

НЕЙРОМЕРЕЖНИЙ МЕТОД ОЦІНЮВАННЯ ТА ПРОГНОЗУВАННЯ СКЛАДНОСТІ ТА ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Вступ

Методологія Safety Case (Safety computer-aided software engineering) розвивається вже більше 20 років. Однією з основних задач методології Safety Case [1] є автоматизація формування: 1) профайлу вимог до програмного забезпечення (з врахуванням стандартів щодо розроблення програмного забезпечення (ПЗ), стандартів предметної галузі та вимог замовника) - аналізуються функційні, нефункційні та інверсні вимоги до ПЗ систем критичного застосування; досліджуються вимоги щодо надійності та безпеки ПЗ і оцінюється повнота всіх видів вимог; 2) профайлу результатів аналізу ПЗ - досліджуються та аналізуються результати метричного аналізу, програмний код та результати тестування ПЗ; 3) оцінки відповідності профайлу результатів та профайлу вимог - оцінюється відповідність одержаного ПЗ висунутим до нього вимогам.

Одним з основних засобів аналізу та оцінювання якості ПЗ є метричний аналіз. Метрика визначається як міра ступеня володіння властивістю, яка має числове значення [2]. На сьогодні, автоматизовано лише процеси збирання, реєстрації та обчислення метричної інформації. При аналізі методики опрацювання результатів метричного аналізу [2] виявлені наступні не вирішені питання: 1) відсутність єдиних стандартів на метрики, що призводить до суб'єктивізму вибору способів оцінювання якості; 2) складність інтерпретації значень метрик, обумовлена індивідуальними особливостями проектів та відсутністю еталонних значень метрик; 3) відсутність можливості порівняння принципово нового проекту з попередніми, що призводить до неможливості інтерпретації одержаних метрик для нового проекту; 4) основними параметрами при виборі варіанту реалізації ПЗ є вартість та тривалість розроблення і репутація фірми-проектувальника, але рішення, прийняті на основі цих параметрів, не завжди гарантують належну якість ПЗ.

З вищесказаного випливає потреба і актуальність наукових досліджень в галузі розроблення нових ефективних методів оцінки та прогнозування якості ПЗ. Наразі перспективними є інтелектуальні методи, зокрема, нейромережний метод оцінювання та прогнозування якості програмного забезпечення (НМОП) [3], [4].

1. Оцінювання результатів проектування та прогнозування характеристик якості програмного забезпечення

На сьогодні, ПЗ є визначальною складовою багатьох систем, серед яких системи критичного застосування, вбудовані та спеціалізовані системи різноманітного призначення. Для зазначених систем наявність помилок та низька якість ПЗ загрожує катастрофами, які призводять до людських жертв, екологічних катаклізмів, економічних втрат. Проблема виявлення та усунення помилок загострюється по мірі збільшення складності ПЗ. Розвиток сучасних технологій розроблення ПЗ вимагає динамічного розвитку засобів оцінки якості ПЗ, причому вже на етапі проектування (з точки зору економічної та часової доцільності).

Сучасна індустрія ПЗ характеризується високою конкуренцією. Для успішної роботи на цьому ринку софтверна компанія повинна розробляти, впроваджувати та супроводжувати ПЗ швидко, вкладаючись в термін та із задовільною якістю. Тому багато софтверних компаній вкладають кошти в модернізацію технологій розроблення ПЗ, пам'ятаючи про те, що таке вкладення коштів обов'язково окупається.

В якості базових для побудови інтелектуального методу оцінювання результатів проектування та прогнозування характеристик якості програмного забезпечення обрано 9 метрик етапу проектування ПЗ з точними значеннями та 15 метрик етапу проектування ПЗ з прогнозованими значеннями. Базові метрики етапу проектування ПЗ та алгоритми їх обчислення описані в [2]-[4].

