© Popereshnyak S. Software for testing small-length bit sequences for randomness // Ukrainian Scientific Journal of Information Security, 2020, vol. 26, issue 2, pp. 80-86.

КРИПТОЛОГІЯ / CRYPTOLOGY

DOI: 10.18372/2225-5036.26.14867

ПРОГРАМНИЙ ЗАСІБ ДЛЯ ТЕСТУВАННЯ БІТОВОЇ ПОСЛІДОВНОСТІ МАЛОЇ ДОВЖИНИ НА ВИПАДКОВІСТЬ

Світлана Поперешняк

Київський національний університет імені Тараса Шевченка, Україна

ПОПЕРЕШНЯК Світлана Володимирівна, к.ф.-м.н., доцент

Рік та місце народження: 1980 рік, м. Кіровоград, Україна.
Освіта: Кіровоградський державний педагогічний університет імені Володимира Винниченка, 2002 рік.
Посада: доцент кафедри програмних систем і технологій факультету інформаційних технологій Київського національного університету імені Тараса Шевченка.
Наукові інтереси: програмна інженерія, автоматизація процесів виробництва, інформаційні технології, захист інформації, використання багатовимірних статистик для тестування бітової послідовності на випадковість.
Публікації: більше 100 наукових публікацій, серед яких навчальні посібники, наукові статті, матеріали та тези доповідей на конференціях.
E-mail: spopereshnyak@gmail.com.
Orcid ID: 0000-0002-0531-9809.

Анотація. *Данна стаття вивчає випадковість і найбільш відомі наборі тестів для її виявлення. Особлива увага приділяється статистичному дослідженню бітових послідовностей. Наявні набори тестів показують низьку гнучкість та універсальність у засобах знаходження прихованих шаблонів у даних невеликої довжини (до 100 біт). Для вирішення цієї проблеми запропоновано використовувати алгоритми на основі багатовимірних статистик. Дані алгоритми поєднують усі переваги статистичних методів та є єдиною альтернативою для аналізу послідовностей короткої та середньої довжини. У даній роботі розглянуто статичне тестування послідовностей з використанням багатовимірної статистики. У роботі наведені формули для тестування випадкових бітових послідовностей на випадковості, з використанням двовимірної або тривимірної статистика, яка може бути застосована для тестування коротких і середніх послідовностей. Для реалізації запропонованої методики було розроблено програмний засіб для тестування бітової послідовності на випадковість. Даний засіб включає в себе тести NIST, а також тести з використанням багатовимірної статистики, які добре себе зарекомендували при тестуванні бітової послідовності малої довжини. В результаті застосування розробленого засобу можливо проаналізувати бітову послідовність та вибирати якісну псевдовипадкову послідовність для використання в тій чи іншій предметної області.*

Ключові слова: *програмний засіб, бітова послідовність, тестування, багатовимірні статистики, випадкові послідовності, псевдовипадкова послідовність, статистичне тестування.*

Вступ

Більшість об'єктів і явищ, що нас оточують, мають випадкову природу. Для адекватного опису, вивчення і моделювання часто виявляється недостатньо детермінованих підходів, тому закономірно залучення стохастичних (тобто таких, що мають випадковий характер) методів вирішення різноманітних завдань. У зв'язку з цим випадкові числа, послідовності таких чисел і генератори, які їх виробляють знаходять все більше широке застосування в науці, техніці, зв'язку, різних інформаційних технологіях, а також у багатьох аспектах повсякденного життя [1-3].

Історично випадкові числа почали використовуватися для проведення вибіркових спостережень замість неперервних. Випадкові числа застосовуються при вирішенні складних обчислювальних задач і реалізації обчислювальних методів. Розвиток ЕОМ, з одного боку, розширило коло завдань, що використовують випадкові числа, а з іншого - пред'явило високі вимоги до якості їх генерації. Таким чином, випадкові числа відіграють важливу роль в інформатиці, розподілених обчисленнях, криптографії та інших областях.

Аналіз існуючих досліджень

Розглянемо найвідоміші набори тестів для перевірки бітових послідовностей. Варто зауважити, що деякі з методів випробувань в наборах збігаються, адже вони всі засновані на одному математичному підґрунті.

NIST Statistical Test Suite

NIST STS – специфікація та відповідна бібліотека на мові С, що були випущені Інститутом Стандартів та

Технологій США. Пакет складається з 15 тестів для аналізу бітових послідовностей, що були згенеровані ГПЧ або АГВЧ. Повний опис тестів доступний в [4].

