

ПРОГНОЗУВАННЯ ВІДМОВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З ВИКОРИСТАННЯМ НЕЙРОННОЇ МЕРЕЖІ НА ОСНОВІ РАДІАЛЬНО-БАЗИСНИХ ФУНКЦІЙ

© Яковина В. С., 2014

Використано радіально-базисну нейронну мережу для прогнозування відмов програмних продуктів. Досліджено вплив функції активації такої нейронної мережі на ефективність навчання та прогнозування відмов програмного забезпечення. Показано, що оптимальною функцією активації для цієї задачі є *Inverse Multiquadric* з 10 нейронами у вхідному шарі та 30 – у прихованому (квадрат коефіцієнта кореляції між прогнозованими та експериментальними даними становить 0,997, а середнє квадратичне відхилення – 14,4).

Ключові слова: надійність програмного забезпечення, часовий ряд, нейронна мережа типу RBF, функція активації

In this paper the radial-basis neural network was used for software failures prediction. The influence of activation function of the RBF neural net on the learning efficiency and software failures prediction is studied. It is shown that the optimal activation function is Inverse Multiquadric with 10 neurons in the input layer and 30 neurons in the hidden one (square of Pearson correlation coefficient is 0.997 and mean deviation is 14.4).

Key words: software reliability, time series, RBF neural network, activation function

Вступ

Програмне забезпечення (ПЗ) сучасних обчислювальних систем та інформаційно-управляючих систем повинно відповідати не тільки своїм функціональним параметрам. Під час його розроблення накладають певні обмеження на вартість і час, а також ставлять жорсткі вимоги до надійності, безпеки, готовності, цілісності та, відповідно, якості функціонування, оскільки відмова програмного забезпечення може спричинити катастрофічні наслідки. Багато організацій, що займаються створенням програмного забезпечення, до 50 % коштів, виділених на розроблення програм, витрачають на тестування.

За традиційного підходу до створення програмного забезпечення про його надійність починали говорити на завершальних стадіях його розроблення. Це пов'язано з тим, що можливість оцінювати надійність програмного продукту в цьому випадку з'являлася тільки після завершення його розроблення, тобто коли накопичувалися реальні дані для проведення статистичного аналізу та отримання відповідних оцінок. Такий підхід дає змогу оцінити надійність вже створеного програмного забезпечення, починаючи з фази тестування. Проте, якщо отримані оцінки не задовольняють поставлені вимоги, то виникає потреба повертатися на попередні етапи, вносити зміни, які іноді приводять до перепроєктування системи чи повторення інших стадій. Наслідком є значне зростання як фінансових, так і часових витрат. Тому виникає необхідність у забезпеченні надійності, починаючи із найперших фаз життєвого циклу програмного забезпечення. Для забезпечення надійності в умовах застосування сучасних технологій потрібно визначити достатність процесу тестування. А це, своєю чергою, дасть змогу зменшити суб'єктивний вплив на вибір кількості тестів за рахунок прогнозування кількості відмов програми та обґрунтовано вибирати необхідну кількість тестів, враховуючи обмеження на фінанси, час та вибраний рівень якості.

Огляд сучасного стану проблеми

Надійність, як наука про закономірності відмов технічних систем, почала формуватися в середині ХХ століття разом з появою складних радіоелектронних інформаційних систем, автоматизованих систем керування транспортом, енергопостачанням, технологічними процесами, систем військового призначення, які виконували надзвичайно складні та відповідальні функції. Відмова цих систем могла призвести до великих матеріально-технічних затрат або людських жертв. Науковою основою теорії надійності є теорія ймовірностей, математична статистика та теорія випадкових процесів. Основною особливістю надійності як науки є те, що вона має комплексний характер і тісно пов'язана зі спеціальними дисциплінами, які займаються завданнями організації контролю функціонування складних технічних систем, діагностування технічного стану, локалізації несправностей, відновлення працездатності, технічного обслуговування, раціонального вибору ресурсів тощо [1].

