УДК 004.9 **DOI: 10.20998/2078-9130.2025.2.344692**

М. В. НЕКРАСОВА

ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ СТВОРЕННЯ ТА ТЕСТУВАННЯ ГЕНЕРАТОРІВ ПСЕВДОВИПАДКОВИХ ЧИСЕЛ

У статті представлено огляд сучасних наукових досліджень у галузі нейрокриптографії, присвячених використанню нейронних мереж для побудови та аналізу генераторів псевдовипадкових чисел (ГПВЧ). Розглянуто основні підходи до створення генераторів псевдовипадкових послідовностей, що застосовуються в криптографічних системах, системах захисту інформації та комп'ютерному моделюванні. Наведено класифікацію різних типів ГПВЧ, включаючи алгоритмічні, апаратні та гібридні реалізації, а також визначено основні критерії їх криптографічної стійкості, зокрема непередбачуваність, рівномірність розподілу, довгий період повторення та стійкість до статистичного аналізу. У роботі окреслено причини вибору конкретних типів генераторів залежно від практичних задач, вимог до продуктивності, рівня безпеки та сфери застосування. Подано короткі теоретичні відомості щодо принципів функціонування нейронних мереж, їх навчання та можливостей використання у криптографічних застосуваннях. Проведено порівняльний аналіз різних архітектур нейронних мереж у контексті їх використання для генерації псевдовипадкових послідовностей і для аналізу статистичних властивостей отриманих даних. Зокрема, розглянуто можливості класичних рекурентних нейронних мереж, таких як мережі Елмана та мережі з довготривалою короткочасною пам'яттю (LSTM), а також сучасних генеративно-змагальних мереж (GAN), що демонструють перспективні результати у задачах моделювання складних випадкових процесів. Окрему увагу приділено методам статистичного тестування вихідних послідовностей ГПВЧ. Наведено огляд відомих наборів тестів, що використовуються для оцінювання випадковості та криптографічної якості послідовностей. Проаналізовано результати ключових наукових досліджень, присвячених використанню нейронних мереж як для генерації, так і для тестування псевдовипадкових послідовностей. Наголошено на важливості комплексної перевірки статистичних характеристик ГПВЧ та підкреслено можливі негативні наслідки недооцінки етапу тестування при оцінюванні надійності та безпеки криптографічних систем.

Ключові слова: псевдовипадкові числа, генератор псевдовипадеових чисел, нейрокриптографія, нейронні мережі, тестування ГПВЧ, створення ГПВЧ, криптографічна стійкість.

M. NEKRASOVA

APPLICATION OF NEURAL NETWORKS IN THE DESIGN AND TESTING OF PSEUDORANDOM NUMBER GENERATORS

The article presents a review of modern research in the field of neurocryptography related to pseudorandom number generators (PRNGs). The main approaches to the construction of generators of pseudorandom sequences used in cryptographic systems, information security systems, and computer modeling are considered. Different types of PRNGs and their implementations are analyzed, including algorithmic, hardware-based, and hybrid solutions. The main criteria of cryptographic strength of generators are also defined, such as unpredictability, uniform distribution of output values, long repetition period, and resistance to statistical analysis. The paper outlines the reasons for selecting specific types of generators depending on practical tasks, performance requirements, security level, and application area. The fundamental concepts of neural network theory are presented, including the principles of their operation, training mechanisms, and potential applications in cryptographic systems. A comparative analysis of different neural network architectures is carried out in the context of their use for generating pseudorandom sequences and for analyzing the statistical properties of generated data. In particular, classical recurrent neural networks such as Elman networks and Long Short-Term Memory (LSTM) networks are considered, as well as modern generative adversarial networks (GANs), which demonstrate promising results in modeling complex random processes. Special attention is paid to the methods of statistical testing of PRNG output sequences. An overview of widely used statistical test suites for evaluating the randomness and cryptographic quality of generated sequences is presented. The results of key studies on PRNG generation based on neural networks are analyzed, including both classical recurrent architectures and modern generative approaches. In addition, methods for testing PRNGs using neural networks are considered, and the potential of machine learning techniques for detecting hidden regularities in generated sequences is discussed. The importance of comprehensive statistical evaluation of PRNGs is emphasized, and the possible negative consequences of underestimating the testing stage when assessing the reliability and security of cryptographic systems are highlighted.

Keywords: pseudorandom numbers, pseudorandom number generator, neurocryptography, neural networks, PRNG testing, PRNG design, cryptographic security

Вступ. Інформаційна безпека стає однією з головних областей дослідження у сфері інформаційних технологій внаслідок великої кількості загроз, що виникають через атаки зловмисників з метою розкрадання даних або порушення роботи системи [1–2]. Уразливості в алгоритмах генераторів псевдовипадкових чисел вже неодноразово призводили до зламів та несанкціонованого доступу до особистої інформації. Наприклад, злом CryptGenRandom компанії Microsoft, яка рекомендувала в рамках Win32 використовувати даний алгоритм повсюдно при генерації випадкових чисел, або атака на протокол SSL від компанії Netscape, який у ранніх версіях мав низьке значення ентропії. Алгоритм шифрування сильний настільки, наскільки якісна за своїми статистичними характеристиками випадкова послідовність отримана з виходу ГПВЧ, оскільки саме ГПВЧ відповідає за непередбачувану частину алгоритму. Найбільш відомими є випадки злому генератора OpenSSL в Debian, коли сильне зменшення ентропії значень, що генеруються, призвело до вразливостей, що дозволяють зробити

атаку на секретні ключі [3]. Виявлення вразливостей у класі 'Secure Random' в Java дозволило генерувати колізії в одноразових числах, що використовуються для ECDSA реалізацій Bitcoin, відновленню приватних ключів RSA і можливому крадіжці біткойнів з гаманців [4]. Спроби зламування не зупиняються ні на мить. В останні роки цікавим об'єктом для атаки зловмисників був генератор випадкових чисел Adobe Flash (Action Script 3.0)3, який використовується в онлайн-покері та онлайн-казино. Наведені вище приклади демонструють важливість ГПВЧ у питаннях забезпечення безпеки інформації.