Тести Diehard

Батарея статистичних тестів призначена для виміру якості ГПЧ та АГВЧ, що була створена Джорджом Марсалі у 1995 році. В основі більшості тестів лежить використання генератора для побудови послідовності відповідно да наданої специфікації і порівняння її характеристик з очікуваними від випадкової. Деякі з наведених випробувань можна виділити в групи за подібністю, а інші являють собою один тест. Більше інформації про тести можна знайти в [5, 6].

TestU01

Об'ємна бібліотека тестів на мові С, що включає реалізацію ГПЧ, тести та батареї тестів. Всі випробування що надаються, поділені в групи відповідно до модулів програми [7].

Аналіз тестів із зазначених статистичних пакетів дає можливість зробити висновок, що область перевірки випадковості далеко не є завершеною і потребує додаткового дослідження та покращення існуючих підходів. До проблем більшості тестів можна віднести:

– Випробування потребують послідовності великої довжини.

Наприклад, мінімальна рекомендована довжина послідовностей для NIST варіюється від 100 до 10^6, а деякі з тестів Diehard потребують по 100-200 тисяч біт. Звісно, якість результату покращується при збільшенні вибірки в будь-якому статистичному дослідженні, але не існує альтернативи для перевірки коротких послідовностей.

– Деякі з параметрів тестів неможливо змінити.

Це здебільшого стосується тестів Diehard, які потребують генерації послідовності фіксованої довжини відповідно до специфікації. Зміна параметрів має ключове значення для проведення якісного дослідження.

– Рішення про проходження тесту приймає тільки два значення (так/ні).

Результатами тесту повинні також бути точні та значущі числові значення. Це дозволить порівнювати результати різних тестів або одного тесту для різних послідовностей.

– Відсутність програмних пакетів для тестування.

Розробка пакетів для дослідження випадкових чисел без програмного забезпечення є доволі сумнівною роботою, адже область застосування повністю складається з інформаційних технологій.

Отже, в методах перевірки бітових послідовностей є достатньо проблем для вирішення та підходів для покращення. Особливий інтерес для дослідження складає відсутність тестів, що можуть дати адекватні результати на коротких послідовностях [8].

Метою даної роботи є покращення існуючих та впровадження нових методів тестування бітової послідовності на випадковість. Це включає розробку формального опису статистичних тестів та реалізацію відповідних програмних продуктів. Мета роботи – формальна та програмна реалізація існуючих тестів (на прикладі тестів NIST) та методів тестування заснованих на використанні багатовимірних статистик.

Основна частина дослідження

Специфіка тестів описаних пакетів є такою, що на основі вхідної послідовності бітів визначається статистика яка або є результатом, або використовується для його пошуку. Цей підхід враховує тільки одну характеристику послідовності при одному випробуванні. Багатовимірні статистики орієнтовані на декілька властивостей, що дозволяє більш точніше оцінити коротку послідовність, але має свої недоліки в тестуванні довгої через надмірно велику кількість варіантів комбінацій статистик.

Проблеми малих і великих вибірок відносяться до основних проблем, що виникають при практичному застосуванні методів аналізу даних. Будемо використовувати класифікацію вибірок за чисельністю наведену в [8], виходячи з вимог представлених в програмі критеріїв:

– дуже малі вибірки - від 5 до 12;
– малі вибірки - від 13 до 40;
– вибірки середньої чисельності - від 41 до 100;
– великі вибірки - від 101 і вище.

Відповідно до [1], генератори випадкових чисел мають тенденцію до створення великої кількості повторюваних шаблонів. Тести багатовимірних статистик, також, надають більш ефективні результати в перевірці шаблонів за рахунок оцінки декількох статистик одночасно.

Математично-статистичний аналіз послідовностей, як правило, відбувається в два етапи. Наведемо опис основних етапів:

1. Перший етап можна назвати підготовчим, він найбільш трудомісткий, тут виконується основна маса обчислень.

1.1. При допомозі дослідного генератора формуються випадкові послідовності (або вводяться задані послідовності).

1.2. Для кожної послідовності обчислюється статистика тесту. Якщо працює батарея тестів (проводиться відразу декілька тестів), то статистика за результатами виписується для кожного тесту.

1.3. Для кожної послідовності, що обчислюється ймовірність значущості.

1.4. Отриманні статистики та ймовірності значущості зберігаються.

2. На другому етапі проводиться обробка, отриманих результати.

2.1. Перевірка статистичної гіпотези

2.1.1. Формулювання нульової та альтернативної гіпотези.

2.1.2. При допомозі критеріїв погодження перевіряють гіпотези на відповідність розподілених статистичних даних і ймовірностей значущих гіпотетичних розподілів.