Ще однією особливістю надійності як науки є те, що в основу розвитку її методів та способів дослідження надійності поведінки складних технічних систем покладено математичне моделювання. Під математичною моделлю надійності технічної системи розуміють аналітично чи статистично представлений об'єкт, який так відображає властивості системи з погляду надійності, що його дослідження дає повну інформацію про надійність системи.

Розвиток методів моделювання надійності стимулюється підвищенням вимог до достовірності оцінок показників надійності складних систем, що, своєю чергою, вимагає підвищення вимог до достовірності оцінок показників надійності складних систем, а отже – підвищення ступеня адекватності їх надійнісних моделей та удосконалення методів їх побудови [1].

У дослідженнях надійності апаратного забезпечення механізм появи відмов часто розглядають як чорну скриньку, а предметом дослідження є процес відмови [2, 3]. У таких дослідженнях особливу увагу приділяють аналізу даних про відмови. Під час розгляду надійності ПЗ дослідників та інженерів насамперед цікавить механізм відмови [3, 4]. Переважна більшість моделей надійності ПЗ є аналітичними моделями, отриманими з припущень про механізми появи відмов. У таких дослідженнях основну увагу звертають на припущення моделі та інтерпретацію її параметрів. Щоб створити адекватну модель надійності ПЗ та у процесі прийняття обґрунтованих рішень на основі такої моделі необхідно глибоко розуміти процеси, методологію та технології створення, тестування ПЗ, етапи та механізми внесення помилок, їх типи, фактори зовнішнього середовища, які дають змогу верифікувати правильність припущень та обмежень моделі, визначити область адекватності моделі в середовищі користувача [3–7].

Найпоширеніші традиційні підходи до моделювання та оцінювання надійності програмного забезпечення використовують моделі, які опрацьовують результати тестування ПЗ та розглядають програмний продукт як чорну скриньку [3, 4]. Такі моделі містять апріорні припущення про розподіл таких випадкових величин, як час між помилками чи кількість помилок [4, 8].

Разом з тим, крім традиційних підходів, існують роботи з використання непараметричних моделей, які не потребують апріорних знань про функціональну залежність зростання надійності програмного забезпечення з часом. Один з таких підходів – використання штучних нейронних мереж [9], оскільки за останнє десятиліття доведено, що нейронні мережі можуть бути універсальним апроксиматором для будь-якої нелінійної неперервної функції з довільною точністю. Більшість нейронних мереж, що застосовують для моделювання надійності програмного забезпечення, можна поділити на два класи: перший використовує для моделювання такі архітектури нейронних мереж, як рекурентні, мережі Елмана тощо, другий клас моделює надійність ПЗ на основі НМ типу "множинний вхід із затримкою – одиничний вихід" [10–12].

Переважає більшість непараметричних моделей зводиться до прогнозування кількості відмов на деякому часовому інтервалі, що є задачею прогнозування часових рядів. До середини 80-х років минулого століття існувало декілька загальноновизнаних методів прогнозування часових рядів: економетричні, регресійні, методи Бокса–Дженкінса (ARIMA, ARMA) [13]. Однак в останні два десятиліття все більшого поширення набувають методи прогнозування часових рядів (зокрема надійності ПЗ) на основі нейронних мереж.

Для задач прогнозування надійності ПЗ досліджено різні нейромережеві моделі [9–12, 14–16]. Зокрема, нейронні мережі Wavelet демонструють ефективне прогнозування на початковій стадії тестування програмного забезпечення [15]. Рекурентні нейронні мережі, які потребують тривалого навчання, також ефективні у задачах прогнозування надійності програмного забезпечення [16].

Разом з тим, одним з нових класів нейронних мереж є мережі на основі радіально-базисних функцій (RBF), які характеризуються великою швидкістю навчання та успішно використовуються для задач апроксимації невідомих функцій [17, 18]. Мережа RBF не містить рекурсії, має такі особливості: єдиний прихований шар нейронів, тільки нейрони прихованого шару мають нелінійну активаційну функцію, синаптичні ваги всіх нейронів прихованого шару дорівнюють одиниці [19].