У зв'язку з цим розглянемо найбільш важливі реалізації ГПВЧ і методи їх тестування в області, що стрімко розвивається – нейрокриптографії. Основи нейрокриптографії вперше були описані в 1990 р., а потім активно розвинені [5- 7]. На початку XXI століття дедалі активніше почали шукати можливості застосування нейронних мереж (НМ) у криптографії, а також розвивати ідеї, розглянуті раніше. До таких робіт можна віднести: використання ваг НМ як симетричного ключа в алгоритмі DES [9], застосування НМ у симетричному шифруванні [10, 11], а також використання НМ для створення хеш-функції та отримання електронного підпису. Також важливим є використання НМ у блоці підстановок (S-блоці), а також поліпшення традиційних ГПВЧ (IDEA, ANS X.9) та потокових генераторів за допомогою НМ [12, 13]. При цьому дедалі більше робіт у нейрокриптографії було присвячено розробці нових методів встановлення безвісного з'єднання між людьми. А саме для синхронізації НМ використовувати ваги і зміщення в якості секретного ключа [14, 15].

Мета та постановка задачі. Основи ГПВЧ. ГПВЧ - це детерміністський алгоритм генерації послідовності чисел, що не відрізняється за своїми статистичними властивостями від істинної випадкової послідовності, починаючись з істинно випадкового секретного початкового стану. ГПВЧ використовуються всюди у сфері шифрування інформації (зашифрування, автентифікації та ідентифікації), хешингу та створення водних знаків. ГПВЧ може мати велику кількість вразливостей, таких як передбачувана залежність між числами, передбачуване початкове значення генератора і мала довжина періоду послідовності, що генерується. Один із перших методів генерування псевдовипадкових чисел ґрунтувався на циклічному процесі, що складається з зведення в квадрат числа з N десяткових цифр та подальшим виділенням з його середини N/2 цифр. Вже через кілька років було виявлено, що період даної генерованої послідовності вкрай малий, і в кінцевому підсумку може видавати числа, які повторюватимуться через кілька ітерацій.

Існує велика кількість некриптостійких ГПВЧ, серед яких найбільш відомими є Blum-Blum-Shub [16], алгоритми Mersenne Twister [17, 18]. Розглянемо найпопулярніші програмні реалізації таких ГПВЧ. Функція rand() в Debian заснована на ідеї регістру зсуву з лінійним зворотним зв'язком з нерівномірно розподіленими числами за послідовністю. Функція random.randint заснована на ідеї Mersenne Twister, і, незважаючи на рівномірний розподіл чисел за послідовністю, не є криптографічно стійкою, оскільки після кінцевого числа вихідних станів може бути відновлено внутрішній стан системи. Також сюди слід віднести arc4random з OpenBSD, який вважався криптографічно стійким ГПВЧ (КСГПВЧ), доки не були виявлені залежності між розташованими на великому видаленні бітами в згенерованому потоці. Їх виявлення не дозволяє повністю відновити потік випадкових цифр, але при цьому не дозволяє вважати цей алгоритм криптостійким.

Якщо розглядати КСГПВЧ, пов'язані з питаннями гарантування безпеки, то до них пред'являється ряд особливих вимог, таких як: непередбачуваність (неможливо відновити послідовність або передбачити (n+1) елемент з попередньої послідовності з n елементів, не знаючи початкового стану генератора), статистичні властивості послідовності повинні бути такими, що не відрізняються від від істинно випадкової; ефективна програмна та апаратна реалізація.

Найчастіше в дослідженнях, пов'язаних з КСГПВЧ, розглядаються Java SecureRandom [4] та його підвид – SHA1PRNG5, які не маючи відомих на даний момент уразливостей. Вони побудовані на основі слабкої Хеш-функції Secure Hash Algorithms 1 (SHA-1) [19]. Незважаючи на те, що виникають складності навіть при передбаченні частини цифр даної послідовності, аналіз статистичних характеристик може пролити світло на вразливість алгоритму.

Найбільш сильний алгоритм /dev/urandom6, заснований на зборі даних із різних зовнішніх джерел шуму, таких як: час пошуку по диску, руху мишки та клавіатури, затримки мережі. Це дозволяє підвищити ентропію та знизити ймовірність передбачення вихідних чисел. Як приклад, ядро Linux не завантажиться, якщо /dev/random не буде мати достатній рівень ентропії для генерації випадкових чисел.

Багато які КСГПВЧ мають суттєво нижчу продуктивність у порівнянні зі звичайними ГПВЧ, тому вони не використовуються масово, незважаючи на те, що вихідна послідовність КСГПВЧ має набагато кращі характеристики порівняно зі звичайними ГПВЧ. Наприклад, у комп'ютерній грі, наближена до реальної випадковості цифр послідовність буде не така важлива, як продуктивність. Таким чином, дослідники під час проєктування нових ГПВЧ постійно шукають баланс між ефективністю (швидкістю генерації) та надійністю (рівнем безпеки).

Різноманітність видів блокових ГПВЧ вимагала нової класифікації, тому пропонується на додаток до традиційних мереж Фестеля розглядати також мережі, такі як «квадрат» і «куб». 2D та 3D схеми таких мереж зосереджені на використанні паралельних обчислень за допомогою розподілених архітектур суперкомп'ютерів і мають високий рівень паралелізму на рівні

елементарних операцій, а також архітектур, раундові функції яких побудовані по 2D та 3D схемам. Також у класифікації треба враховувати поділ ГПВЧ щодо наявності доступу до будь-якого елемента псевдовипадкової послідовності. Це дозволяє використовувати розпаралелювання у обчисленнях, збільшує швидкість і розширює сферу застосування ГПВЧ.