2.1.3. Визначається кількість послідовностей, які пройшли тест. Будується довірчий інтервал для останньої величини.

2.1.4. Порівняння долі послідовностей які попали в довірчий інтервал з рівнем значущості та прийняття рішення про проходження тестів.

2.2. Приймається рішення про те, чи можна вважати тест таким, що пройшов.

2.3. Якщо результати задовільні приймається рішення про завершення тесту, в противному разі переходимо до кроку 1.2.

2.4. Остаточні висновки.

Методи що представлені в роботі засновані на дослідженні кількості входжень двох- та трьох-бітових шаблонів в послідовність бітів. Тести на основі багатовимірних статистик в результаті виконання надають спільну вірогідність відповідної кількості шаблонів в послідовності заданої довжини. Той самий результат можна отримати за допомогою емпіричного підрахунку. Припустимо, що виконується розрахунок спільної вірогідності для всіх можливих значень $k_1 = \eta(11)$, $k_2 = \eta(000)$ та послідовності довжиною 3. Кількості входжень k_1 та k_2 до послідовності наведено в табл. 1.

Таблиця 1

Поява шаблонів в послідовності довжиною 3

Послідовність	**k_1**	**k_2**
000	0	1
001	0	0
010	0	0
011	1	0
100	0	0
101	0	0
110	1	0
111	2	0

Підрахувавши кількість появи для всіх можливих комбінацій k_1 та k_2, можна знайти відповідні вірогідності (табл. 2).

Таблиця 2

Входження шаблонів в послідовність довжиною 3

k_1	**k_2**	**Кількість**	**Вірогідність**
2	0	1	0,125
1	0	2	0,25
0	0	4	0,5
0	1	1	0,125

Емпіричним методом знайдено спільну вірогідність для заданої довжини і всіх можливих значень k. Цей підхід є доволі простим і наглядно показує для чого використовуються методи багатовимірних статистик, але не є ефективним (кількість послідовностей які необхідно перевірити при довжині 32 – 2^{32}). Випробування побудовані на формулах спільної вірогідності є більш доцільними як в математичному сенсі, так і в програмному.

Тести багатовимірних статистик відрізняються тільки шаблонами, на які перевіряється послідовність. Кожен метод отримує на вхід випадкову величину:

$\gamma 1, \gamma 2 \ldots, \gamma n$, де $\gamma i \in \{0, 1\}$, $i = 1,2\ldots,n$, $n > 0$.

Для даної величини визначається кількість специфічних шаблонів k1, k2 та k3 (якщо це визначено методом) і виконується обчислення за допомогою формули специфічної для методу.

Перший тест виконується, щоб знайти спільну вірогідність появи подій $k1 = \eta(tt^*)$ та $k_2 = \eta(t1t^*) + \eta(t0t^*)$, при $t \in \{0, 1\}$, $t^*=1 - t$:

$$P\left\{\eta\left(tt^*\right) = k_1, \eta\left(t1t^*\right) + \eta\left(t0t^*\right) = k_2\right\} = \sum_{m_1=k_1}^{n-k_1} p^{m_1} q^{m_0} \sum \prod_{i=0}^{1} C_{k_1}^{\delta_i} C_{m_i-k_1}^{k_1-\delta_i},$$

де n – довжина бітової послідовності, p – вірогідність появи t, q – вірогідність появи t* (q = 1 - p), $m_0 = n - m_1$, $\sum$ - сума по всім комбінаціям δ_0 та δ_1, таким, що: $\delta_0 + \delta_1 = 2k_1 + k_2$.

Другий метод тестування знаходить спільну вірогідність появи подій $k_1 = \eta(tt^*)$ та $k_2 = \eta(ttt^*)$:

$$P\left\{\eta\left(tt^*\right) = k_1, \eta\left(ttt^*\right) = k_2\right\} = \sum_{m_1=k_1}^{n-k_1} p^{m_1} q^{m_0} C_{k_1}^{k_2} C_{m_1-k_1}^{k_2} C_{m_0}^{k_1}.$$

Третій метод оцінює вірогідність появи шаблонів $k_1 = \eta(tt^*)$, $k_2 = \eta(t1t^*)$ та $k_3 = \eta(t0t^*)$:

$$P\left\{\eta\left(tt^*\right) = k_1, \eta\left(t1t^*\right) = k_2, \eta\left(t0t^*\right) = k_3\right\} = \sum_{m_1=k_1}^{n-k_1} p^{m_1} q^{m_0} C_{k_1}^{k_2} C_{k_1}^{k_3} C_{m_1-k_1}^{k_2} C_{m_0-k_1}^{k_3}.$$