Постановка задачі

У попередніх роботах розпочато дослідження оцінювання надійності за допомогою нейронних мереж [20]. Мета цієї роботи – дослідити вплив функції активації нейронної мережі на ефективність прогнозування кількості виявлених у ПЗ помилок, поданих у вигляді часового ряду. Оскільки задача прогнозування є випадком задачі регресії, для цього дослідження вибрано радіально-базисну нейронну мережу (RBF) та чотири найпоширеніші функції активації: Gaussian, Multiquadric, Inverse Multiquadric та Mexican Hat [21].

Опис проведених експериментів

Для експериментів використано програмну реалізацію нейронної мережі RBF, яка давала змогу змінювати основні параметри мережі: функцію активації, кількість нейронів вхідного та прихованого шарів, похибку та кількість епох навчання. Програмну реалізацію модуля нейромережі RBF створено за допомогою бібліотеки Encog [22]. Encog – це бібліотека нейронних мереж, яка містить класи для створення найрізноманітніших мереж, а також підтримку класів для нормалізації та обробки даних для цих нейронних мереж. Encog активно розробляється з 2008 р., доступна для Java, .Net і Silverlight. Загальну діаграму компонентів програмного засобу для прогнозування надійності програмного забезпечення з використанням нейромереж зображено на рис. 1.

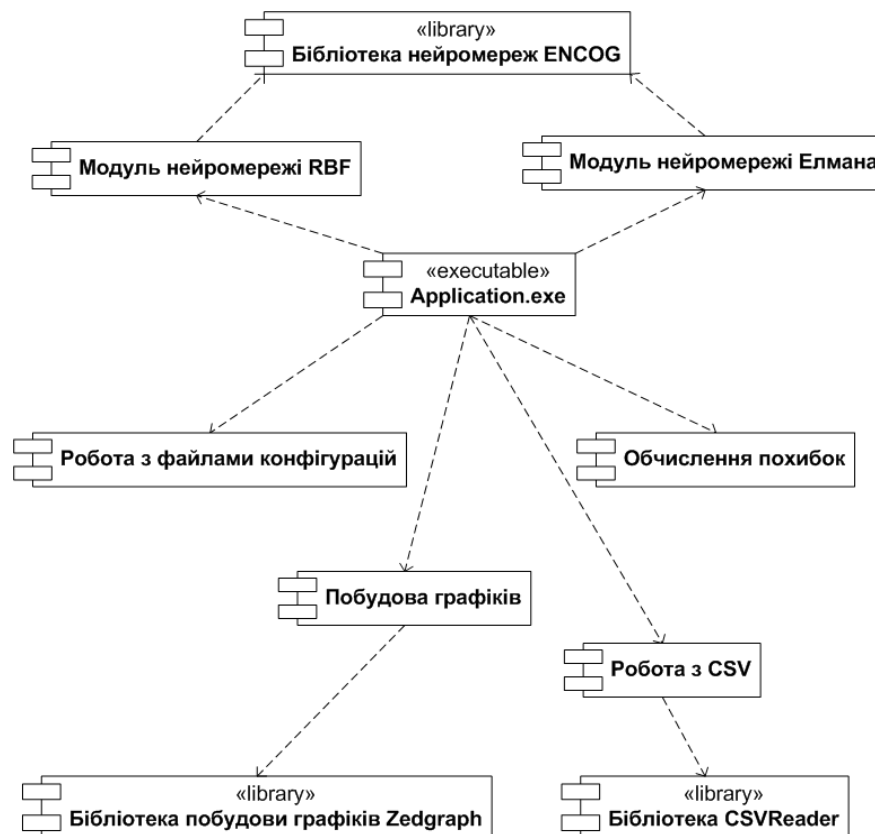


Рис. 1. Діаграма компонентів програмного продукту для прогнозування надійності ПЗ

Нехай задано n дискретних відліків – результатів тестування, а саме кількостей відмов $\{y(t_1), y(t_2), \dots, y(t_n)\}$ у послідовні моменти часу t_1, t_2, \dots, t_n . Тоді завдання прогнозування полягає в передбаченні значення $y(t_{n+1})$ – кількості відмов у деякий майбутній момент часу t_{n+1} . Для застосування нейронної мережі у задачі прогнозування часових рядів користувач вибирає довільний часовий ряд, що містить N відліків, і ділить його на три множини: навчальну, тестувальну і контрольну вибірки, які потім подаються на вхід мережі [19]. Результатом прогнозування є значення часового ряду в необхідний момент часу.

