МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

*Кваліфікаційна наукова праця*
*на правах рукопису*

**ПРОСКУРІН ДМИТРО ПЕТРОВИЧ**

УДК 003.26: 629.7.05

# ДИСЕРТАЦІЯ

# ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ОЦІНЮВАННЯ ЯКОСТІ ГЕНЕРАТОРІВ ПОСЛІДОВНОСТЕЙ ПСЕВДОВИПАДКОВИХ ЧИСЕЛ НА ОСНОВІ МАШИННОГО НАВЧАННЯ

122 «Комп’ютерні науки»

12 «Інформаційні технології»

Подається на здобуття ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____Д.П.Проскурін

Наукові керівники:

**Гнатюк Сергій Олександрович**

доктор технічних наук, професор

**Явіч Максим Павлович**

PhD, професор

Київ – 2024

# АНОТАЦІЯ

*Проскурін Дмитро Петрович*. Інформаційна технологія оцінювання якості генераторів послідовностей псевдовипадкових чисел на основі машинного навчання. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття ступеня доктора філософії з галузі знань 12 «Інформаційні технології» за спеціальністю 122 «Комп'ютерні науки». – Національний авіаційний університет, Київ, 2024.

Актуальність теми дослідження полягає в необхідності забезпечення високого рівня надійності та безпеки інформаційних систем у сучасному світі, де кіберзагрози постійно зростають. Генератори послідовностей псевдовипадкових чисел (ГППВЧ) є ключовими компонентами криптографічних алгоритмів, які використовуються в телекомунікаціях, фінансових послугах та державних службах. Сучасні методи оцінювання якості ГППВЧ вимагають великих обсягів даних і значних обчислювальних ресурсів, що ускладнює їх застосування в умовах обмежених ресурсів.

У дисертації запропоновано нову інформаційну технологію, яка використовує методи машинного навчання, зокрема гібридні та згорткові нейронні мережі, для підвищення точності та швидкості оцінювання якості ГППВЧ навіть за умов обмеженої кількості вхідних даних. Це забезпечує нові можливості для використання генераторів у реальних умовах, де доступ до великих обсягів даних може бути обмеженим, і гарантує високий рівень надійності та безпеки інформаційних систем.

Наукова новизна роботи полягає в розробці моделі ідентифікації джерела послідовностей псевдовипадкових чисел на основі гібридної нейронної мережі та малоресурсної інформаційної технології, що дозволяє проводити комплексне оцінювання якості генераторів. Запропоновані підходи дозволяють виявляти неякісні та ненадійні генератори, а також здійснювати точне передбачення наступних послідовностей псевдовипадкових чисел.

Практична цінність роботи полягає у можливості застосування отриманих результатів для криптографії, стільникових мереж LTE/5G/6G, технологій на основі UAV, а також для захисту критичної інформаційної інфраструктури. Результати дослідження були впроваджені у навчальний процес Національного авіаційного університету та в діяльність науково-дослідних лабораторій.

В першому розділі було проведено аналіз існуючих підходів до генерування та оцінювання якості послідовностей псевдовипадкових чисел, визначення ефективності методів і засобів штучного інтелекту в контексті вирішення зазначених завдань.

В другому розділі було розроблено та досліджено модель ідентифікації джерела послідовностей псевдовипадкових чисел, для виявлення генераторів (якими були сформовані послідовності) в умовах обмеженої кількості вхідних даних. Також було удосконалено модель передбачення наступної послідовності псевдовипадкових чисел з високою точністю.

В третьому розділі було розвинуто та дослідити метод оцінювання якості послідовностей псевдовипадкових чисел з використанням алгоритмів штучного інтелекту для застосувань в галузі комп’ютерних наук.

В четвертому розділі розроблено інформаційну технологію для комплексного оцінювання якості генераторів послідовностей псевдовипадкових чисел в умовах обмеженої кількості вхідних даних.

Дисертація складається з вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. Повний обсяг дисертації становить 130 сторінок, з них 110 – основного тексту. Робота містить 41 рисунків, 20 таблиць і 20 додатків. Список використаних джерел налічує 80 найменувань.

**Ключові слова**: згорткова нейронна мережа, рекурентна нейронна мережа, гібридна нейронна мережа, генератор псевдовипадкових

послідовностей, хі-квадрат, машинне навчання.

СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Статті проіндексовано в наукометричній базі Scopus:

1. Proskurin D., Gnatyuk S., Okhrimenko T., Iavich M. ML-Based Cryptographic Keys Quality Assessment for 5G / 6G Networks Privacy and Security, Proceedings of the IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS. 2023. С. 1025-1030.

2. Gnatyuk S., Okhrimenko A., Navrotskyi D., Proskurin D., Horbakha B. Dataset of Cryptographic Algorithms for UAV Image Encryption based on Artificial Neural Networks, CEUR Workshop Proceedings. 2023. Вип. 3504. С. 63-71.

3. Hu Z., Ryabyy M., Prystavka P., Janisz K., Proskurin D. Advanced Method for Compressing Digital Images as a Part of Video Stream to Pre-processing of UAV Data Before Encryption, Lecture Notes on Data Engineering and Communications Technologies. 2023. Вип. 181. С. 371-381.

4. Proskurin D., Gnatyuk S., Okhrimenko T. Predicting Pseudo-Random and Quantum Random Number Sequences using Hybrid Deep Learning Models, CEUR Workshop Proceedings. 2023. Вип. 3426. С. 77-88.

5. Proskurin D., Gnatyuk S., Bauyrzhan M. Distributive Training Can Improve Neural Network Performance based on RL-CNN Architecture, CEUR Workshop Proceedings. 2021. Вип. 3187. С. 48-57.

Статті у фахових виданнях:

1. Рябий М., Кінзерявий О., Проскурін Д., Сорокопуд В. An advanced method of compressing digital images as part of a video stream to pre-process the data before encrypting, Проблеми інформатизації та управління. 2023. Т. 1, № 73. С. 128-137.

2. Гнатюк С.О., Поліщук Ю.Я., Кінзерявий В.М., Горбаха Б.М., Проскурін Д.П. Формування датасету криптоалгоритмів для забезпечення конфіденційності даних, які передаються з розвідувально-пошукового

БПЛА, Кібербезпека: освіта, наука, техніка. 2023. № 4 (20). С. 205–219.

3. Проскурін Д.П., Явіч М.П., Гнатюк С.О. Модель ідентифікації джерела послідовностей псевдовипадкових чисел на основі гібридної нейронної мережі, Проблеми інформатизації та управління. 2024. Т. 1, № 73. С. 54-62.

Тези доповідей на конференціях:

1. Проскурін Д.П., Гнатюк С.О. Дистрибутивне навчання покращує роботу нейронних мереж на основі RL-CNN архітектури, ABIA-2021: XVI міжнар. наук.-техн. конф., 20-22 квітня 2021 р.: тези доп. Київ: НАУ, 2021. С. 16.14-16.17.

2. Гнатюк С.О., Проскурін Д.П. Імплементація дистрибутивного навчання покращує роботу RL-CNN архітектури для ідентифікації об'єктів на зображеннях // Всеукраїнська науково-практична інтернет-конференція здобувачів вищої освіти і молодих учених «Інформаційно-комп’ютерні технології: стан, досягнення та перспективи розвитку», 25-26 листопада 2021, Житомир, Україна.