Основи нейронних мереж. При взаємодії людини з навколишнім середовищем, тіло вловлює зміни завдяки рецепторам та органам почуттів, які надалі передають сигнали в тілі за допомогою нервової системи та нейронів, зокрема, у мозок, який приймає рішення щодо реакції на подію [20]. При цьому функціональність нейронної мережі визначається зв'язками, утвореними між нейронами. Ця ідея покладена в основу штучного нейрона, де функціональні зв'язок між нейронами характеризувалися за допомогою числової характеристики – ваги. На теперішній момент все це зазнало безліч змін і якщо спочатку ваги мали вибиратися вручну, то згодом було розроблено метод зворотного поширення помилки, при якому задані випадковим чином ваги коригуються в процесі навчання [21, 22]. Система навчається до тих пір, поки не буде отримано найкраще з можливих значень критерію якості, тобто поки не буде отримано оптимальний набір для вирішення поставленого завдання.

Однією з основних властивостей НМ є узагальнююча (підсумовуюча) здатність, що дозволяє отримувати логічні результати на раніше недосліджених даних. Основне завдання НМ полягає у поширенні сигналу від входу НМ до її виходу через групу пов'язаних шарів, де під шаром розуміються незалежні нейрони, об'єднані в одну групу. Кількість шарів, нейронів на вхідному, вихідному та прихованому шарах, а також алгоритм навчання є гіперпараметрами мережі, значення яких підбираються під час експерименту з метою отримання найкращого значення метрики.

Для зменшення розбіжності (помилки) між вихідними даними мережі та реальними значеннями використовують алгоритм навчання, який при заданих гіперпараметрах дозволяє знайти оптимальне значення ваги. Значення ваги тим ближче до 0, чим менше зв'язок між нейронами.

Для зручності роботи та швидкої побудови архітектури НМ були створені фреймворки Tensor-Flow, Pytorch [25, 26], що здатні проводити обчислення на графічних та тензорних процесорах і сильно масштабуються, що дозволяє запускати навчання як простих систем на домашньому комп'ютері, так і ресурсомістких на кластерах.

Застосування нейронних мереж в ГПВЧ. Дослідники намагаються використовувати НМ для генерації криптографічно стійких ГПВЧ вже понад двадцять років. Наприклад, програми мережі Хопфілда до потокових генераторів та інше застосування було розглянуто в [27, 28]. У рамках криптографії більш переважними є нелінійні системи, які для відтворення певної підпослідовності заданої послідовності вимагають повторення всієї послідовності з початку, а не з певного елемента [29].

НМ можна використовувати як хеш-функцію. За допомогою навченої НМ можна легко знайти вихідні значення за вхідними значеннями, а ось за вихідними визначити вхідні набагато важче, особливо не володіючи знанням про гіперпараметри мережі. Остання властивість робить НМ нелінійним алгоритмом, що дуже важливо у питаннях криптографії.

Виділимо основні види навчання НМ, що використовуються в нейрокриптографії, можливі пов’язані з цим складності та їх недоліки. Навчання з учителем найбільше підходить для завдання тестування послідовності ГПВЧ, де в якості переваги варто виділити можливість генерації вибірки необхідного розміру, завдяки доступу до ГПВЧ, що тестується. Далі вибірка розбивається на набір ознак та правильних відповідей мережі. Навчену мережу використовують надалі для передбачення та оцінки статистичних характеристик ГПВЧ. До недоліків можна віднести необхідність розбиття вихідної послідовності з ГПВЧ на підпослідовності, не маючи інформації про її період. При незв'язаній зміні поведінки системи (подачі на вхід модифікованих даних), алгоритм ніяк не зможе цього вловити, а обробка даних з ГПВЧ з великим періодом послідовності викликатиме додаткові складності внаслідок обмежених ресурсів машини, на якій проводиться тестування.

Для алгоритму навчання з підкріпленням характерна наявність агента, що діє у певному середовищі. Там йому доступний обмежений набір дій, які він здатний незалежно комбінувати для отримання максимальної сумарної нагороди при досягненні мети. Кожна дія переводить агента з одного стану в інший, за що нараховується нагорода - якесь числове значення (позитивне або негативне в залежності від дії). Агент вибирає послідовність дій відповідно до заданої стратегії. У цьому алгоритм налаштовується в такий спосіб, щоб за кожну ітерацію навчання збільшувалася нагорода для агента [30]. Під час навчання з підкріпленням виникає складність оцінки нагороди для процесу навчання. Оцінити її можна на основі отриманого знання, що дуже важко при тестуванні та використанні випадкових чисел як вхідних даних.

Результати застосування статистичних тестів. Існує велика кількість тестів для тестування вихідної послідовності з ГПВЧ: DieHarder7, його розширена версія TestU01 та John Walker's ENT8. Набір тестів NIST використовується в більшості робіт даного дослідження та містить 15 задокументованих тестів, що дозволяють визначити, чи є бінарна послідовність псевдовипадковою. [31, 32].

Кожен з тестів повертає P-величину в діапазоні від 0 до 1, що характеризує тестовану послідовність, і формулюється для тестування нульової гіпотези H_0 (послідовність випадкова) і альтернативної H_1

(послідовність невипадкова). У результаті тестування можуть виникати, зазвичай, помилки двох типів: помилка першого типу - послідовність істинно випадкова, але при цьому нульова гіпотеза відкидається; помилка другого типу – якщо послідовність невипадкова, але при цьому нульова гіпотеза H_0 приймається.

Традиційно позначимо ймовірність помилки першого типу як α (рівень значущості). І в залежності від величини Р та певного критичного значення $P_{кр}(\alpha)$, нульова гіпотеза H_0 або приймається, або відкидається на користь альтернативної H_1. Незважаючи на те, що значення α можна змінювати в залежності від мети дослідження програми в більшості криптографічних завдань, він встановлюється на рівні 0.01.