Як вхідні дані для навчання та прогнозування нейронної мережі використано результати тестування веб-браузера з відкритим вихідним кодом Chromium, загальнодоступні звіти за 870 днів тестування, під час якого виявлено близько 1000 помилок. Для навчання мережі інтервали, щодо яких є відомості про помилки, мають бути рівномірно розподілені, тому вхідні дані перед навчанням повинні проходити нормалізацію. Оскільки у тестуванні брала участь різна кількість тестувальників, результати тестування переводять у людино-дні.

Проведено дві серії експериментів, у кожному з яких входило по одному експерименту з кожною функцією активації. В першій серії конструювалась нейронна мережа з 10 нейронами вхідного шару та 30 нейронами прихованого шару. В другій серії експериментів нейронна мережа містила 30 нейронів у вхідному шарі та 10 – у прихованому. Навчання нейронної мережі здійснювалось до похибки 0,005 або ж до 5000 епох навчання, залежно від того, що досягалось раніше.

Проведені під час попередніх досліджень експерименти показали, що часовий ряд у вигляді залежності “кількість помилок”–“часовий інтервал” незадовільно прогнозується як нейронною мережею Елмана, так і RBF [20]. Натомість в цій роботі використано часовий ряд у вигляді кумулятивної кількості помилок: “кількість помилок, виявлених до часу t , – час t ”. Таке подання результатів тестування програмних продуктів істотно покращило якість прогнозування помилок у програмному продукті за допомогою нейронних мереж (як рекурентних, так і RBF).

Після такої попередньої обробки отриманий часовий ряд використовували для навчання нейронної мережі. Вхідні дані в цій роботі поділено на 150 рівномірних часових інтервалів. Перших 100 інтервалів використано для навчання нейронної мережі, а останніх 50 – для перевірки точності прогнозу (прогнозовані нейронною мережею результати порівнювали з цими контрольними значеннями).

Для оцінки ефективності прогнозування використано такі параметри: кількість епох навчання, яка характеризує швидкість навчання нейронної мережі; квадрат коефіцієнта кореляції Пірсона між передбаченими та експериментальними даними, який свідчить про лінійну залежність цих двох вибірок, та середнє квадратичне відхилення, що характеризує апроксимацію моделі до статистичних даних у рівномірній метриці та показує близькість передбачених та експериментальних даних. Результати статистичного опису ефективності прогнозування помилок ПЗ за допомогою нейронної мережі RBF під час обох серій експериментів наведено в табл. 1.

Аналіз отриманих результатів

Як видно з таблиці, для прогнозування найпридатніша функція активації Inverse Multiquadric, яка в обох конфігураціях показала найкращі статистичні показники якості прогнозування в поєднанні з не дуже великим часом навчання. Функція активації Gaussian може бути хорошою альтернативою для експрес-аналізу надійності, оскільки час її навчання є найменшим, а точність прогнозу майже не поступається найкращому випадку (за умови оптимально підібраних інших параметрів архітектури мережі). Натомість функція активації Mexican Hat виявилась непридатною для задачі прогнозування надійності ПЗ у вигляді кумулятивного часового ряду в обох конфігураціях нейронної мережі, і продемонструвала найбільшу схильність до “паралічу” під час навчання. Зауважимо також, що функція активації Gaussian виявила найбільшу стійкість до “паралічу” навчання, який часто виникав у другій серії експериментів, а у разі функції активації

Multiquadric також і в першій серії. Тому, на нашу думку, функцію Multiquadric, незважаючи на задовільні характеристики прогнозування, не слід використовувати для цієї задачі через схильність такої нейронної мережі до “паралічу” під час навчання.