3. Proskurin D. P. Assessing Randomness in Number Sequences in Cryptography: A Comparative Study of the Chi-Squared Test and Neural Network-Based Approaches, EEML 2023: Eastern European Machine Learning Conference, June 2023.

4. Проскурін Д.П., Гнатюк С.О. Підхід до оцінювання рівня випадковості бінарних послідовностей на основі одновимірної згорткової нейронної мережі, Information, Communication, Society (ICS-2023), 18-20 травня 2023 р., Зозулі (Львівська область), Україна.

5. Проскурін Д.П., Гнатюк С.О. Оцінювання випадковості бінарних послідовностей на основі одновимірної згорткової нейронної мережі 1D-CNN для криптографічних застосувань // Новітні дослідження культури і мистецтва: пошуки, проблеми, перспективи : матеріали Всеукр. наук.-практ. конф. / М-во культ. України та інформ. політики ; Нац. акад.

кер. кадрів культ. і мистец. ; Наук. тов. студ., асп., доктор. і молод. вч. (Київ, 18 травня 2023 р.). Київ : НАКККіМ, 2023. С. 27-32.

**ЗМІСТ**

# Вступ

**Актуальність теми дослідження та її зв'язок із планами науково-дослідних робіт.**

У сучасному світі стійкість інформаційних систем є критично важливою, особливо в контексті зростаючої кількості кіберзагроз та необхідності захисту критичної інфраструктури. Генератори послідовностей псевдовипадкових чисел (ГППВЧ) є ключовими компонентами в багатьох протоколах безпеки, що використовуються в різних галузях, включаючи телекомунікації, фінансові послуги та державні служби.

Проте, існуючі методи оцінювання якості ГППВЧ часто вимагають великих обсягів даних і значних обчислювальних потужностей, що може бути проблематичним в умовах обмежених ресурсів. У цьому контексті розробка нових методів, моделей та інформаційних технологій на основі машинного навчання для швидкого і точного оцінювання якості ГППВЧ набуває особливої актуальності.

Запропонована у роботі інформаційна технологія використовує сучасні методи штучного інтелекту, включаючи гібридні та згорткові нейронні мережі, що дозволяє суттєво підвищити точність та швидкість оцінювання якості генераторів навіть за умови обмеженої кількості вхідних даних. Це відкриває нові можливості для їх застосування в реальних умовах, де доступ до великої кількості даних може бути обмеженим, і забезпечує високий рівень стійкості та безпеки інформаційних систем.

Результати роботи пов'язані з НДР №0122U002361 «Інтелектуалізована система захищеного передавання пакетних даних на базі розвідувально-пошукового безпілотного літального апарату», що фінансується МОН України і виконується згідно планів НДР Національного авіаційного університету протягом 2022-2024 років, а також дослідницьким грантом NFR-22-14060 «AI-based multilayer 5G security assurance methodology for the needs of special groups of subscribers in Georgia», що фінансується Shota Rustaveli National Foundation of Georgia.

Тема дисертації відповідає освітньо-науковій програмі “Комп’ютерні науки” за спеціальністю 122 “Комп’ютерні науки” галузі знань 12 “Інформаційні технології” в Національному авіаційному університеті.

**Мета і завдання дослідження.**

Метою роботи є забезпечення швидкого та точного оцінювання генераторів послідовностей псевдовипадкових чисел на основі розроблених методів, моделей та інформаційної технології із застосуванням машинного навчання.

Для цього сформульовано комплекс наступних науково-технічних задач:

1. Аналіз існуючих підходів до генерування та оцінювання якості послідовностей псевдовипадкових чисел, визначення ефективності методів і засобів штучного інтелекту в контексті вирішення зазначених завдань;

2. Розробити та дослідити модель ідентифікації джерела послідовностей псевдовипадкових чисел, для виявлення генераторів (якими були сформовані послідовності) в умовах обмеженої кількості вхідних даних;

3. Удосконалити та дослідити модель передбачення наступної послідовності псевдовипадкових чисел з високою точністю;

4. Розвинути та дослідити метод оцінювання якості послідовностей псевдовипадкових чисел з використанням алгоритмів штучного інтелекту для застосувань в галузі комп’ютерних наук;

5. Розробити інформаційну технологію для комплексного оцінювання якості генераторів послідовностей псевдовипадкових чисел в умовах обмеженої кількості вхідних даних

*Об'єкт дослідження* – процес оцінювання якості генераторів послідовностей псевдовипадкових чисел.

*Предмет дослідження* – методи, моделі та інформаційні технології

оцінювання якості генераторів послідовностей псевдовипадкових чисел в умовах обмеженої кількості вхідних даних на основі машинного навчання.

У дисертаційній роботі вирішено науково-прикладну задачу щодо забезпечення комплексного оцінювання якості генераторів послідовностей псевдовипадкових чисел в умовах обмеженої кількості вхідних даних із застосуванням машинного навчання.

**Наукові положення, розроблені особисто здобувачем, та їх новизна** полягають у тому, що:

*вперше*

▪ розроблено модель ідентифікації джерела послідовностей псевдовипадкових чисел, що за рахунок використання гібридної нейронної мережі та обмеженої кількості вхідних даних для навчання та оптимізації, дає можливість виявляти генератори, якими були сформовані послідовностей псевдовипадкових чисел;

▪ розроблено малоресурсну інформаційну технологію, яка за рахунок використання моделей ідентифікації джерела послідовності псевдовипадкових чисел і передбачення наступної послідовності, а також методу оцінювання якості послідовностей псевдовипадкових чисел, дає можливість здійснювати комплексне оцінювання якості генераторів послідовностей псевдовипадкових чисел в умовах обмеженої кількості вхідних даних, виявляти неякісні, ненадійні і скомпрометовані генератори;

*удосконалено:*

▪ модель передбачення наступної послідовності псевдовипадкових чисел, що за рахунок використання гібридної нейронної мережі та обмеженої кількості вхідних даних для навчання, дозволяє передбачати чергові послідовності для неякісних генераторів псевдовипадкових чисел;

*отримав подальшого розвитку* :

▪ метод оцінювання якості послідовностей псевдовипадкових чисел, який за рахунок використання одновимірної рекурентної нейронної мережі та датасетів, сформованих різними генераторами псевдовипадкових чисел, дозволяє більш швидко оцінювати якість генераторів для криптографічних та інших застосувань в галузі комп'ютерних наук;

**Обґрунтованість і достовірність наукових положень, висновків, рекомендацій, які захищаються**.

Наукові положення, висновки й рекомендації, сформульовані в дисертації, відповідають вимогам до такого виду досліджень. Високий рівень обґрунтованості наукових положень, висновків, рекомендацій сформульованих у дисертації, їхня достовірність забезпечені:

– професійним вирішенням автором низки наукових завдань, що сприяло реалізації поставленої мсти дослідження, та адекватністю структурно-логічної схеми дослідження визначеній меті: кожен наступний розділ чи підрозділ органічно пов'язаний із попереднім і доповнює його;

– використанням широкої бази наукових джерел за темою дисертації й достатнім масивом аналітичних даних.