Для визначення діапазонів значень використовуваних метрик етапу проектування введемо певні припущення та накладемо певні обмеження на проекти та ПЗ [5], які будуть аналізуватись за допомогою нейромережного методу оцінювання результатів проектування та прогнозування якості програм-

ного забезпечення (НМОП): 1) 100 змінних для розрахунків і виведення, 100 модифікованих або створених в модулі змінних, 100 керуючих змінних та 100 невикористовуваних в модулі ("паразитних") змінних для кожного модуля програми; 2) ПЗ має не більше 50 модулів; 3) максимальна кількість міжмодульних зв'язків становить 2450; 4) в кожному модулі є не більше 50 процедур, які оновлюють структуру даних, не більше 50 процедур, які читають інформацію зі структури даних, та не більше 50 процедур, які читають і оновлюють структуру даних; 5) кожен модуль реально одержує доступ до глобальної змінної стільки разів, скільки може одержати такий доступ; 6) ПЗ містить не більше 50000 рядків вихідного коду; 7) на постановку задачі потрібно 10% часу, на проектування - 35% часу, на програмування - 35% часу та на тестування, налагодження та перевірку якості - 20% часу; 7) під час проектування приблизно 25% часу витрачається на побудову та модифікацію моделей; 9) максимальна кількість помилок моделей та прототипів одного модуля не повинна перевищувати 100; 10) кількість унікальних операторів програми включно з іменами підпрограм не перевищує 25000, загальна кількість операндів програми не перевищує 50000, кількість унікальних операндів програми не перевищує 400; 11) кількість логічних операторів або операторів циклу не перевищує 50000; 12) кількість операторів програми не перевищує 50000; 13) кожна змінна модуля передається по його інтерфейсу; 14) максимальною очікуваною вартістю розроблення ПЗ в доларах США вважаємо кількість рядків вихідного коду, поділену навпіл; 15) 50% часу етапу тестування, налагодження та перевірки якості забирає перевірка якості ПЗ; 16) кількість зовнішніх входів кожного модуля - 49, кількість зовнішніх виходів кожного модуля - 49, кількість зовнішніх запитів до кожного модуля - 49; 17) кожен модуль має максимум 50 внутрішніх логічних файлів та використовує 50 зовнішніх логічних файлів.

Діапазони значень метрик етапу проектування програмного забезпечення наведені в таблиці 1.

Таблиця 1. Діапазони значень метрик етапу проектування ПЗ

№	Метрики етапу проектування з точними значеннями	Метрики етапу проектування з прогнозованими значеннями
1	Метрика Чепіна: -1, 0..32500	Очікувана LOC-оцінка: -1, 0..50000
2	Метрика Джилба (абсолютна модульна складність): -1, 0..2450	Метрика Холстеда: -1, 0..1562500
3	Метрика Мак-Клура: -1, 0..120050	Метрика Маккейба: -1, 0..2402
4	Метрика Кафура: -1, 0..497500	Метрика Джилба (відносна логічна складність): -1, 0..1
5	Метрика зв'язності: -1, 0..10	Прогнозована кількість операторів ПЗ: -1, 0..50000
6	Метрика зчеплення: -1, 0..9	Прогнозована оцінка складності інтерфейсів ПЗ: -1, 0..1
7	Метрика звертання до глобальних змінних: -1, 0..1	Прогнозований загальний час розроблення ПЗ: -1, 0..520 (в робочих днях)
8	Час модифікації моделей: -1, 0..46 (в робочих днях)	Час виконання робіт процесу проектування: -1, 0..182 (в робочих днях)
9	Кількість знайдених помилок моделей та прототипів: -1, 0..5000	Очікувана вартість розроблення ПЗ: -1, 0..200000 (грн)
10		Прогнозована вартість перевірки якості ПЗ: -1, 0..20000 (грн)
11		Прогнозована продуктивність розроблення ПЗ: -1, 0..5 (в хвилини на один рядок коду)
12		Прогнозовані витрати на реалізацію програмного коду: -1, 0..70000 (грн)
13		Прогнозований функційний розмір: -1, 0..2945
14		Прогнозована оцінка трудовитрат за моделлю Боєма: -1, 0..394 (людиномісяців)
15		Прогнозована оцінка тривалості проекту за моделлю Боєма: -1, 0..520 (в робочих днях)

В результаті опрацювання метрик, наведених в таблиці 1, за допомогою штучної нейронної мережі матимемо оцінку результатів проектування та прогноз характеристик складності та якості ПЗ, що розробляється на основі того чи іншого проекту.