За допомогою четвертого методу можна визначити вірогідність подій k1 = η(tt*) та k2 = η(ttt):

$$P\left\{\eta\left(tt^*\right) = k_1, \eta\left(ttt\right) = k_2\right\} = \sum_{m_1=k_1}^{n-k_1} p^{m_1} q^{m_0} C_{m_0}^{k_1} \times \sum_{i \in \{k_1, k_1+1\}} C_i^{m_1-k_2-i} Z\left(m_1 - i, m_1 - i - k_2\right),$$

де $Z(a,b) = \begin{cases} C_{a-1}^{b-1}, \text{ якщо } a \geq b \geq 0; \\ 1, \text{ якщо } a = b = 0; \\ 0, \text{ в іншому випадку} \end{cases}$.

Враховуючи, що обробка вхідних запитів є найважливішою задачею серверу, розглянемо типовий сценарій за допомогою діаграми діяльності (рис. 1). Як можна бачити, процес є доволі складним, і містить багато етапів на яких можуть виникнути критичні та помилкові ситуації.

Згідно з специфікацією NIST [1], та описаних методів багатовимірних статистик [9-11] створено бібліотеку що надає користувачам два інтерфейси для виконання відповідних статистичних тестів в мові програмування Java. Ціллю програмного засобу є забезпечення користувачів можливістю тестування бітових послідовностей на випадковість за допомогою графічного інтерфейсу. Він повинен забезпечити високий рівень зручності використання: надавати свободу дій, врахувати можливість помилок, повідомляти інформацію про стан системи та містити довідкові матеріали.

Відповідно до основного функціоналу який повинен забезпечувати додаток, між задачами, що виконує веб-додаток та задачами серверу є чіткий розподіл. Перший працює незалежно більшу частину часу, і викликає другого тільки коли не може виконати поставлену задачу самостійно.