Розглянемо деякі з тестів набору NIST. У частотному тесті ведеться підрахунок одиниць і нулів у всій послідовності (або її частині) для визначення рівномірності їх розподілу. У тестах послідовностей, що перекриваються і неперекриваються, відбувається підрахунок виявлених раніше патернів у послідовності, але в першому зсув завжди відбувається на один біт, а в другому при знаходженні патерну відбувається стрибок по послідовності на довжину знайденого патерну. Тест на періодичність дозволяє підрахувати частоти появи всіх 2^m підпослідовностей довжини m досліджуваної послідовності, і якщо спостерігається сильна відмінність від рівномірного розподілу, то послідовність вважається псевдовипадковою. Визначення частотних особливостей послідовності за допомогою дискретного перетворення Фур'є (спектральний тест) дозволяє визначити, чи відрізняється кількість піків, що перевищують порогове значення в 95%, від 5%.

В якості вдалої реалізації NIST для практичного використання можна навести NistRng9 мовою Python, що дозволяє в автоматичному режимі вибрати відповідний за довжиною послідовності набір тестів, повернувши значення і прапорець про успішне проходження тесту.

Створення ГПВЧ на основі нейронних мереж. Спочатку розглянемо повнозв'язні мережі, які добре справляються з розв'язанням більшості задач класифікації та регресії, особливо на табличних даних, але водночас мають такі недоліки, як швидке збільшення кількості навчуваних параметрів із зростанням кількості вхідних ознак та проблему затухання градієнта (чим більше шарів у НМ, тим слабше змінюватимуться ваги, що розташовані найближче до входу).

Припустімо, що кожен вхідний нейрон відповідає одному біту вихідної послідовності ГВПЧ. Чим більшим буде період послідовності, тим більше потрібно буде вхідних нейронів, що відповідним чином може вплинути на збільшення кількості нейронів прихованого шару та призвести до ускладнення НМ. При цьому початково, якщо це не зазначено, період вихідної послідовності невідомий, що значно ускладнює пошук патернів. Найчастіше дана архітектура використовується у складі більш складних мереж. Наприклад, як генератор у GAN [3], але інколи й окремо, під час тестування бінарних послідовностей [33] або, скажімо, для використання її як хеш-генератора внаслідок своєї нелінійності та однобічності [34].

Однією з основних переваг рекурентних мереж є наявність зворотного зв'язку, що дає змогу знаходити залежності між елементами, які розташовані на значній відстані один від одного (НМ Elman, яка має лише два шари, зі зворотним зв'язком на першому шарі та функціями активації tansig і purelin відповідно). Основною особливістю є саме наявність зворотного зв'язку на першому шарі мережі, що дає змогу подавати на вхід дані, отримані з виходу першого шару на попередньому кроці. Завдяки зворотному зв'язку одні й ті самі вхідні дані, подані до НМ з однаковими вагами та вільними членами, можуть давати різні вихідні значення через відмінності в ланцюгу зворотного зв'язку. Перевагою методу є невисокі вимоги до обчислювальних ресурсів і простота реалізації різних бітових операцій.

Також можна використовувати метод генерації псевдовипадкової послідовності (ПВП) довжиною від 25 до 1000 біт за допомогою матриці ваг, отриманих із ваг багатошарової рекурентної НМ. Послідовність, отримана на виході, провалила лише 1 з 15 тестів набору NIST. Звернувшись до навчання з підкріпленням, варто сказати про покрокову поелементна генерація ПВП, подібно до того, як агент у межах доступної йому частини марківського процесу приймає рішення. Одним з основних параметрів виступає винагорода за прагнення агента видати наступний елемент таким чином, щоб наблизити ПВП до істинно випадкової. Ідея ухвалення агентом рішення на підставі останніх N біт дає змогу подолати обмеження, накладені на довжину послідовності апаратною частиною. За допомогою глибокого навчання з підкріпленням можливо генерувати ПВП зі «змінним» періодом завдяки випадковому характеру навчання. Для моделювання взаємозв'язку між спостереженнями можна використовувати LSTM-мережу, яка виокремлює основні ознаки зі схованої частини стану марківського процесу прийняття рішень, таким чином отримуючи максимально можливу кількість даних із доступних без експоненційного зростання розмірності простору станів. У результаті вдається збільшити довжину вихідної послідовності та отримати вищий результат на наборі тестів NIST порівняно з алгоритмом навчання з підкріпленням на основі повнозв'язної мережі, якій були доступні всі стани марківського процесу.

Окремо можна розгдянути варіант, що пов'язаний з розгалуженням мережі навпіл після шару LSTM, де один шар займається обчисленням ймовірності наступних можливих дій, у той час як другий дає змогу обчислити значення величини (value function). Як алгоритм для навчання з підкріпленням було обрано фреймворк USienaRL11, причому максимально можлива довжина послідовності становила 1000 біт. Стандартна двоступенева модель генерації ПВП, де

перший ступінь відповідає за забезпечення добрих статистичних властивостей і довгого періоду, а другий - за непередбачуваність і криптографічну стійкість, має бути доповнена ще одним ступенем, пов'язаним із відновленням хороших статистичних властивостей, які могли бути втрачені на другому етапі, або ж має бути проведена аналогічна модифікація другого етапу. Цю ідею можна використати в подальших дослідженнях стосовно ГВПЧ, побудованих на основі НМ.

Однією з найцікавіших і найсучасніших розробок є ГВПЧ, заснований на архітектурі GAN. Ідея застосування архітектури GAN для створення більш досконалих ГВПЧ лежить на поверхні, оскільки однією з основних властивостей хорошого ГВПЧ є мінімально допустима ймовірність розпізнавання майбутніх послідовностей за попередніми, чого генератор може досягти шляхом послідовних ітерацій у процесі навчання GAN [3]. Якщо коротко, у її основі лежать дві НМ: генератор, який навчається відтворювати патерни істинно випадкової послідовності, і дискримінатор, який на виході видає припущення щодо того, чи є послідовність істинно випадковою або псевдовипадковою. Циклічний процес полягає у навчанні дискримінатора до досягнення певного рівня точності у розпізнаванні послідовностей, а потім — у навчанні генератора створювати таку послідовність, на якій дискримінатор допустить найбільшу помилку [38–41].

