші вимоги. Замовник платить гроші і вимагає розробити програму дуже швидко. Ця ситуація вимагає від розробника з огляду на терміни виконання проектів поступатись якістю ПЗ. Зрозуміло, що розробнику не хочеться проводити повторне тестування розробленого та протестованого програмного забезпечення і не хочеться, щоб замовник знав про наявність в розробленому програмному забезпеченні прихованих помилок

та про недосконалість тестів, проведених розробником. Вирішенням такої проблеми може стати мета розробника створити дійсно якісний програмний продукт, для досягнення якої розробник не відмовиться проаналізувати звіт про основне тестування за допомогою запропонованої системи і виконати повторне тестування, якщо в цьому виникне необхідність (якщо система спрогнозує, що в програмному забезпеченні залишилися приховані помилки).

Наступною проблемою є відсутність чітких правил обрання структури ШНМ. В якості категорійної моделі було обрано узагальнену складну ШНМ, в якій структура багат шарового перцептрона типу MLP (multi-layer-perceptron) поєднується зі структурою простого перцептрона Розенблатта. Цієї структури достатньо для вирішення задач повторного тестування програмного забезпечення. Мотивів для використання нейромереж іншого типу я не знайшла, тому що жодна інша нейромережа не використовується для задач такого типу, як задача повторного тестування. Наприклад, радіальні базисні мережі (РБМ) вимагають великої кількості навчальних векторів, а для задачі повторного тестування важко зібрати велику кількість інформації по звітах основного тестування; мережі GRNN (регресійні) – один з видів РБМ, використовуються для аналізу числових рядів, в задачі повторного тестування зв'язків, аналогічних числовому ряду, немає; мережі PNN (імовірнісні) – один з видів РБМ, призначені для розв'язання імовірнісних задач, зокрема, задач класифікації, а в задачі повторного тестування і в навчальній вибірці немає жодних імовірнісних оцінок; карта Кохонена – призначена для кластеризації даних, мережі для класифікації вхідних векторів (LVQ) – використовуються для кластеризації і класифікації, ці задачі під час повторного тестування не вирішуються; мережі Елмана та Хопфілда – це мережі з динамічними оберненими зв'язками, орієнтовані на опрацювання динамічних моделей, що враховують передісторію процесів, які спостерігаються, задача ж повторного тестування не містить обернених зв'язків і не вимагає врахування передісторії. При використанні нейромережі

іншого типу для розв'язання задачі повторного тестування природа задачі буде штучно спотворюватись, в результаті чого нейромережа буде працювати невірно, результати роботи мережі будуть трактуватись невірно.

Висновки

В даній статті висвітлюється категорійна модель процесу повторного тестування програмного забезпечення, метод ідентифікації прихованих помилок програмного забезпечення на основі нейромережних інформаційних технологій, описується реалізація системи ідентифікації прихованих помилок програмного забезпечення. Основна увага в статті приділяється проблемам реалізації методу ідентифікації прихованих помилок програмного забезпечення та можливим шляхам їх вирішення.

Література

1. Локазюк В.М., Пантелєєва (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.
2. Говорущенко Т.О. Дослідження моделі вирішувача системи повторного тестування прикладного програмного забезпечення // Вісник ХНУ. – Хмельницький: ХНУ, 2007. – № 3, т. 1. – С. 236-244.
3. Говорущенко Т.О. Реалізація та функціонування системи повторного тестування прикладного програмного забезпечення // Вісник ХНУ. – Хмельницький: ХНУ, 2007. – № 2, т. 2. – С. 113-120.
4. Говорущенко Т.О. Оцінка ефективності виявлення прихованих помилок у програмному забезпеченні // Вісник Хмельницького національного університету. – Хмельницький, 2005. – Ч.1, т. 2. – С. 190-195.

Надійшла до редакції 22.01.2008

Рецензент: канд. техн. наук, Д.М. Медзатий, Хмельницький національний університет, Хмельницький.