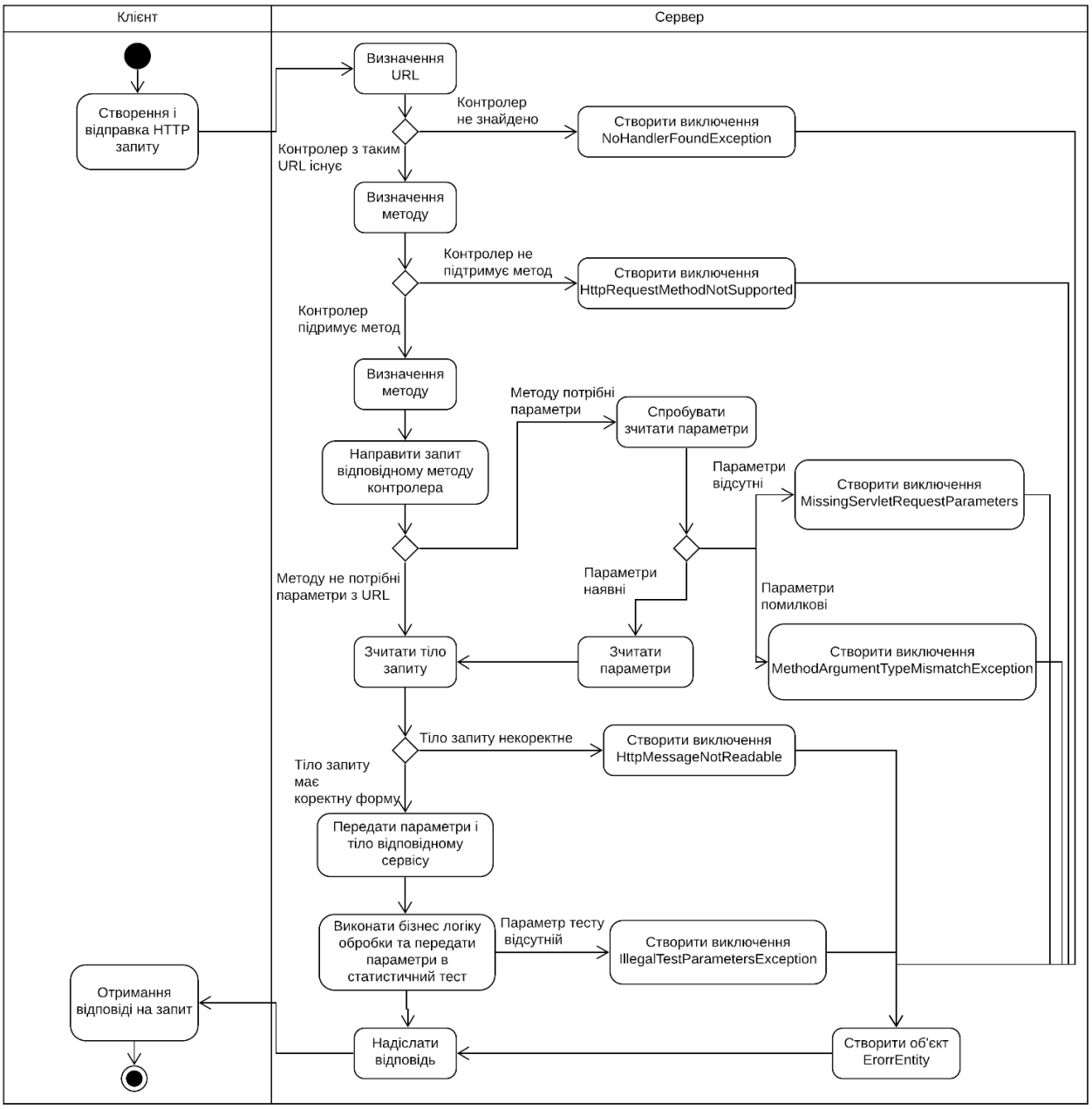

Рис. 1. Діаграма діяльності

Застосунок складається з однієї «сторінки», відповідно до принципів створення front end за допомогою React. Так, верхній та нижній та нижній колонтитули залишаються незмінними незалежно від URL, а зміст який знаходиться посередині завжди змінюється в залежності поточної адреси. Верхня навігаційна панель надає можливість перейти на 3 сторінки, зміст яких описано в табл. 3.

Таблиця 3

Зміст сторінок навігаційної панелі

Назва сторінки	Опис сторінки
Головна сторінка	Загальна інформація про тести та бітові послідовності і про призначення програмного продукту
Сторінка тестів NIST	Містить 16 сторінок що відповідають окремим тестам NIST та сторінку з комплексним тестом
Сторінка тестів багатовимірних статистик	Містить 9 сторінок що відповідають окремим тестам багатовимірних статистик та сторінку з комплексним тестом

Всі сторінки з тестами містять додаткове меню в якому можна обрати один окремий, або комплексний тест. На сторінці останнього, користувачу надається можливість ввести вхідні параметри, та вибрати один або декілька методів одночасно. Сторінки окремих тестів містять форми введення даних і теоретичну інформацію.

Загальний принцип спільної роботи пакету програм представлений на діаграмі діяльності (рис. 2). Як

можна бачити, виконання навіть одного тесту не є тривіальною задачею, що включає багато етапів з використанням всіх модулів. За рахунок чітко визначених протоколів та інтерфейсів комунікації між модулями системи, та «лінивого виконання» всіх етапів, досягається високий рівень ефективності та надійності системи.