За допомогою GAN-мережі можна отримати якісний ГВПЧ, у якому повнозв'язна мережа виступає як генератор, а вихідна послідовність має хороші статистичні властивості й здатна пройти 99% тестових прикладів із набору тестів NIST. Новий підхід до розв'язання цього завдання полягає в тому, що, генератор навчають не тому, щоб вихідна послідовність за своїми статистичними характеристиками наближалася до істинно випадкової, а тому, щоб дискримінатор не був здатний передбачити ці значення.

Тут варто зазначити, що цей підхід не позбавлений недоліків, оскільки для дискримінатора можна обрати таку архітектуру мережі, яка заздалегідь буде нездатною коректно передбачати вихідне значення генератора, тим самим позбавляючи цей підхід будь-якого сенсу.

Тестування ГВПЧ за допомогою нейронних мереж. Незважаючи на велику кількість тестів, що підтверджують наявність у ГВПЧ необхідних статистичних характеристик КСГВПЧ, відсутня можливість довести істинну випадковість послідовності. Генератори, які нині вважаються криптографічно стійкими, можуть не пройти нові тести, спрямовані на пошук нових патернів у вихідних послідовностях. Як уже зазначалося вище, будь-яка вразливість у системі в подальшому, зі зростанням технологій, може бути використана зловмисником під час атаки. Таким чином, ця галузь постійно розвивається, і одним із нових напрямів є тестування ГВПЧ за допомогою НМ.

На перший погляд може здатися, що використання НМ для аналізу вихідних послідовностей ГВПЧ, однією з основних здатностей яких є знаходження патернів у вхідних даних, виглядає ідеєю, неспроможною дати логічні результати, оскільки у випадкових числах за визначенням мають бути відсутні будь-які патерни. Проте вихідна послідовність ГВПЧ є псевдовипадковою, тобто в ній апріорі присутні патерни, а випадковою вона є лише з точки зору статистичних характеристик, які відповідають істинно випадковій. Отже, НМ є ідеальним інструментом для тестування ГВПЧ.

Тестування вихідної послідовності з ГПВПЧ на випадковість проводиться або за допомогою теоретичного аналізу, або статистичних тестів. Дефекти можуть бути виявлені обома методами, і найкращим рішенням є їх спільне використання. Оскільки довести випадковість цифр послідовності неможливо, ці підходи шукають патерни та статистичні характеристики, притаманні невипадковій послідовності, тобто дають змогу довести, що послідовність є невипадковою. Варто зазначити, як було згадано раніше, що кожен пропущений дефект може бути використаний атакувальною стороною, якщо не зараз, то в майбутньому з появою більш досконалих технологій.

Існують різні підходи до тестування ГВПЧ. Це або передбачення статистичних характеристик послідовності для знаходження відхилень від характеристик істинно випадкової послідовності, або спроба передбачити (n + 1)-й елемент або m елементів послідовності на основі n попередніх елементів, не маючи знання про початковий стан ГВПЧ. Варто зазначити, що нейронні мережі при навчанні з учителем є відмінними кандидатами для реалізації алгоритму тесту на наступний біт вихідної послідовності ГВПЧ для оцінки його криптографічної стійкості. Основна ідея тесту полягає в тому, що жоден з алгоритмів не повинен бути здатний передбачити на основі k елементів послідовності (k + 1)-й елемент з ймовірністю вище 0,5.

Популярним видом архітектури, який повсюдно використовується при аналізі двовимірної інформації (зображень), є згорткові нейронні мережі, які враховують просторову залежність між елементами. Їх також можна використовувати для одномірних даних, таких як аудіо, текст або числові послідовності, тому не дивно, що виникла ідея застосування їх для аналізу вихідних послідовностей ГВПЧ. Але у згорткових нейронних мереж залежності зазвичай шукаються у досить невеликому діапазоні сусідніх значень, що не дозволяє їх ефективно використовувати для тестування – особливо КСГВПЧ, які мають великий період повторюваності.

Була створена програма, яка дозволила протестувати велику кількість ГВПЧ. Цікавим висновком дослідження є погіршення результатів при використанні двонаправлених рекурентних мереж (як LSTM, так і GRU), що спочатку є контринтуїтивним. Порівнюючи цю систему з набором тестів DieHarder,

слід зазначити, що на відміну від цієї системи, DieHarder вважає послідовності, отримані за допомогою алгоритмів arc4random та SHA1PRNG, істинно випадковими. Замість результатів по кожному з наборів тестів система видає єдиний результат про те, чи пройшла послідовність тест чи ні. Тобто система здатна замінити цей набір тестів, але при цьому, як і DieHarder, розроблена програма не здатна розпізнати послідовність з /dev/urandom та Hardware Random Number Generator (HRNG), через високу якість згенерованих чисел.

Було досягнуто точності 98 % у передбаченні вихідної послідовності для XORShift-128 та Mersenne Twister за рахунок копіювання структури даних генераторів шляхом відтворення нейронними мережами аналогів елементів, що використовуються в цих ГВПЧ (наприклад, відтворення XOR-елемента за допомогою групи нейронів).

Висновок. У роботі були розглянуті два основні напрямки в області ГВПЧ – створення нових видів ГВПЧ, включно з криптографічно стійкими ГВПЧ, а також тестування вихідних бітових послідовностей цих генераторів. Було розглянуто основні види архітектур нейронних мереж, що застосовуються в нейрокриптографії, та роботи, побудовані на їх основі.

Метою цієї статті було продемонструвати важливість нейронних мереж у сфері ГВПЧ, виділити основні тенденції та структурувати їх. Нейронні мережі є швидкозростаючою областю, у якій завдяки стрімкому розвитку апаратної складової відбувся якісний стрибок, що опосередковано вплинув і на сферу криптографії.