**Наукове значення роботи** полягає у вирішенні актуальної науково-технічної задачі щодо комплексного аналізу статистичних характеристик генераторів послідовностей псевдовипадкових чисел на основі машинного навчання.

**Практичне значення та використання результатів** дисертаційного дослідження полягає у тому, що отримані результати можуть бути використані відповідними органами для ефективного оцінювання генераторів псевдовипадкох послідовностей, крім цього було:

▪ Використання гібридної нейронної мережі (HNN) дало можливість дослідити 8 реалізованих генераторів та 4 000 згенерованих послідовностей (для навчання та оптимізації) і ідентифікувати на основі нових для HNN даних з усередненою точністю 87,14% джерела

послідовностей псевдовипадкових чисел;

▪ Використання гібридної нейронної мережі (HNN) дало можливість більш точно передбачати наступні послідовності псевдовипадкових чисел у порівнянні з рекурентною нейронною мережею (RNN) та згортковою нейронною мережею (CNN) – на 2,52% та 1,44% відповідно;

▪ Реалізація одновимірної згорткової нейронної мережі (1D-CNN) та генераторів псевдовипадкових чисел середовища розробки Python підтвердила можливість на 40% швидшого оцінювання якості генераторів у порівнянні з методом chi-квадрат (з незмінним рівнем точності P-value), а також можливість аналізу менших за довжиною послідовностей

▪ Зазначені результати лягли в основу малоресурсної інформаційної технології, що дозволяє здійснювати комплексне оцінювання якості генераторів послідовностей псевдовипадкових чисел в умовах обмеженої кількості вхідних даних, виявляти неякісні, ненадійні і скомпрометовані генератори;

▪ Отримані результати будуть корисні для криптографії, стільникових мереж LTE / 5G / 6G, технологій на основі UAV, захисту критичної інформаційної інфраструктури держави. Результати були впроваджені в навчальний процес кафедри комп’ютерних інформаційних технологій Факультету комп’ютерних наук та технологій (акт впровадження №03 від 09.05.2024) і в наукову діяльність Науково-дослідної лабораторії протидії кіберзагрозам в авіаційній галузі Національного авіаційного університету (акт впровадження 30.11.2024) і Головного управління розвідки Міністерства оборони України.

**Повнота викладення матеріалів дисертації в публікаціях та особистий внесок у них автора.** Дисертація «Інформаційна технологія оцінювання якості генераторів послідовностей псевдовипадкових чисел на основі машинного навчання» Проскуріна Дмитра Петровича є

самостійною науковою працею, в якій наведено теоретичні і практичні положення, висновки, власні ідеї та розробки автора, які дають змогу вирішити поставлені завдання. Усі висновки та практичні рекомендації, винесені на захист, розроблені дисертантом особисто.

Апробація результатів дисертації. Основні результати дисертаційної роботи були представлені та обговорені на дванадцяти міжнародних науково- технічних та науково-практичних конференціях:

1. Міжнародна науково-технічна конференція «ABIA» (Київ, 2021)
2. Information, Communication, Society, (Зозулі, 2023)
3. Інформаційно-комп’ютерні технології: стан, досягнення та перспективи розвитку, (Житомир, 2021)
4. IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS (Дортмунд, 2023)
5. Proceedings of the Workshop on Classic, Quantum, and Post-Quantum Cryptography (CQPC 2023) co-located with International Conference on Problems of Infocommunications. Science and Technology (PICST 2023) (Kyiv, 2023)
6. Modern Machine Learning Technologies and Data Science Workshop, (MoMLeT&DS) (Львів, 2021)
7. Modern Machine Learning Technologies and Data Science Workshop, (MoMLeT&DS) (Львів, 2023)
8. Eastern European Machine Learning Summer School (Кошице, 2023).

Основні положення та результати дисертаційного дослідження викладено

у 11.5 наукових працях, у тому числі: 1.5 статті, опубліковані у наукових виданнях, включених до переліку фахових видань України, 5 – в зарубіжних наукових виданнях, включених до наукометричної бази Scopus; 5 – у матеріалах тез доповідей на науково-практичних конференціях різного рівня.

**Структура та обсяг дисертації.** Дисертація складається з анотації, вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. Повний обсяг дисертації становить 185 сторінок, із них 160 – основного тексту. Робота містить 45 рисунків, 20 таблиць, 10 додатків. Список використаних джерел налічує 80 найменувань.

**Оцінка мови та стилю дисертації.** Текст дисертації викладено грамотною мовою, логічно та послідовно. Матеріали дослідження викладені з дотриманням вимог наукового стилю. Дисертація оформлена згідно з вимогами Міністерства освіти і науки України.

**Характеристика особистості здобувача.** Під час підготовки дисертаційної роботи здобувач проявив себе як висококваліфікований та творчий дослідник, здатний самостійно вирішувати наукові та практичні завдання. Володіє сучасними методами аналізу та має глибокі знання у своїй галузі дослідження. Здобувач відповідальний, дисциплінований, активно бере участь у наукових заходах і демонструє високий рівень аналітичного мислення та комунікативних навичок.

# РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ ПІДХОДІВ ДО ГЕНЕРУВАННЯ ТА ОЦІНЮВАННЯ ЯКОСТІ ПОСЛІДОВНОСТЕЙ ПСЕВДОВИПАДКОВИХ ЧИСЕЛ, ВИЗНАЧЕННЯ ЕФЕКТИВНОСТІ МЕТОДІВ І ЗАСОБІВ ШТУЧНОГО ІНТЕЛЕКТУ В КОНТЕКСТІ ВИРІШЕННЯ ЗАЗНАЧЕНИХ ЗАВДАНЬ

## 1.1. Огляд сучасних методів генерування псевдовипадкових чисел

Сьогодні нейронні мережі використовуються для вирішення багатьох бізнес-завдань, таких як прогнозування продажів, дослідження клієнтів, перевірка даних, управління ризиками, виявлення аномалій і навіть розуміння природної мови. Вони можуть бути тією частиною технологічного розвитку, яка зараз має найбільший потенціал. На додаток до загальної архітектури існують інтегровані реалізації з використанням RL, або розширеного навчання, які визначають, які дії інтелектуальні агенти повинні виконувати в певному середовищі, щоб максимізувати поняття сукупної винагороди.

Поєднання криптографії та штучного інтелекту (AI) відкриває нові можливості для покращення безпеки, конфіденційності та ефективності в багатьох сферах, включаючи кібербезпеку, обробку даних і автоматизацію. Ось кілька ключових аспектів, які підкреслюють важливість їх поєднання: покращення безпеки; захист конфіденційності; Автоматизація та оптимізація; Розробка квантово-стійкої криптографії; Боротьба з фішингом і шахрайством; Розумні контракти та блокчейн; Підвищена довіра; Адаптація до нових загроз; Оптимізація використання ресурсів; Розвиток нових технологій.

Інтеграція ШІ та криптографії відкриває великі перспективи для забезпечення безпеки, конфіденційності та ефективності в сучасному цифровому світі, адаптації до його швидких змін і нових викликів. Аналізуючи сучасні наукові дослідження щодо синтезу ШІ та криптографії або методів криптоаналізу, необхідно відзначити наступні

запропоновані підходи:

• спільне використання генетичних алгоритмів та алгоритмів асиметричної криптографії (Zolfaghari B, Koshiba T.)