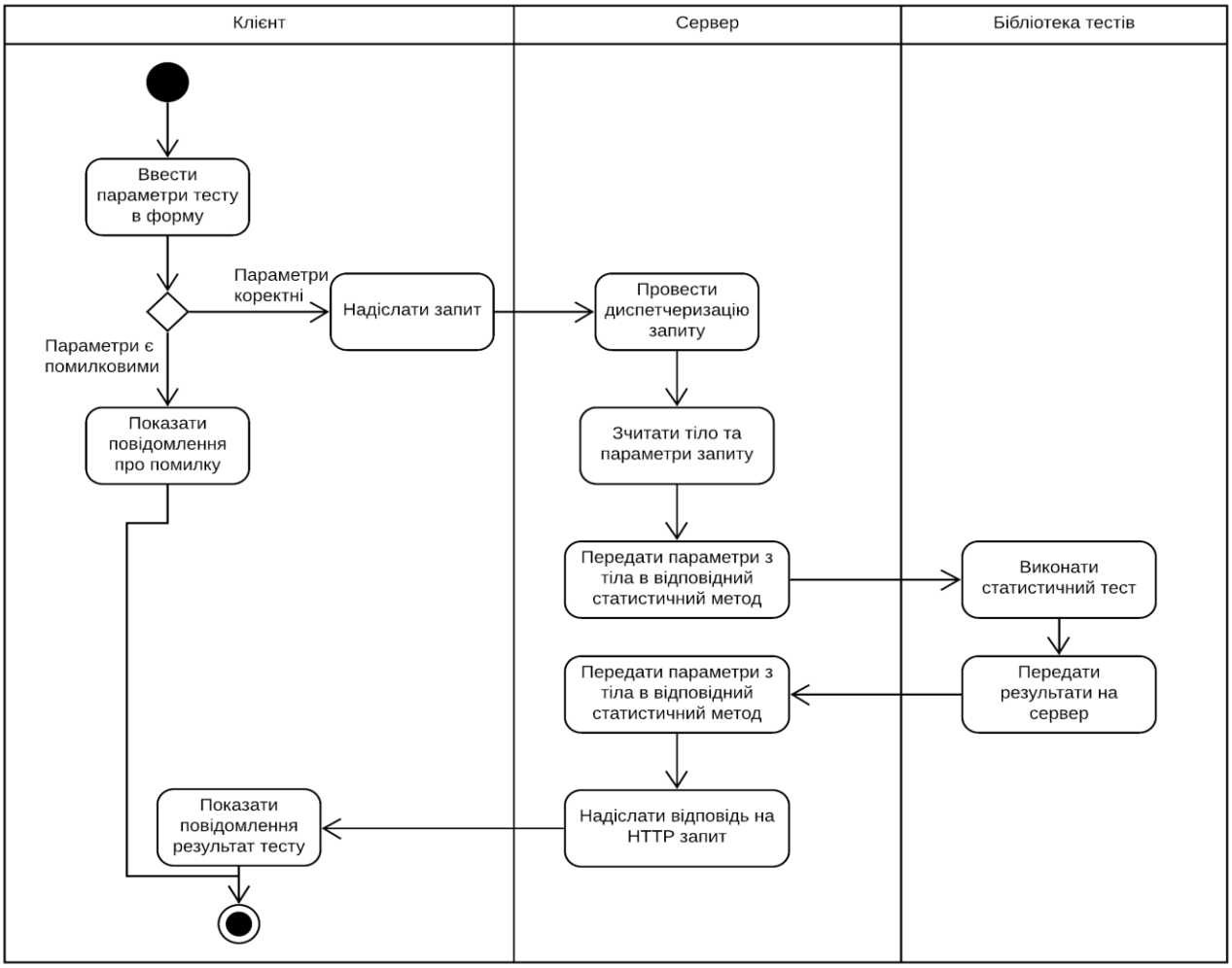

Рис. 2. Діаграма діяльності для повного циклу запит-відповідь

Доступ до прикладного інтерфейсу можна отримати за рахунок виконання запитів. Створення HTTP запитів пропонується проводити з використанням мов програмування, однак для тестування API можна використати HTTP клієнт на зразок

Веб-додаток

При переході на сайт з пошукової системи, користувача буде направлено на головну сторінку. Якщо було введено адресу іншої сторінки веб-додатку, або перейдено за посиланням з іншого ресурсу, відкриється сторінка за що знаходиться за конкретною адресою.

Головна сторінка надає довідкову інформацію про зміст веб-додатку та корисні посилання. Використавши верхню навігаційну панель, можна перейти на сторінки що відповідають тестам NIST (рис. 3) та багатовимірних статистик.

За замовчуванням відкриваються комплексні тести, тобто для одночасного виконання декількох методів. Кожна форма містить елементи управління «Checkbox» для вибору окремих тестів та вибору/зняття всіх одночасно. В текстові поля вводиться послідовність бітів.

Бокові панелі на обох сторінках містять меню в якому можна обрати окремо один з тестів. На сторінці з методом можна отримати довідкову інформацію і виконати обчислення.

Форми містять текстові поля для введення кожного з параметрів тесту, в самих полях є підказки про дані які необхідно вводити. Кнопка «Тест» слугує для виконання методу, а результат буде показано справа. Використання даної кнопки при некоректних даних в полях форми, призводить до появи повідомлення про помилку, текст якого завжди відповідає змісту помилки.