2. Нейромережний метод оцінювання та прогнозування якості програмного забезпечення (НМОП)

НМОП дозволяє оцінити проект та спрогнозувати характеристики складності та якості розроблюваного ПЗ на основі точних або прогнозованих значень метрик складності та якості ПЗ етапу проектування. Для цього використовується штучна нейронна мережа (ШНМ), яка здійснює опрацювання та апроксимацію метрик.

Вхідними даними для ШНМ є множина метрик етапу проектування з точними значеннями та множина метрик етапу проектування з прогнозованими значеннями.

Результатом роботи ШНМ є 4 показники: 1) оцінка складності проекту; 2) оцінка якості проекту; 3) прогноз складності розроблюваного ПЗ; 4) оцінка якості розроблюваного ПЗ (рис.1).

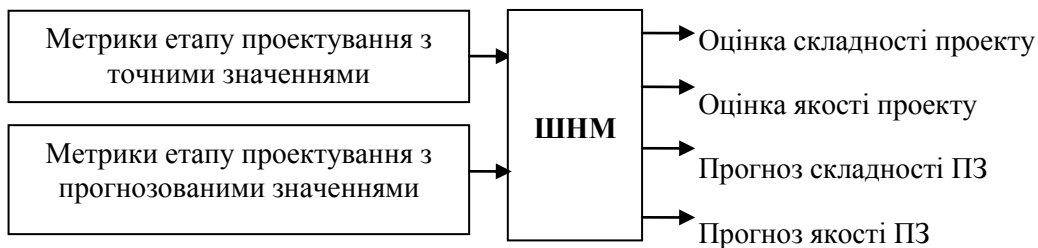


Рис.1. Концепція НМОП

ШНМ має 9 входів одного типу для кількісних значень метрик етапу проектування з точними значеннями та 15 входів іншого типу для кількісних значень метрик етапу проектування з прогнозованими значеннями. Якщо метрика не визначалась, то на відповідний вхід подається -1.

Оцінка складності проекту є значенням з діапазону $[0, 1]$, де 0 - метрики складності етапу проектування з точними значеннями не визначались; близько 0 - проект складний для реалізації, а 1 - проект простий для реалізації; оцінка якості проекту є значенням з діапазону $[0, 1]$, де 0 - метрики якості етапу проектування з точними значеннями не визначались; близько 0 - проект неякісний, а 1 - проект задовольняє вимоги замовника щодо якості; прогноз складності ПЗ є значенням з діапазону $[0, 1]$, де 0 - метрики складності етапу проектування з прогнозованими значеннями не визначались; близько 0 - розроблюване ПЗ буде мати суттєву складність, а 1 - розроблюване ПЗ очікується простим; прогноз якості ПЗ є значенням з діапазону $[0, 1]$, де 0 - метрики якості етапу проектування з прогнозованими значеннями не визначались; близько 0 - розроблюване ПЗ буде неякісним, а 1 - ПЗ очікується високої якості.

На основі аналізу 4-х одержаних результатів робиться висновок про якість і складність проекту та очікувану якість і складність розроблюваного за проектом програмного забезпечення.

3. Реалізація ШНМ

Для вибору архітектури для аналізу метрик етапу проектування ПЗ та прогнозу характеристик якості ПЗ було проведено аналіз відомих архітектур штучних нейронних мереж [4]. Регресійні радіальні базисні мережі використовуються для аналізу числових рядів. Імовірнісні радіальні базисні мережі призначені для розв'язання імовірнісних задач, зокрема, задач класифікації. Карта Кохонена призначена для кластеризації даних. Мережі для класифікації вхідних векторів використовуються для кластеризації і класифікації. Мережі Елмана та Хопфілда - це мережі з динамічними оберненими зв'язками, орієнтовані на опрацювання динамічних моделей, що враховують передісторію процесів. Оскільки задачі аналізу метрик та прогнозування характеристик якості ПЗ не притаманні властивості числового ряду, відсутні обернені зв'язки та немає потреби в класифікації та кластеризації даних, то для вирішення такої задачі оберемо багатосаровий перцептрон. При використанні нейромережі іншого типу для розв'язання цієї задачі її природа буде штучно спотворюватись, в результаті чого результати роботи ШНМ будуть невідповідними.