Статистичні характеристики якості прогнозування надійності ПЗ нейронною мережею типу RBF з різними функціями активації

Функція активації та експеримент	Кількість епох навчання	Квадрат коефіцієнта кореляції	Середнє квадратичне відхилення
Gaussian I	105	0,997	15,4
Multiquadric I	432	0,995	22,3
Inverse Multiquadric I	293	0,997	14,4
Mexican Hat I	130	0,980	146,1
Gaussian II	186	0,986	113,9
Multiquadric II	4185	0,988	174,0
Inverse Multiquadric II	4149	0,994	63,2
Mexican Hat II	3060	0,946	360,9

Графіки залежності кількості прогнозованих та реальних помилок веб-браузера Chromium на інтервалах 100–150 наведено на рис. 2 і 3 для випадків використання функцій активації Inverse Multiquadric та Mexican Hat відповідно. Рис. 2 демонструє близькість прогнозованих та реальних точок та загальну якість прогнозу й ефективність використання RBF нейронних мереж для цієї задачі за умови оптимального вибору параметрів нейронної мережі.

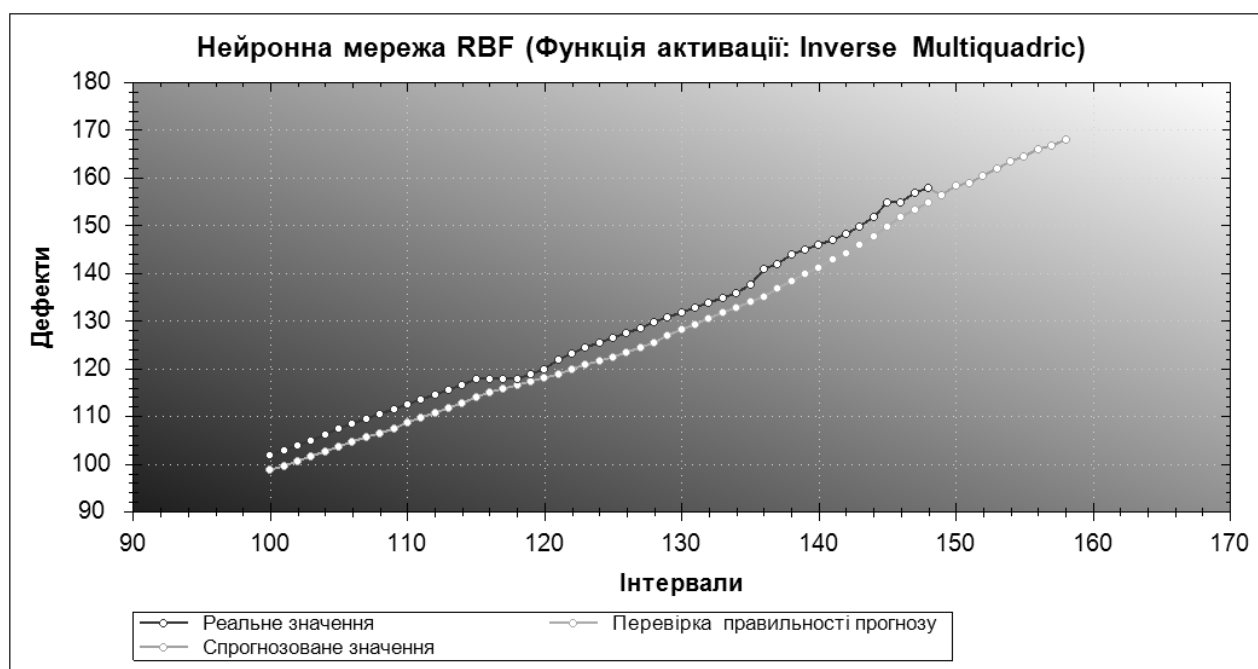


Рис. 2. Часові залежності прогнозованої та реальної кількості помилок веб-браузера Chromium (прогноз – нейронна мережа RBF з функцією активації Inverse Multiquadric)

Натомість з рис. 3 можна отримати наочне підтвердження даних табл. 1 про непридатність функції активації Mexican Hat для прогнозування надійності програмного забезпечення – починаючи з мінімального відхилення на сотому часовому інтервалі (який ще входив в навчальну вибірку), відхилення зростає і перевищує 30 % на 150-му часовому інтервалі.