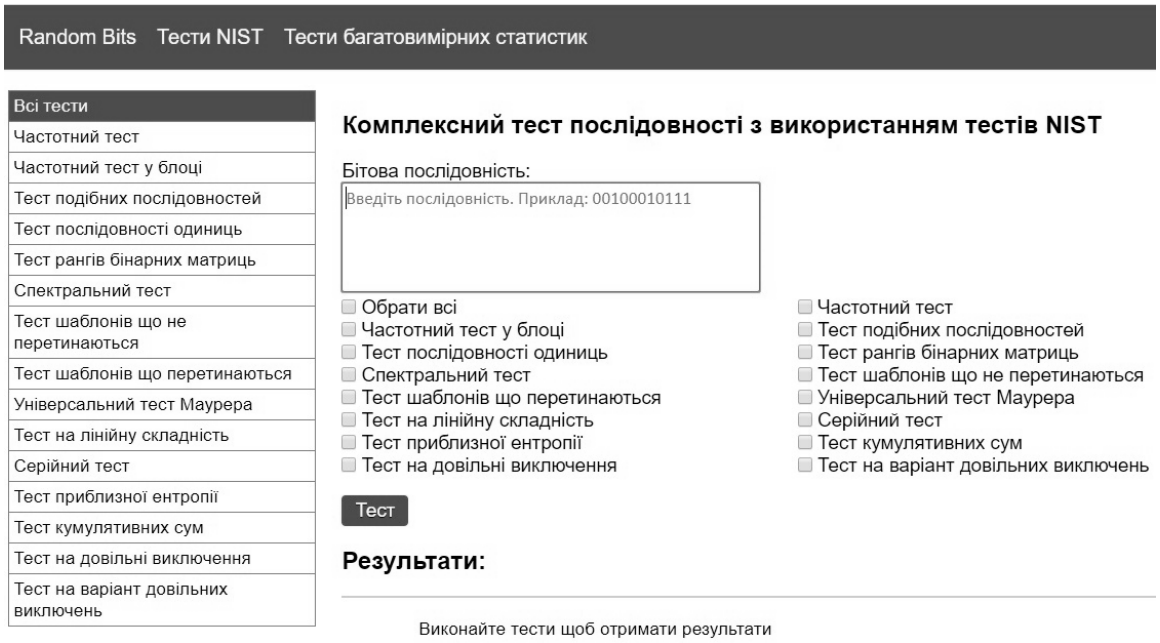

Рис. 3. Сторінка NIST

Висновки

Тестування бітової послідовності на випадковість не є новою проблемою. Наразі існує велика кількість пакетів тестів, що вирішують дану задачу. Однак, специфіка предметних галузей, системи тестування та проблеми існуючих методів, вказують на актуальність даного питання та необхідність покращення існуючих методів тестування.

Тести багатовимірних статистик дозволяють краще дослідити послідовність за рахунок використання одночасно декількох характеристик послідовності. Вони засновані на дослідженні шаблонів довжин два та/або три, і допомагають виявляти приховані залежності між даними. Головною перевагою тестів є їх ефективність на послідовностях короткої довжини.

Запропонований в роботі підхід і програмний засіб надає декілька можливих рівнів використання, в залежності від вимог користувача, і складається з:

– Бібліотека на мові Java, що включає 15 тестів NIST та 9 тестів багатовимірних статистик.

– Прикладний програмний інтерфейс що надає можливість використовувати тести за допомогою HTTP запитів.

– Веб-додаток, який може бути використано для тестування послідовностей через браузер.

Даний програмний засіб рекомендовано використовувати при дослідженні послідовностей на випадковість. Вони можуть бути застосовані в одні з наступних областей:

– Наукові дослідження – встановлення залежності між будь-якими експериментальними даними, розробка генераторів псевдовипадкових чисел, створення нових методів перевірки послідовності на випадковість.

– Криптографія – перевірка послідовностей згенерованих генераторами псевдовипадкових чисел, дослідження алгоритмів шифрування.

– Розробка та супровід програмних продуктів – тестування ефективності алгоритмів та систем заснованих на випадковості, перевірка криптографічних засобів системи.

Література

[1]. Д. Кнут, *Искусство программирования. Том 2. Получисленные алгоритмы*, М.: Вильямс, 2007, 832 с.

[2]. М. Иванов, Д. Михайлов, И. Чугунков, *Стохастические методы и средства защиты информации в компьютерных системах и сетях*, М.: Кудиц-Пресс, 2009, 512 с.

[3]. М. Иванов, И. Чугунков, *Криптографические методы защиты информации в компьютерных системах и сетях*, М.: НИЯУ МИФИ, 2012, 400 с.

[4]. A. Rukhin, *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, National Institute of Standards and Technology, 2010. [Electronic resource]. Online: https:// nvlpubs. nist. gov / nistpubs / Legacy / SP / nistspecial publication800-22r1a.pdf.

[5]. *DIEHARD Statistical Tests*. [Electronic resource]. Online: https://stat.fsu.edu/pub/diehard/.

[6]. *Diehard Tests*. [Electronic resource]. Online: https://en.wikipedia.org/wiki/Diehard_tests.

[7]. *TestU01: A software library in ANSI C for empirical testing of random number generators*. Department d'Informatique et de Recherche Operationnelle, University of Montreal. 2013. [Electronic resource]. Online: http://simul.iro.umontreal. ca/testu01/ guide-shorttestu01.pdf.

[8]. И. Гайдышев, *Программное обеспечение анализа данных AtteStat. Руководство пользователя. Версия 13*, 2012, 505 с.