Практично всі види нейронних мереж застосовуються у сфері ГВПЧ. Так, наприклад, для тестування більше підходять рекурентні мережі, здатні аналізувати просторову залежність елементів послідовності, що знаходяться на великій відстані один від одного. У той час як для генерації набирає популярності архітектура GAN, що дозволяє шляхом послідовних ітерацій покращувати характеристики вихідної послідовності за допомогою її аналізу іншою мережею на кожній епосі. Нейронні мережі дозволили проаналізувати вихідні послідовності з регістрів зсуву з лінійним зворотним зв'язком та Mersenne Twister і відтворити їх без знання початкового стану систем

Список літератури

1. *Apdullah Y., Yakup K.* Neural Network Based Cryptography. Neural Network World. 2014, no. 24, p. 177–192. – https://doi.org/10.14311/NNW.2014.24.011
2. *Kannan M. R. Gnanam V.* Neural networkbased decryption for random encryption algorithms. Proceedings of 3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication, Hong Kong, 20-22 August 2009, p. 603–605. – https://doi.org/10.1109/ICASID.2009.5277002
3. *Oak R., Rahalkar C., Dhaval G.* Using Generative Adversarial Networks for Secure Pseudorandom Number Generation. Proceedings of 2019 ACM SIGSAC Conference, London, 11–15 November 2019, p. 2597–2599. – https://doi.org/10.1145/3319535.3363265
4. *İnce K.* Security Analysis of Java SecureRandom Library. Avrupa Bilim ve Teknoloji Dergisi. 2021, no. 24, p. 157–160.
5. *Lauria F. E.* On Neurocryptology. Proceedings of the Third Italian Workshop on Parallel Architectures and Neural Networks. Salerno, 15-18 May, 1990. p. 337–343.
6. *Pointcheval D.* Neural networks and their cryptographic applications. Livre des resumes Eurocode Institute for Research in Computer Science and Automation. 1994. p. 1–7.
7. *Schneider B.* Applied Cryptography. Protocols, Algorithms, and Source codes in C. New York, Wiley, 1996. – 758 p.
8. *Luciano D., Prichett G.* Cryptology: From Caesar Ciphers to Public-key Cryptosystems. The College Mathematics Journal. 1987, no. 18(1), p. 2–17. – https://doi.org/10.1080/07468342.1987.11973000
9. *Godhavari T., Alainelu N. R., Soundararajan R.* Cryptography using neural network. IEEE INDICON 2005 Conf., Chennai, India; 11-13 December 2005, p. 258–261. – https://doi.org/10.1080/07468342.1987.11973000
10. *Arvandi M., Wu S., Sadeghian A., Melek W. W.,* Woungang I. Symmetric Cipher Design Using Recurrent Neural Networks. The 2006 IEEE International Joint Conference on Neural Network Proceedings, Vancouver, BC, Canada, 2006, p. 2039–2046. – https://doi.org/10.1109/IJCNN.2006.246972
11. *Guo D., Cheng L.-M., Cheng L. L.* A new symmetric probabilistic encryption scheme based on chaotic attractors of neural networks. Applied Intelligence. 1999, no. 10(1), p. 71–84. – https://doi.org/10.1023/A:1008337631906
12. *Noughabi A. M. N., Sadeghiyan B.* Design of S-boxes based on neural networks. Proceedings of "International Conference on Electronics and Information Engineering", Kyoto, 1-3 August 2010, p. 172–178. – https://doi.org/10.1109/ICEIE.2010.5559741
13. *Karras D. A. Zorkadis V.* On neural network techniques in the secure management of communication systems through improving and quality assessing pseudorandom stream generators. Neural networks: the official journal of the International Neural Network Society. 2003, no. 16(5-6), p. 899–905. – https://doi.org/10.1016/S0893-6080(03)00124-2
14. *Ruttor A.* Neural Synchronization and Cryptography. PhD thesis. Bayerischen Julius-Maximilians-Universitat at Wurzburg. 2006.
15. *Sadırodlu S., Ozkaya N.* Neural Solutions for Information Security. Journal of Polytechnic. 2007, vol. 10, no. 1, p. 21–25.
16. *Wu X., Han Y., Zhang M. та ін.* Pseudorandom Number Generators Based on Neural Networks: A Review. Journal of King Saud University – Computer and Information Sciences. 2025. – https://doi.org/10.1007/s44443-025-00007-4
17. *Fei C., Zhang X., Wang D. та ін.* EPRNG: Effective Pseudo-Random Number Generator on the Internet of Vehicles Using Deep Convolution Generative Adversarial Network. Information. 2025. Vol. 16, № 1. – https://doi.org/10.3390/info16010021
18. *Rivas Becerra M. L., Raygoza Panduro J. J., Cisneros S. O.* Design of a New Neuro-Generator with a Neuronal Module to Produce Pseudorandom Sequences. Electronics. 2024. Vol. 13, № 10. – https://doi.org/10.3390/electronics13101955
19. *Eastlake D. E., Jones P. E.* US Secure Hash Algorithm 1 (SHA1). Request for Comments. 2001, no. 3174, p. 1–22. – https://doi.org/10.17487/rfc3174
20. *Zhang Y. et al.* GAN-Based Pseudo Random Number Generation Optimized through Genetic Algorithms, Complex & Intelligent Systems, 2024.
21. *McCulloch W. S., Pitts W.* A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics. 1943, no. 5(4), p.115–133. – https://doi.org/10.1007/BF02478259
22. *Hassan A. A. et al.* Artificial Neural Network Chaotic PRNG and Encryption on FPGA, Engineering Applications of Artificial Intelligence, 2023.
23. *Vrahatis M., Magoulas G., Parsopoulos K., Plagianakos V.* Introduction to artificial neural network training and applications. Proceedings of 15th Annual Conference of Hellenic Society for Neuroscience, Greece, 27 – 29 October 2000, p. 1–14.
24. *Kingma D. P., Ba J.* Adam: A method for stochastic optimization. Proceedings of International Conference on Learning Representations (ICLR), San Diego, 7 – 9 May 2015, p. 1–13.
25. *Abadi M., Barham P., Chen J.* Tensorflow: A system for large-scale machine learning. Proceedings of 12th USENIX Symposium on Operating Systems Design and Implementation is sponsored by USENIX, Savannah, 2 – 4 November 2016, p. 265–283.