• механізми інтеграції ШІ та блокчейн технологій (B. Chavali, S. K. Khatri and S. A. Hossain)

• теорія так званої «нейронної криптографії», в якій алгоритми криптографічної обробки даних і розподілу ключів базуються на алгоритмах синхронізації нейронних мереж (T. Dong and T. Huang)

• оцінка (псевдо)випадковості згенерованих послідовностей для криптографічних програм з використанням ШІ (Y. Feng and L. Hao)

• концепція штучного інтелекту під впливом криптографії та криптографії під впливом штучного інтелекту (B. Zolfaghari, T. Koshiba)

З іншого боку, наразі існує багато публікацій, пов’язаних із використанням ШІ для завдань криптоаналізу, щоб вибрати найбільш ефективну криптоаналітичну атаку на основі можливостей зловмисника (криптоаналітика) та характеристик перехоплених даних (ключові фрагменти, зашифрований текст тощо) (Y. Xiao, Q. Hao and D. D. Yao)

Однак, незважаючи на значну кількість досліджень у цій галузі, багато важливих аспектів залишаються невивченими або недостатньо вивченими, включаючи створення сучасних методів PQC, керованих ШІ; інтеграція ШІ в підсистеми управління ключами шифрування; оцінка якості ключа та стійкості шифру тощо. Цей дослідницький проект буде зосереджений на цих актуальних і цінних аспектах

Хоча існуючі підходи для аналізу якості генераторів псевдовипадкових чисел, такі як методи NIST, забезпечують високу точність і надійність, їх застосування вимагає великої обчислювальної потужності і тривалих обчислювальних процесів (як зазначається в «*National Institute of Standards and Technology. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic*

*Applications»*).

З іншого боку, технології машинного навчання мають потенціал для значного покращення процесу оцінки якості послідовностей псевдовипадкових чисел. Вони можуть бути використані для аналізу якості послідовностей (*Y. Feng and L. Hao*) і навіть для їх передбачення (*Glauco Amigo et al*.). Такі підходи можуть прискорити процес і зробити його більш ефективним.

Проте, існуючі методи машинного навчання мають свої обмеження. По-перше, вони вимагають великої кількості даних для тренування. Наприклад, для отримання високої точності в передбаченні псевдовипадкових послідовностей потрібно до 1 000 000 зразків (*Glauco Amigo et al*.). По-друге, багато з цих методів мають обмежені можливості для практичного застосування, особливо в реальних умовах, де отримана послідовність може не вказувати на конкретний генератор або мати інші невизначеності.

Отже, необхідно провести детальний аналіз існуючих підходів до генерування та оцінювання якості послідовностей псевдовипадкових чисел, а також визначити ефективність методів і засобів штучного інтелекту в контексті вирішення зазначених завдань. Такий аналіз дозволить виявити переваги і недоліки кожного підходу, а також розробити рекомендації щодо їх оптимального використання в різних сценаріях.

Кілька досліджень досліджували використання методів глибокого навчання для прогнозування послідовностей випадкових чисел. Наприклад, повідомлялося про використання CNN і LSTM для прогнозування послідовностей PRNG [23]. В іншому дослідженні RNN були використані для прогнозування послідовностей QRNG [24]. Однак існує обмежене дослідження гібридних моделей глибокого навчання, які поєднують декілька архітектур нейронних мереж для прогнозування послідовностей PRNG і QRNG.

Основною метою цього дослідження є дослідження ефективності різних архітектур глибокого навчання, включаючи моделі MLP, CNN, LSTM і RNN, для завдання прогнозування наступного значення в послідовності випадкових чисел, згенерованих комбінацією PRNG і QRNG. джерела. Досліджуючи різні архітектури нейронних мереж, ми мали на меті визначити найбільш прийнятну модель для цієї проблеми, враховуючи такі аспекти, як точність прогнозування, складність моделі та час навчання.

Інша мета полягає в тому, щоб оцінити, чи можуть навчені моделі досягти кращих результатів прогнозування, ніж випадкова базова лінія, що свідчить про те, що вони засвоїли значущі моделі в даних. Щоб забезпечити справедливе порівняння, будуть використані відповідні показники оцінки, такі як середня квадратична помилка (MSE) і середня абсолютна помилка (MAE), для кількісної оцінки ефективності кожної моделі та порівняння її з еталонним показником випадкового прогнозу.

І, нарешті, метою було дати розуміння практичних наслідків використання моделей глибокого навчання для прогнозування послідовностей випадкових чисел, згенерованих із квантових джерел, а також обговорити потенційні майбутні напрямки досліджень у цій галузі. Розуміючи сильні сторони та обмеження різних моделей для цього завдання, автори сподіваються зробити внесок у розробку більш досконалих методів аналізу та прогнозування послідовностей випадкових чисел у різних контекстах.

На останні досягнення в прогнозуванні послідовності суттєво вплинула розробка та вдосконалення рекурентних нейронних мереж (RNN) і довготривалої короткочасної пам’яті (LSTM). Ці моделі показали надзвичайну майстерність у обробці послідовних даних, особливо в областях, де розуміння часової динаміки має вирішальне значення.

LSTM для прогнозування часових рядів. Дослідження продемонстрували ефективність моделей LSTM у прогнозуванні часових рядів, області, де традиційно домінують статистичні методи, такі як ARIMA. На відміну від цих методів, LSTM можуть фіксувати складні нелінійні зв’язки в даних часових рядів [39, 41]. Дослідники успішно застосували моделі LSTM для прогнозування цін на акції, попиту на енергію та погодних умов, досягнувши вищої точності, ніж традиційні моделі, особливо в сценаріях із довгостроковими залежностями та високою волатильністю.

RNN в обробці природної мови (NLP): RNN відіграли ключову роль у просуванні NLP. Їх здатність обробляти послідовні текстові дані призвела до прориву в машинному перекладі, генерації тексту та аналізі настроїв [38, 39, 42]. Можливість послідовної обробки RNN дозволяє їм підтримувати контекст у тексті, критичний фактор для розуміння людської мови [42]. Однак ванільні RNN часто борються з довгостроковими залежностями [46], що призводить до прийняття LSTM і GRU (Gated Recurrent Units) у більш складних завданнях NLP.

3. Навчання від послідовності до послідовності: структура навчання від послідовності до послідовності, часто реалізована за допомогою LSTM, революціонізувала такі завдання, як машинний переклад. Цей підхід передбачає навчання моделей на парах вихідних і вихідних послідовностей, що дозволяє моделі вивчати відображення однієї послідовності в іншу. Ця структура була вирішальною для розробки моделей , які можуть перекладати цілі речення з контекстом, а не перекладати слово за словом [39].

4. Проблеми та обмеження: незважаючи на їхні успіхи, RNN та LSTM не позбавлені проблем. Проблема зникнення градієнта в RNN, де модель втрачає здатність вивчати довгострокові залежності, була частково вирішена LSTM, але все ще має обмеження [41]. Крім того, навчання цих моделей може бути обчислювально інтенсивним, вимагаючи значних ресурсів для великих наборів даних.