[9]. S. Popereshnyak, G. Dimitrov, "The Testing of Pseudorandom Sequences using Multidimensional Statistics", *Proceedings of the 1st International Workshop on Digital Content & Smart Multimedia (DCSMart 2019) Lviv, Ukraine, December 23-25*, pp. 151-161, 2019.

[10]. V. Masol, S. Popereshnyak, "Statistical analysis of local sections of bits sequences", *Journal of Automation and Information Sciences,* Vol. 51, pp. 31-45, 2019. DOI: 10.1615/JAutomatInf Scien.v51.i10.30.

[11]. V. Masol, S. Popereshnyak, "Checking the Randomness of Bits Disposition in Local Segments of the (0, 1)-Sequence", *Cybernetics and Systems Analysis,* no. 56(3), pp. 1-8, 2020. DOI: 10.1007/s10559-020-00267-0.

УДК 519.212.2 : 681.51

Поперешняк С.В. Программное средство для тестирования битовых последовательностей малой длины на случайность

Аннотация. *Данная статья изучает случайность и наиболее известные наборе тестов для ее обнаружения. Особое внимание уделяется статистическому исследованию битовых последовательностей. Имеющиеся наборы тестов показывают низкую гибкость и универсальность в средствах нахождения скрытых шаблонов в данных небольшой длины (до 100 бит). Для решения этой проблемы предложено использовать алгоритмы на основе многомерных статистик. Данные алгоритмы сочетают все преимущества статистических методов и является единственной альтернативой для анализа последовательностей короткой и средней длины. В данной работе рассмотрены статическое тестирование последовательностей с использованием многомерной статистики. В работе приведены формулы для тестирования битовых последовательностей на случайность, с использованием двумерных или трехмерных статистик, которые могут быть применены для тестирования коротких и средних последовательностей. Для реализации предложенной методики было разработано программное средство для тестирования битовой последовательности на случайность. Данное средство включает в себя тесты NIST, а также тесты с использованием многомерной статистики, которые хорошо себя зарекомендовали при тестировании битовой последовательности малой длины. В результате применения разработанного средства возможно проанализировать битную последовательность и выбрать наиболее качественную псевдослучайную последовательность для использования в той или иной предметной области.*

Ключевые слова: *программное средство, битная последовательность, тестирование, многомерные статистики; случайные последовательности; псевдослучайная последовательность; статистическое тестирование.*

Popereshnyak S. Software for testing small-length bit sequences for randomness

Abstract. *This article dedicated to systematization of scientific positions about the static testing of sequences, widely used in cryptographic systems of information protection for the production of key and additional information (random numbers, vectors of initialization etc.) In this paper, randomness and the best-known test suite for detecting it is examined. Testing a bit sequence for randomness is not a new problem. Now there are a large number of test packages that solve this problem. Particular attention is paid to the statistical study of bit sequences. However, the specificity of subject areas, testing systems and problems of existing methods indicate the relevance of this issue and the need to improve existing testing methods. The available test suites show low flexibility and versatility in finding hidden patterns in small data lengths (up to 100 bits). To solve this problem, it is proposed to use algorithms based on multivariate statistics. Tests for multivariate statistics allow you to better explore a sequence by using multiple sequence characteristics simultaneously. They are based on examining patterns of length two and/or three and help to uncover hidden dependencies between data. These algorithms combine all the advantages of statistical methods and are the only alternative for analyzing short and medium length sequences. In this paper, static testing of sequences using multivariate statistics is considered. The paper provides formulas for testing bit sequences for randomness, using two-dimensional or three-dimensional statistics, which can be used to test short and medium sequences. To implement the proposed technique, a software tool was developed to test the bit sequence for randomness. This tool includes NIST tests as well as tests using multivariate statistics, which have worked well for testing short bit sequences. As a result of using the developed tool, it is possible to analyze a bit sequence and select the highest quality pseudo-random sequence for use in a particular subject area.*

Keywords: *software, bit sequence, testing, multidimensional statistics; random sequences; pseudo-random sequence; statistical testing.*

Поперешняк Світлана Володимирівна, кандидат фізико-математичних наук, доцент, доцент кафедри програмних систем і технологій факультету інформаційних технологій Київського національного університету імені Тараса Шевченка.

Поперешняк Светлана Владимировна, кандидат физико-математических наук, доцент, доцент кафедры программных систем и технологий факультета информационных технологий Киевского национального университета имени Тараса Шевченко.

Popereshnyak Svitlana, PhD (Theory of Probability and Mathematical Statistics), Associate Professor of Department of Software Systems and Technologies, Taras Shevchenko National University of Kyiv.

Отримано *28 липня 2020 року*, затверджено редколегією *12 серпня 2020 року*