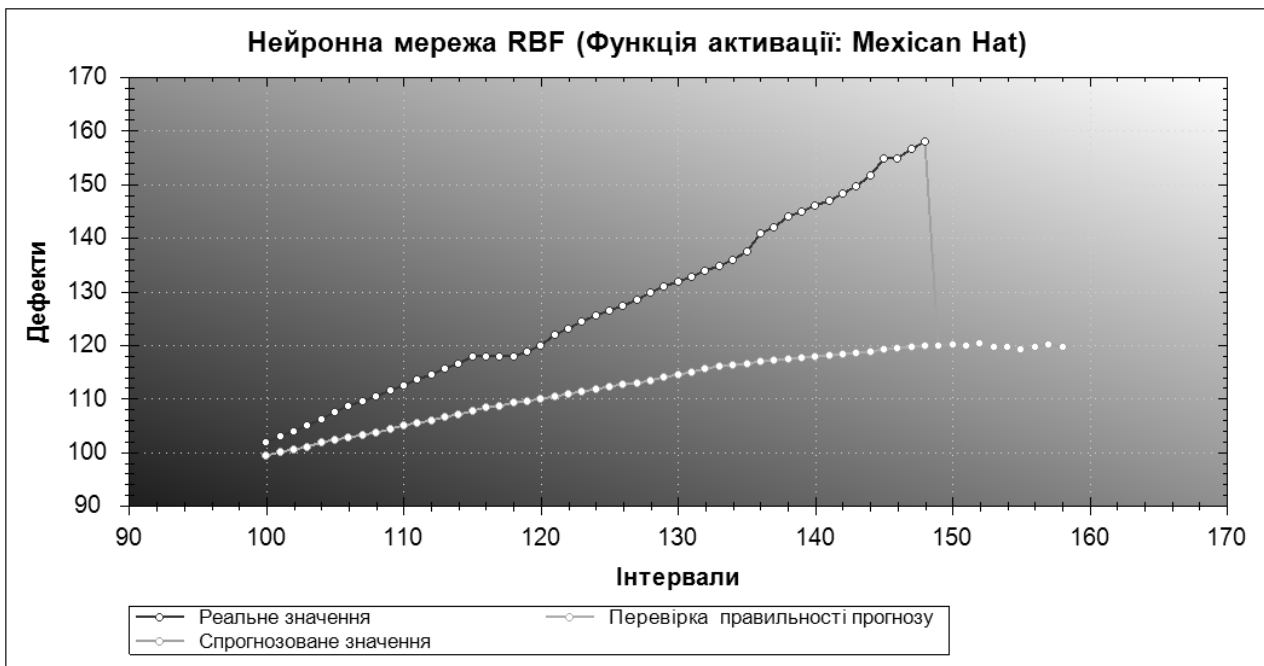


Рис. 3. Часові залежності прогнозованої та реальної кількості помилок веб-браузера Chromium (прогноз – нейронна мережа RBF з функцією активації Mexican Hat)

Висновки і перспективи подальших наукових розвідок

Робота стосується прогнозування відмов програмного забезпечення у вигляді часових рядів з використанням нейронної мережі на основі радіально-базисних функцій. Досліджено вплив функції активації нейронної мережі RBF на ефективність прогнозування відмов програмних продуктів. Для ефективного прогнозування вхідний часовий ряд слід подати в кумулятивному вигляді – значення відмов на i -му інтервалі часу є сумою кількості відмов на усіх попередніх інтервалах. Показано, що оптимальною з погляду швидкості навчання є функція Gaussian, тоді як стосовно точності прогнозування такою функцією є Inverse Multiquadric. Функція активації Mexican Hat непридатна для такої постановки задачі прогнозування відмов програмного забезпечення.