У якості ШНМ аналізу метрик та прогнозування характеристик якості ПЗ використано багатосаровий перцептрон (рис.2)

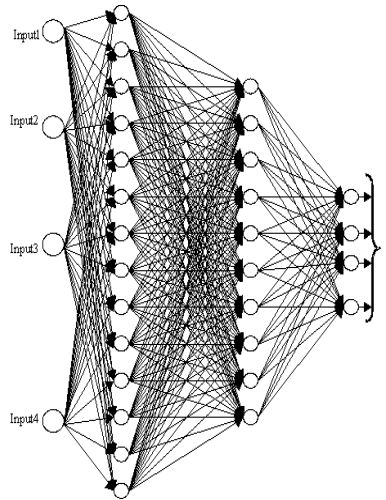


Рис.2. Архітектура ШНМ для аналізу метрик та прогнозування характеристик якості ПЗ

ШНМ реалізовано в пакеті Matlab. Представлення ШНМ подано в пакеті Simulink (рис.3, рис.4).

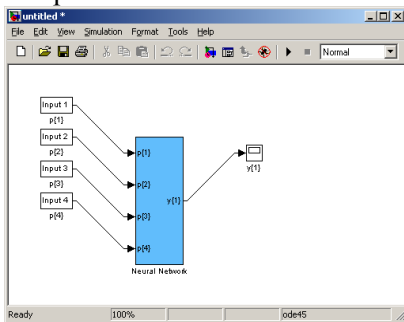


Рис.3. Архітектура ШНМ в пакеті Simulink

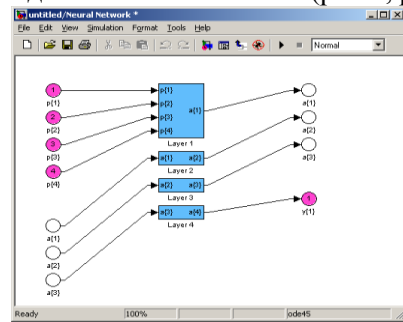


Рис.4. Структурна схема шарів ШНМ

4. Аналіз результатів навчання ШНМ

Для вибору алгоритму навчання проводилось навчання ШНМ навчальною вибіркою з 1935 векторів за різними алгоритмами навчання: GDA - метод градієнтного спуску з вибором параметра швидкості налагодження, LM - метод Левенберга-Марквардта, OSS - однокроковий алгоритм методу січної, RP - пороговий алгоритм оберненого поширення помилки.

В ході дослідження виявлено, що похибка навчання суттєво не відрізняється для різних алгоритмів навчання (приблизно 0,102197), тому вибір алгоритму навчання обумовлювався показниками "кількість епох" та "час навчання", за якими обрано алгоритм навчання OSS.

Тестування ШНМ здійснювалось на основі тестової вибірки з 324 векторів

Процес навчання і тестування за алгоритмом GDA зображено на рис. 5, за алгоритмом LM - на рис.6, за алгоритмом OSS - на рис. 7, за алгоритмом RP - на рис.8. Нижня лінія відображає навчання, а верхня лінія - тестування ШНМ.