5. Майбутні напрямки: поточні дослідження вивчають ефективніші та результативні варіанти RNN та LSTM, такі як механізми уваги та моделі трансформатора. Ці розробки спрямовані на усунення існуючих обмежень, одночасно збільшуючи здатність моделей обробляти довші послідовності та підтримувати контекст протягом тривалих періодів.

## 1.2. Аналіз методів оцінювання якості послідовностей псевдовипадкових чисел

Оцінка випадковості числових послідовностей є темою, яка цікавить дослідників і практиків у різних галузях, включаючи криптографію, статистичну вибірку та комп’ютерне моделювання [63, 64, 66]. Не можна недооцінювати важливість випадковості в цих програмах, оскільки вона безпосередньо впливає на надійність, безпеку та точність базових систем. Протягом багатьох років було розроблено численні методи оцінки якості послідовностей випадкових чисел [63, 66].

Статистичні тести, такі як тест хі-квадрат, були традиційним вибором для оцінки випадковості числових послідовностей. Тест хі-квадрат порівнює спостережуваний розподіл частот даної послідовності з очікуваним розподілом за припущення справжньої випадковості. Хоча цей метод довів свою ефективність у багатьох випадках, він має деякі обмеження, такі як чутливість до розміру вибірки та вибору групування, що може призвести до хибнопозитивних або хибнонегативних результатів у деяких випадках [64].

Зі швидким прогресом машинного навчання та штучного інтелекту зростає інтерес до вивчення потенціалу нейронних мереж для оцінки

випадковості [62, 68, 70]. Нейронні мережі, зокрема згорткові нейронні мережі (CNN) і рекурентні нейронні мережі (RNN), продемонстрували надзвичайний успіх у широкому діапазоні задач розпізнавання образів і класифікації. Здатність цих мереж вивчати складні шаблони з даних і їх адаптованість до різних проблемних областей робить їх перспективними кандидатами для оцінки випадковості числових послідовностей [68, 70].

### 1.3. Використання штучного інтелекту для оцінювання якості псевдовипадкових чисел

Кілька досліджень досліджували використання нейронних мереж для оцінки випадковості, хоча дослідження в цій галузі все ще знаходяться на початковій стадії [72, 73]. Деякі дослідження були зосереджені на застосуванні CNN до проблеми, розглядаючи числові послідовності як одновимірні «зображення», тоді як інші досліджували RNN на їхню здатність фіксувати тимчасові залежності в послідовностях [72, 73]. Ці дослідження показали багатообіцяючі результати, але повне порівняння з традиційними статистичними методами, такими як тест хі-квадрат, ще не проведено.

Поточне дослідження має на меті заповнити цю прогалину шляхом порівняння ефективності тесту хі-квадрат і підходів на основі нейронної мережі в оцінці випадковості числових послідовностей, надаючи цінну інформацію про потенційні переваги та обмеження цих методів.

### 1.4. Сучасні тенденції та проблеми в контексті генерування та оцінювання псевдовипадкових чисел

Для ефективного порівняння ефективності тесту хі-квадрат і підходів на основі нейронної мережі в оцінюванні випадковості числових послідовностей ми встановили такі показники та критерії оцінки:

Точність: основним показником оцінки є точність класифікації кожного методу в ідентифікації випадкових і невипадкових послідовностей. Вища точність вказує на кращу продуктивність у розрізненні випадкових і невипадкових послідовностей.

Час обчислення: також слід враховувати час, який витрачається кожним методом на оцінку випадковості числових послідовностей. Швидший метод є кращим, особливо коли ви маєте справу з великими наборами даних або коли потрібна оцінка в реальному часі.

Масштабованість: оцініть, як змінюється продуктивність кожного методу зі збільшенням розміру набору даних. Більш масштабований метод повинен підтримувати або покращувати свою продуктивність при застосуванні до більших наборів даних.

Надійність: оцініть здатність кожного методу узагальнювати різні типи послідовностей випадкових чисел із різною довжиною, розподілом і характеристиками. Більш надійний метод має добре працювати в широкому діапазоні типів послідовностей.

Можливість інтерпретації: хоча і не є прямим показником ефективності, здатність до інтерпретації може бути важливим фактором при виборі методу оцінки випадковості. Здатність зрозуміти базовий процес прийняття рішень методу може бути цінним у певних додатках.

**Висновки**

У цій статті було представлено нову гібридну модель глибокого навчання, яка поєднує в собі сильні сторони згорткових нейронних мереж (CNN), мереж довготривалої короткочасної пам’яті (LSTM) і RNN для прогнозування генератора псевдовипадкових чисел (PRNG) і квантових випадкових чисел. Генератор (QRNG) послідовностей. Наші результати демонструють, що гібридна модель перевершує традиційні моделі CNN, LSTM і RNN з точки зору точності прогнозування для послідовностей PRNG і QRNG.

У цьому розділі проведено детальний аналіз сучасних методів генерування та оцінювання якості послідовностей псевдовипадкових чисел (ПВЧ), а також розглянуто використання штучного інтелекту (ШІ) в контексті вирішення цих завдань.

Огляд сучасних методів генерування псевдовипадкових чисел: Визначено, що сучасні методи генерування ПВЧ включають лінійні конгруентні генератори, генератори на основі зсувних регістрів та криптографічні генератори. Кожен з цих методів має свої переваги та обмеження, що визначають їх застосування в різних галузях.

Аналіз методів оцінювання якості послідовностей псевдовипадкових чисел: Оцінювання якості ПВЧ здійснюється через статистичні тести, такі як тест на серійність, тест на розподіл, тест на автокореляцію та інші. Встановлено, що ці тести часто вимагають великих обсягів даних і значних обчислювальних ресурсів.

Використання штучного інтелекту для оцінювання якості псевдовипадкових чисел: Проаналізовано сучасні підходи до використання ШІ для оцінювання якості ПВЧ. Зокрема, розглянуто методи, що базуються на глибокому навчанні, зокрема гібридні нейронні мережі та згорткові нейронні мережі, які дозволяють суттєво підвищити точність та швидкість оцінювання навіть за умов обмеженої кількості вхідних даних.

Сучасні тенденції та проблеми в контексті генерування та оцінювання псевдовипадкових чисел: Виявлено, що однією з головних тенденцій є інтеграція методів ШІ у процеси генерування та оцінювання ПВЧ. Основні проблеми включають необхідність великих обсягів даних для навчання моделей ШІ та високу обчислювальну складність, що обмежує їх застосування в умовах обмежених ресурсів.

Таким чином, у розділі 1 визначено, що використання сучасних методів ШІ для оцінювання якості генераторів ПВЧ є перспективним напрямом, який дозволяє значно підвищити ефективність та надійність процесів генерування та оцінювання ПВЧ. Однак, для реалізації цих підходів необхідні подальші дослідження та розробка нових методів, які б зменшували залежність від великих обсягів даних та знижували обчислювальні витрати.