1. Бобало Ю. Я. Математичні моделі та методи аналізу надійності радіоелектронних, електротехнічних та програмних систем: монографія / Ю. Я. Бобало, Б. Ю. Волочій, О. Ю. Лозинський, Б. А. Мандзій, Л. Д. Озірковський, Д. В. Федасюк, С. В. Щербовських, В. С. Яковина. – Львів: Видавництво Львівської політехніки, 2013. – 300 с. 2. Половко А. М., Гуров С. В. Основы теории надежности. – СПб.: БХВ-Петербург, 2008. – 704 с. 3. Hoang Pham, Michelle Pham. Software Reliability Models for Critical Applications // EGG—2663 Technical Report (1991). Idaho National Engineering Laboratory, EG&G Idaho Inc. – 98 p. 4. Hoang Pham. System software reliability. – Springer-Verlag London Limited, 2006. – 440 p. 5. Tariq Hussain Sheakh, S.M.K. Quadri, and VijayPal Singh. A Study of Analytically Improving the Reliability of Software // International Journal of Research and Reviews in Computer Science, Vol. 3, No. 1, February 2012, P. 1404–1406. 6. Теїєр Т., Лунев М., Нельсон Э. Надежность программного обеспечения. Пер. с англ. – М.: Мир, 1981. – 323 с. 7. Лунев В.В. Надежность программных средств. – М.: СИНТЕГ, 1998. – 232 с. 8. Goel A. L. Software reliability models: assumptions, limitations, and applicability // IEEE Trans. on software engineering SE-11 (1985), No 12, 1411–1423. 9. Khoshgoftaar T. M., Szabo R. M. Predicting software quality, during testing, using neural network models: A comparative study // International Journal of Reliability, Quality and Safety Engineering 1 (1994), 303–319. 10. Su Y.-S., Huang C.-Y. Neural-network-based approaches for software reliability estimation using dynamic weighted combinational models // The Journal of Systems and Software 80 (2007) 606–615. 11. Cai K. Y., Cai L., Wang W. D., Yu Z. Y., Zhang D. On the neural network approach in software reliability modeling // The Journal of Systems and Software 55 (2001),

47–62. 12. Zheng J. *Predicting software reliability with neural network ensembles* // *Expert Systems with Applications* 36 (2009), 2116–2122. 13. *НейроПроект. Аналитические технологии для прогнозирования и анализа данных* [Электронный ресурс]: книга / Компания «НейроПроект» – Электрон. дан. (1 файл). – 2005. – 1 с. – Режим доступа: http://www.neuroproject.ru/forecasting_tutorial.php. – Назва з екрана. 14. Dohi T., Nishio Y., Osaki S. *Optimal software release scheduling based on artificial neural networks* // *Annals of Software Engineering*, Vol. 8 (1999). – P. 167–185. 15. Denghua Mei. *Early Software Reliability Prediction with Wavelet Networks Models* // *The 2007 International Conference on Intelligent Systems and Knowledge Engineering*, Oct. 15–16, 2007. 16. Karunanithi N., Whitley D., Malaiya Y. K. *Using neural networks in reliability prediction* // *IEEE Software*, Vol. 9 (1992), no. 4. – P. 53–59. 17. Salmeron M., Ortega J., Puntonet C. G. and Damas M. *Parallel Computation of an Adaptive Optimal RBF Network Predictor* // *LNCS*, Vol. 2687 (2003). P. 425–432. 18. Sing J. K., Thakur S., Basu D. K., Nasipuri M., Kundu M. *High-speed face recognition using self-adaptive radial basis function neural networks* // *Neural Computing and Applications*, Vol. 18 (2009). – P. 979–990. 19. Вороновский Г. К., Махотило К. В., Петрашев С. Н., Сергеев С. А. *Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности*. – Харків: Основа, 1997. – 112 с. 20. Yakovyna V., Synytska O., Kremen T., Smirnov V. *On the possibility of software reliability prediction using RBF neural network* // *Proc. VI-th Int. Sci. Tech. Conf. "Computer Science and Information Technologies" CSIT 2011, Lviv, Ukraine*, 39–42. 21. Paliwal M., Kumar U. A. *Neural networks and statistical techniques: A review of applications* // *Expert Systems with Applications* 36 (2009), 2–17. 22. Heaton Research. *Encog Java and DotNet Neural Network Framework. About the Encog project* [Электронный ресурс]: (проект) / Проект Heaton Research – Электр. дан. (1 файл). – 2005. – 1 с. – Режим доступа: <http://www.heatonresearch.com/encog>. – Назва з екрана.