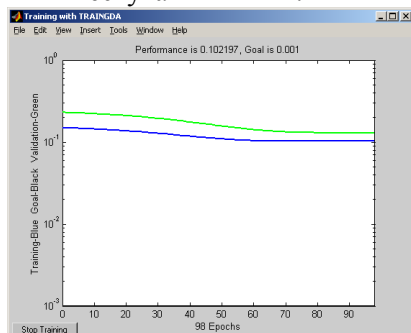


Рис.5. Навчання та тестування ШНМ за алгоритмом GDA

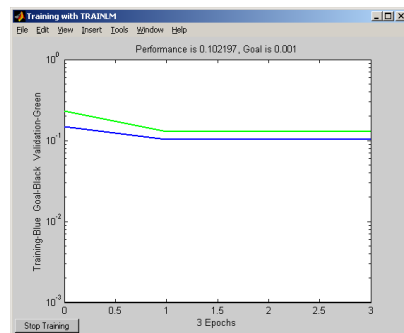


Рис.6. Навчання та тестування ШНМ за алгоритмом LM

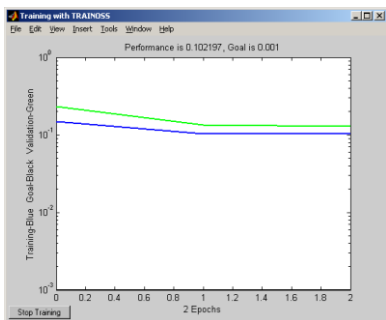


Рис.7. Навчання та тестування ШНМ за алгоритмом OSS

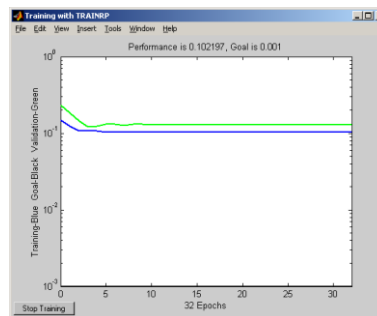


Рис.8. Навчання та тестування ШНМ за алгоритмом RP

Графіки навчання і тестування ШНМ показали, що мережа навчена з похибкою, яка обумовлюється недостатньою кількістю векторів навчальної вибірки.

Аналіз результатів ШНМ здійснювався на основі проектів, розроблених софтверною компанією "СТУ-Електронікс" м.Хмельницького.

На основі результатів ШНМ, вартості та часу розроблення здійснюється вибір варіанту реалізації проекту (табл. 2).

Таблиця 2. Критерії вибору версії проекту

№	Значення Y_1, Y_2	Значення Y_3, Y_4	Вартість розроблення	Час розроблення
1	$Y_1=0,51$ $Y_2=0,56$	$Y_3=0,60$ $Y_4=0,57$	87000 грн	200 робочих днів
2	$Y_1=0,32$ $Y_2=0,35$	$Y_3=0,38$ $Y_4=0,37$	89000 грн	210 робочих днів

Характеристики варіантів реалізації з таблиці 2 свідчать, що обидва варіанти мають приблизно однакові вартість та час розроблення, але суттєво різні оцінки складності та якості проекту і прогнози щодо складності та якості розроблюваного ПЗ, тому на основі лише вартості та часу розроблення софтверна організація може прийняти хибний висновок щодо вибору варіанту реалізації проекту. Саме оцінки, надані ШНМ, допоможуть зробити вірний вибір.

Висновки

З результатів аналізу методів метричної оцінки ПЗ впливає потреба і актуальність наукових досліджень в галузі оцінки та прогнозування якості ПЗ.

Основними параметрами при виборі варіанту реалізації ПЗ є вартість та тривалість розроблення і репутація фірми-проектувальника, але рішення, прийняті на основі цих параметрів, не завжди гарантують належну якість ПЗ. Прогнозовані оцінки характеристик якості розроблюваного ПЗ дають прогноз щодо якості реалізації конкретної версії проекту та дозволяють порівняти між собою різні версії проектів у ситуації, коли вартість і тривалість приблизно однакові. Запропонований підхід дає змогу прийняти мотивоване та обгрунтоване рішення щодо вибору проекту та його реалізації на основі не лише вартісних та часових характеристик, але й з врахуванням характеристик якості.

Література

1. A. Gordeyev, V. Kharchenko, A. Andrashov, B. Konorev, V. Sklyar, A. Boyarchuk. Case-Based Software Reliability Assessment by Fault Injection Unified Procedures // Proceedings of the 2008 international workshop on Software Engineering in east and south Europe – Germany, Leipzig, 2008. – pp. 1-8
2. Поморова О.В., Говорущенко Т.О. Аналіз методів та засобів оцінки якості програмних систем // Радіоелектронні і комп'ютерні системи – Харків: НАУ "ХАІ", 2009 – № 6, с.148-158
3. Поморова О.В., Говорущенко Т.О. Інтелектуальний метод оцінювання результатів проектування та прогнозування характеристик якості програмного забезпечення // Радіоелектронні і комп'ютерні системи – Харків: НАУ "ХАІ", 2010 – № 6, с.211-218
4. О.В.Поморова, Т.О.Говорущенко, С.Я.Тарасек. Аналіз та опрацювання метрик якості програмного забезпечення на етапі проектування // Вісник Хмельницького національного університету – Хмельницький: ХНУ, 2010. - № 1, с54-63
5. Брауде Э. Технология разработки программного обеспечения - СПб:Питер,2004 -655 с.