26. *Paszke A., Gross S., Massa F.* et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. Advances in Neural Information Processing Systems, 2019, no. 32, p. 8024–8035.
27. *Chi-Kwong C., Cheng L. M.* The convergence properties of a clipped Hopfield network and its application in the design of key stream generator. IEEE Transactions on Neural Networks. 2001, no. 12, p. 340–348. https://doi.org/10.1109/72.914528
28. *Wang, Y., Wang, G., Zhang, H.* (2010). Random Number Generator Based on Hopfield Neural Network and SHA-2 (512). In: Luo, Q. (eds) Advancing Computing, Communication, Control and Management. Lecture Notes in Electrical Engineering, vol. 56. Springer, Berlin, Heidelberg. p. 198–205. https://doi.org/10.1007/978-3-642-05173-9_26
29. *Desai V. V., Deshmukh V. B., Rao D. H.* Pseudo Random Number Generator Using Elman. Proceedings of 2011 IEEE Recent Advances in Intelligent Computational Systems, Trivandrum, 22-24 September 2011, p. 251–254. https://doi.org/10.1109/RAICS.2011.6069312
30. Sutton, R.; Barto, A. Reinforcement Learning: An Introduction. IEEE Transactions on Neural Networks, vol. 9, no. 5, p. 1054–1054, Sept. 1998. https://doi.org/10.1109/TNN.1998.712192
31. *L'Ecuyer P., Simard R.* TestU01: A C library for empirical testing of random number generators. ACM Transactions on Mathematical Software. 2007, vol. 33, no. 4, p. 22–40. https://doi.org/10.1145/1268776.1268777
32. *Rukhin A., Soto J., Nechvatal J.* et. al. A statistical test suite for random and pseudorandom number generators for cryptographic applications. NIST Special Publication 800-22, 2010.
33. *Kadhim M.H., Wijdan R.A.* Binary Sequences Randomness Test Using Neural Networks. International Journal of Computer Science and Mobile Computing. 2015, vol. 4, p. 246–258.
34. *Pol Yee L., De Silva L. C.* Application of MultiLayer Perceptron Network as a one-way hash function. Proceedings of "2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290)", Hawaii, 12 – 17 May 2002, p. 1459–1462.
35. *Anderson J. A., Rosenfeld, E.* Neurocomputing: Foundations of research. Cambridge, USA, The MIT Press, 1988. – 685 p. https://doi.org/10.7551/mitpress/4943.001.0001
36. *Desai V., Patil R., Rao D.* Using Layer Recurrent Neural Network to Generate Pseudo Random Number Sequences. International Journal of Computational Science Issues. 2012, vol. 9, p. 324–334.
37. *Pasqualini L., Parton M.* Pseudo Random Number Generation: a Reinforcement Learning approach. Procedia Computer Science. 2020, vol. 170, p. 1122–1127. https://doi.org/10.1016/j.procs.2020.03.057

References (transliterated)