У рамках майбутньої роботи планується вивчити інші архітектури гібридної моделі, які могли б ще більше підвищити продуктивність нашої поточної моделі. Також була спрямована на дослідження використання додаткових функцій, таких як інформація з частотної області, для покращення можливостей прогнозування нашої моделі. Крім того, автори мають намір вивчити можливість узагальнення нашої гібридної моделі для інших завдань прогнозування послідовності. Ці завдання можуть включати передбачення криптографічних ключів, безпечних протоколів зв’язку та інших програм, пов’язаних із безпекою. Крім того, буде слід розглянути розробку більш надійних та ефективних стратегій навчання, щоб гарантувати, що запропонована модель залишається ефективною навіть перед обличчям загроз безпеці, що швидко розвиваються. Продовжуючи покращувати й удосконалювати нашу гібридну модель глибокого навчання, автори сподіваються зробити внесок у розвиток безпечного зв’язку та захисту даних у цифрову епоху.

# РОЗДІЛ 2. РОЗРОБКА МОДЕЛЕЙ ОЦІНКИ ЯКОСТІ ГЕНЕРАТОРІВ НА ОСНОВІ ТЕХНОЛОГІЙ МАШИННОГО НАВЧАННЯ В УМОВАХ ОБМЕЖЕНОЇ КІЛЬКОСТІ ВХІДНИХ ДАНИХ

## 2.1. Огляд алгоритмів машинного навчання, придатних для оцінювання якості псевдовипадкових чисел

Генерація випадкових чисел є ключовим компонентом багатьох програм, включаючи криптографію, системи безпечного зв'язку, моделювання та ймовірнісні алгоритми. Генератори псевдовипадкових чисел (PRNG) і квантові генератори випадкових чисел (QRNG) є двома основними типами генераторів випадкових чисел, причому QRNG забезпечують кращу безпеку через притаманну їм непередбачуваність [21]. Однак прогнозування послідовностей PRNG і QRNG залишається важливим завданням для оцінки їх безпеки та надійності. Методи глибокого навчання, такі як згорткові нейронні мережі (CNN), мережі довгострокової короткочасної пам'яті (LSTM) і RNN , широко використовуються в різних задачах прогнозування часових рядів [22]. У цій статті була запропонована гібридна модель глибокого навчання, яка поєднує CNN, LSTM і RNN для прогнозування послідовностей PRNG і QRNG. Модель навчається та оцінюється на наборі даних, що містить послідовності PRNG і QRNG.

Набір даних, використаний у цьому дослідженні, складається з послідовностей PRNG і QRNG, згенерованих за допомогою різних алгоритмів, таких як Twister Mersenne , лінійний конгруентний генератор і комерційний пристрій QRNG [25]. Набір даних розділено на набори для навчання, перевірки та тестування, що забезпечує збалансоване представлення послідовностей PRNG і QRNG у кожному наборі.

Запропонована гібридна модель глибокого навчання поєднує в собі сильні сторони CNN, LSTM і RNN для прогнозування послідовностей PRNG і QRNG. Модель складається з рівня CNN для вилучення ознак, за

яким слідує рівень LSTM для захоплення тимчасових залежностей і рівень RNN для захоплення довгострокових залежностей. Кінцевим виходом є один лінійний блок активації, який виробляє прогнозоване значення. Модель навчена за допомогою оптимізатора Адама та середньоквадратичної помилки (MSE) як функції втрат [26].

Першим кроком у аналізі ефективності моделі було візуальне порівняння прогнозованих значень із справжніми значеннями. Це було досягнуто шляхом побудови перших 100 справжніх значень і відповідних прогнозованих значень на одному графіку. Ця візуалізація дозволяє нам оцінити загальну відповідність моделі даним і виявити будь-які помітні розбіжності між прогнозованими та справжніми значеннями.

### 2.2. Розробка моделі оцінювання якості на основі машинного навчання

Щоб кількісно визначити подібність між справжніми значеннями та прогнозованими значеннями, був розрахований коефіцієнт кореляції Пірсона. Цей показник вимірює лінійний зв’язок між двома наборами даних, причому значення, близьке до 1, вказує на сильний позитивний зв’язок. Попередньо встановлений поріг 0,9 використовувався для визначення того, чи прогнозовані значення вважаються близькими до справжніх значень. На основі обчисленого коефіцієнта кореляції було зроблено висновок про те, чи були прогнози моделі близькі до справжніх значень чи ні.

### 2.3. Особливості роботи з обмеженою кількістю вхідних даних

Щоб оцінити ефективність моделі, її продуктивність порівнювали з базовою лінією випадкового прогнозу. Це було зроблено шляхом створення випадкових прогнозів у тому ж діапазоні, що й справжні значення, та обчислення середньої квадратичної помилки (MSE) як для прогнозів моделі, так і для випадкових прогнозів. Порівнюючи ці значення MSE, ми змогли визначити, чи прогнози моделі GRU були кращими за випадкові.

Результати візуального огляду, оцінки подібності та порівняння продуктивності моделі забезпечують всебічний аналіз ефективності моделі для прогнозування наступного значення в послідовності випадкових чисел. Ці висновки сприяють нашому розумінню ефективності моделі для цього конкретного завдання та пропонують розуміння можливих удосконалень або альтернативних підходів.

Гібридна модель навчається на наборі даних, і її продуктивність порівнюється з традиційними RNN, CNN і LSTM. Моделі оцінюються з використанням коефіцієнта кореляції Пірсона та середньої квадратичної помилки (MSE), щоб оцінити подібність між справжніми та прогнозованими значеннями.

Проста RNN є найосновнішою формою рекурентної нейронної мережі, яка характеризується своїм єдиним прихованим шаром, який отримує вхідні дані з попереднього часового кроку та повертає їх у мережу для наступного часового кроку (рис. 1).

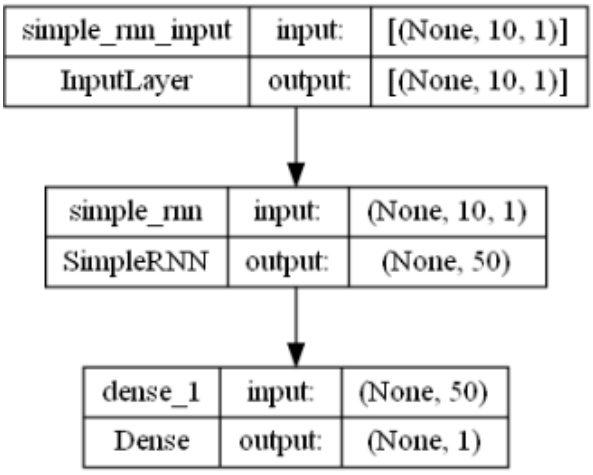



Рис. 1: Архітектура RNN

Незважаючи на свою простоту, продуктивність простих РНМ у передбаченні послідовностей PRNG і QRNG обмежена через їхню нездатність охоплювати довгострокові залежності в результаті проблеми зникнення градієнта (рис. 2).