1. *Apdullah Y., Yakup K.* Neural Network Based Cryptography. Neural Network World. 2014, no. 24, p. 177–192. – https://doi.org/10.14311/NNW.2014.24.011
2. *Kannan M. R. Gnanam V.* Neural network based decryption for random encryption algorithms. Proceedings of 3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication, Hong Kong, 20-22 August 2009, p. 603–605. – https://doi.org/10.1109/ICASID.2009.5277002
3. *Oak R., Rahalkar C., Dhaval G.* Using Generative Adversarial Networks for Secure Pseudorandom Number Generation. Proceedings of 2019 ACM SIGSAC Conference, London, 11–15 November 2019, p. 2597–2599. – https://doi.org/10.1145/3319535.3363265
4. *İnce K.* Security Analysis of Java SecureRandom Library. Avrupa Bilim ve Teknoloji Dergisi. 2021, no. 24, p. 157–160.
5. *Lauria F. E.* On Neurocryptology. Proceedings of the Third Italian Workshop on Parallel Architectures and Neural Networks. Salerno, 15-18 May, 1990. p. 337–343.
6. *Pointcheval D.* Neural networks and their cryptographic applications. Livre des resumes Eurocode Institute for Research in Computer Science and Automation. 1994. p. 1–7.
7. *Schneider B.* Applied Cryptography. Protocols, Algorithms, and Source codes in C. New York, Wiley, 1996. – 758 p.
8. *Luciano D., Prichett G.* Cryptology: From Caesar Ciphers to Public-key Cryptosystems. The College Mathematics Journal. 1987, no. 18(1), p. 2–17. – https://doi.org/10.1080/07468342.1987.11973000
9. *Godhavari T., Alainelu N. R., Soundararajan R.* Cryptography using neural network. IEEE INDICON 2005 Conf., Chennai, India; 11-13 December 2005, p. 258–261. – https://doi.org/10.1080/07468342.1987.11973000
10. *Arvandi M., Wu S., Sadeghian A., Melek W. W., Woungang I.* Symmetric Cipher Design Using Recurrent Neural Networks. The 2006 IEEE International Joint Conference on Neural Network Proceedings, Vancouver, BC, Canada, 2006, p. 2039–2046. https://doi.org/10.1109/IJCNN.2006.246972
11. *Guo D., Cheng L.-M., Cheng L. L.* A new symmetric probabilistic encryption scheme based on chaotic attractors of neural networks. Applied Intelligence. 1999, no. 10(1), p. 71–84. – https://doi.org/10.1023/A:1008337631906
12. *Noughabi A. M. N., Sadeghiyan B.* Design of S-boxes based on neural networks. Proceedings of "International Conference on Electronics and Information Engineering", Kyoto, 1-3 August 2010, p. 172–178. – https://doi.org/10.1109/ICEIE.2010.5559741
13. *Karras D. A., Zorkadis V.* On neural network techniques in the secure management of communication systems through improving and quality assessing pseudorandom stream generators. Neural networks: the official journal of the International Neural Network Society. 2003, no. 16(5-6), p. 899–905. – https://doi.org/10.1016/S0893-6080(03)00124-2
14. *Ruttor A.* Neural Synchronization and Cryptography. PhD thesis. Bayerischen Julius-Maximilians-Universitat at Wurzburg. 2006.
15. *Sadırodlu S., Ozkaya N.* Neural Solutions for Information Security. Journal of Polytechnic. 2007, vol. 10, no. 1, p. 21–25.
16. *Wu X., Han Y., Zhang M. ma ін.* Pseudorandom Number Generators Based on Neural Networks: A Review. Journal of King Saud University – Computer and Information Sciences. 2025. – https://doi.org/10.1007/s44443-025-00007-4
17. *Fei C., Zhang X., Wang D. ma ін.* EPRNG: Effective Pseudo-Random Number Generator on the Internet of Vehicles Using Deep Convolution Generative Adversarial Network. Information. 2025. Vol. 16, № 1. – https://doi.org/10.3390/info16010021
18. *Rivas Becerra M. L., Raygoza Panduro J. J., Cisneros S. O.* Design of a New Neuro-Generator with a Neuronal Module to Produce Pseudorandom Sequences. Electronics. 2024. Vol. 13, № 10. – https://doi.org/10.3390/electronics13101955
19. *Eastlake D. E., Jones P. E.* US Secure Hash Algorithm 1 (SHA1). Request for Comments. 2001, no. 3174, p. 1–22. – https://doi.org/10.17487/rfc3174
20. *Zhang Y. et al.* GAN-Based Pseudo Random Number Generation Optimized through Genetic Algorithms, Complex & Intelligent Systems, 2024.
21. *McCulloch W. S., Pitts W.* A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics. 2003, no. 5(4), p.115–133. – https://doi.org/10.1007/BF02478259
22. *Hassan A. A. et al.* Artificial Neural Network Chaotic PRNG and Encryption on FPGA, Engineering Applications of Artificial Intelligence, 2023.
23. *Vrahatis M., Magoulas G., Parsopoulos K., Plagianakos V.* Introduction to artificial neural network training and applications. Proceedings of 15th Annual Conference of Hellenic Society for Neuroscience, Greece, 27 – 29 October 2000, p. 1–14.
24. *Kingma D. P., Ba J.* Adam: A method for stochastic optimization. Proceedings of International Conference on Learning Representations (ICLR), San Diego, 7 – 9 May 2015, p. 1–13.
25. *Abadi M., Barham P., Chen J.* Tensorflow: A system for large-scale machine learning. Proceedings of 12th USENIX Symposium on Operating Systems Design and Implementation is sponsored by USENIX, Savannah, 2 – 4 November 2016, p. 265–283.
26. *Paszke A., Gross S., Massa F.* et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. Advances in Neural Information Processing Systems, 2019, no. 32, p. 8024–8035.
27. *Chi-Kwong C., Cheng L. M.* The convergence properties of a clipped Hopfield network and its application in the design of key stream generator. IEEE Transactions on Neural Networks. 2001, no. 12, p. 340–348. – https://doi.org/10.1109/72.914528
28. *Wang, Y., Wang, G., Zhang, H.* (2010). Random Number Generator Based on Hopfield Neural Network and SHA-2 (512). In: Luo, Q. (eds) Advancing Computing, Communication, Control and Management. Lecture Notes in Electrical Engineering, vol. 56. Springer, Berlin, Heidelberg. p. 198–205. – https://doi.org/10.1007/978-3-642-05173-9_26
29. *Desai V. V., Deshmukh V. B., Rao D. H.* Pseudo Random Number Generator Using Elman. Proceedings of 2011 IEEE Recent Advances

in Intelligent Computational Systems, Trivandrum, 22-24 September 2011, p. 251–254. – https://doi.org/10.1109/RAICS.2011.6069312
30. Sutton, R.; Barto, A. Reinforcement Learning: An Introduction. IEEE Transactions on Neural Networks, vol. 9, no. 5, p. 1054–1054, Sept. 1998. – https://doi.org/10.1109/TNN.1998.712192
31. *L'Ecuyer P., Simard R.* TestU01: A C library for empirical testing of random number generators. ACM Transactions on Mathematical Software. 2007, vol. 33, no. 4, p. 22–40. – https://doi.org/10.1145/1268776.1268777
32. *Rukhin A., Soto J., Nechvatal J.* et. al. A statistical test suite for random and pseudorandom number generators for cryptographic applications. NIST Special Publication 800-22, 2010.
33. *Kadhim M. H., Wijdan R. A.* Binary Sequences Randomness Test Using Neural Networks. International Journal of Computer Science and Mobile Computing. 2015, vol. 4, p. 246–258.
34. *Pol Yee L., De Silva L. C.* Application of MultiLayer Perceptron Network as a one-way hash function. Proceedings of "2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290)", Hawaii, 12 – 17 May 2002, p. 1459–1462.
35. *Anderson J. A., Rosenfeld, E.* Neurocomputing: Foundations of research. Cambridge, USA, The MIT Press, 1988. – 685 p. – https://doi.org/10.7551/mitpress/4943.001.0001
36. *Desai V., Patil R., Rao D.* Using Layer Recurrent Neural Network to Generate Pseudo Random Number Sequences. International Journal of Computational Science Issues. 2012, vol. 9, p. 324–334.
37. *Pasqualini L., Parton M.* Pseudo Random Number Generation: a Reinforcement Learning approach. Procedia Computer Science. 2020, vol. 170, p. 1122–1127. – https://doi.org/10.1016/j.procs.2020.03.057

Надійшла (received) 26.11.2025
Прийнята до друку (accepted) 19.11.2025
Опублікована (published) 29.12.2025

Відомості про авторів/ About the Authors

Некрасова Марія Володимирівна (Nekrasova Mariia) – кандидат технічних наук, доцент, Національний технічний університет «Харківський політехнічний інститут», доцент кафедри комп'ютерного моделювання процесів та систем; м. Харків, Україна; тел.: (057)-707-64-54; ORCID: https://orcid.org/0009-0006-9285-0740; e-mail: masha12dec@gmail.com