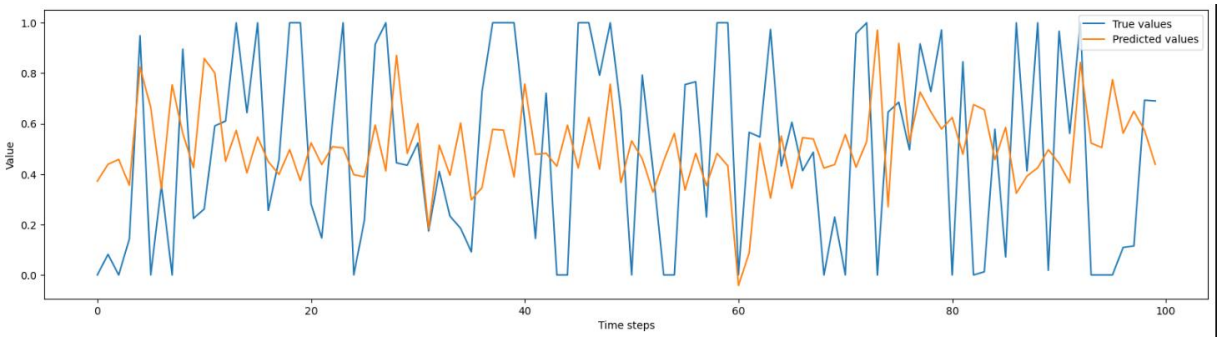



Рис. 2: Результати RNN

GRU — це вдосконалена архітектура RNN, яка вирішує проблему зникнення градієнта, яка спостерігається в простих RNN. Завдяки впровадженню механізмів стримування ГРУ можуть дізнаватися, коли оновлювати прихований стан, а коли підтримувати існуючий стан, що дозволяє їм більш ефективно фіксувати довгострокові залежності (рис. 3).

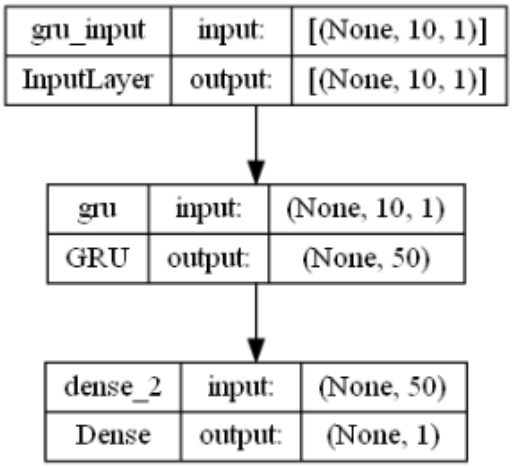



Рис. 3: Архітектура GRU

У застосуванні до прогнозування послідовності PRNG і QRNG GRU демонструють покращену продуктивність порівняно з простими RNN (рис. 4).

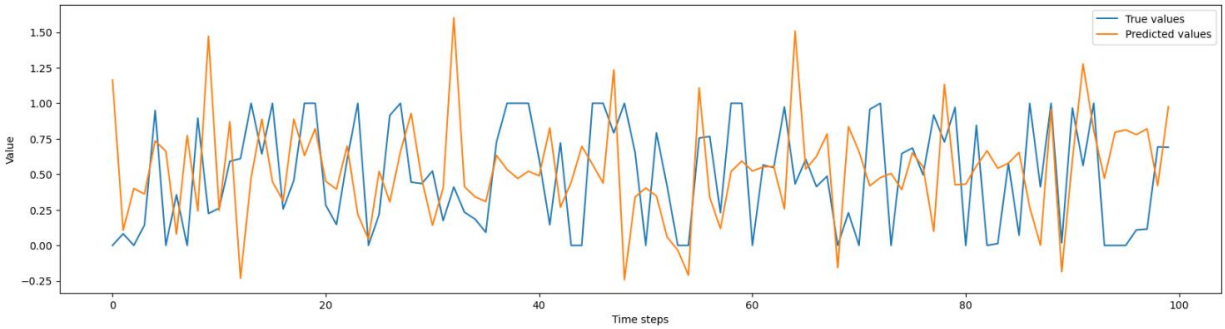



Рис. 4: Результати GRU

Двонаправлені RNN обробляють вхідну послідовність як у прямому, так і в зворотному напрямках, дозволяючи мережі отримувати інформацію як з минулих, так і з майбутніх часових кроків. Ця можливість виявляється корисною для завдань, де важливий контекст з обох сторін, наприклад, обробка природної мови та розпізнавання мовлення (рис. 5).

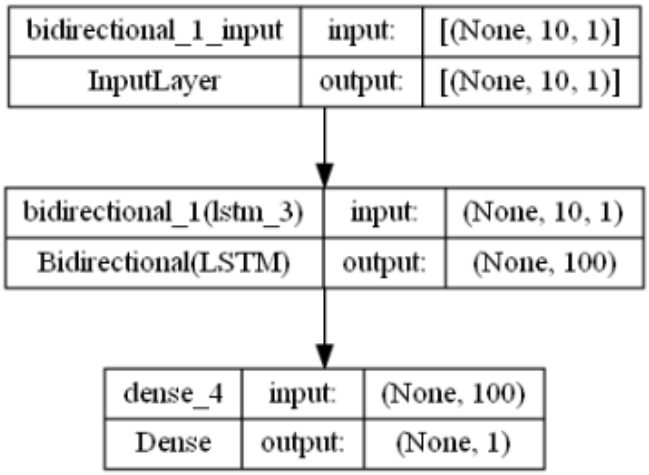



Рис. 5: Архітектура LSTM

У контексті передбачення послідовності PRNG і QRNG двонаправлені RNN демонструють підвищену продуктивність завдяки своїй здатності включати інформацію з усієї послідовності (рис. 6).

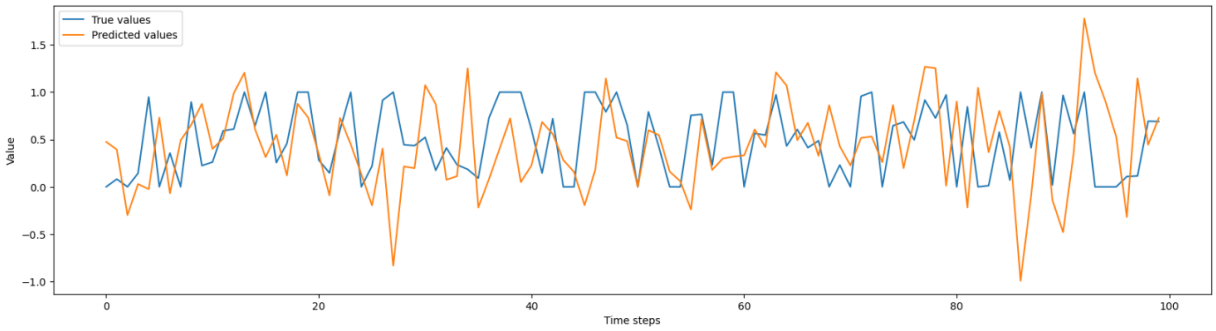



Рис. 6: Результати LSTM

Багатошарова архітектура RNN складається з кількох шарів RNN, розташованих один на одному, що дозволяє мережі вивчати більш складні функції та представлення вхідної послідовності. Ця підвищена складність може призвести до покращення продуктивності передбачення для послідовностей PRNG і QRNG (рис. 7).

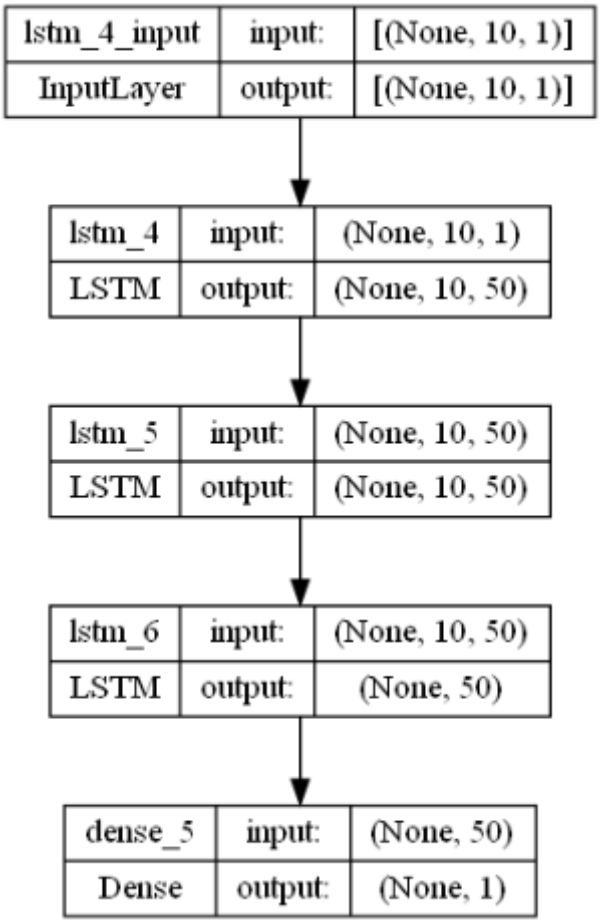



Рис. 7: Архітектура Deep RNN

У порівнянні з іншими варіантами RNN, складені RNN, демонструють чудову продуктивність у захопленні складних шаблонів у

вхідних даних (рис. 8).

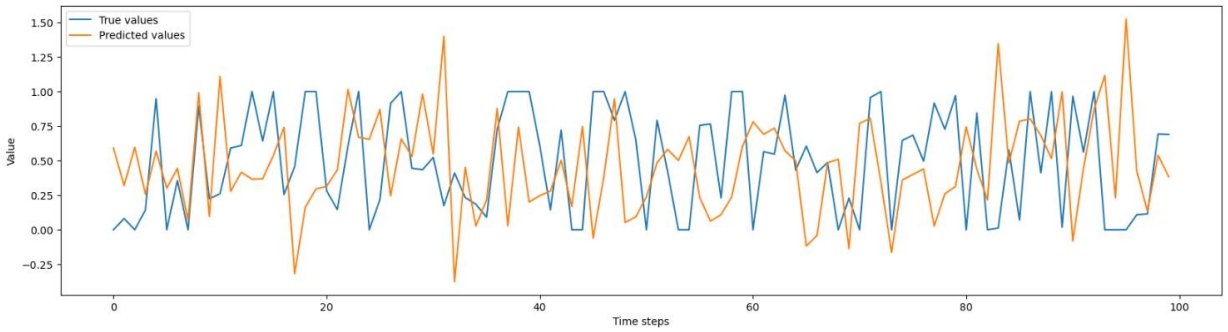



Рис. 8: Результати Deep RNN

CNN показали успіх у задачах прогнозування часових рядів завдяки своїй здатності вловлювати локальні моделі та залежності [27]. У наших експериментах модель CNN тренується на наборі даних послідовностей PRNG і QRNG (рис. 9) .

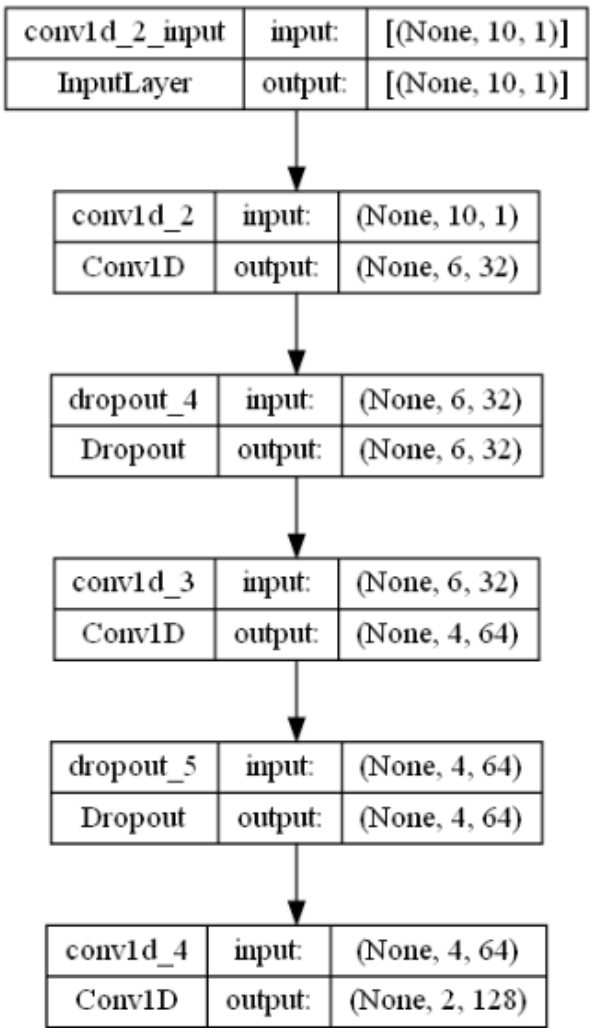



Рис. 9: Архітектура Deep CNN

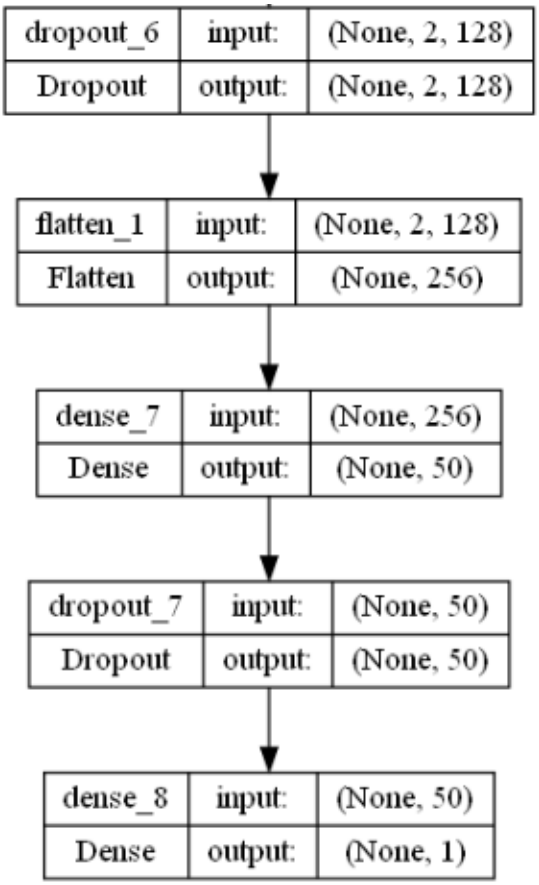



Рис. 10: Архітектура Deep CNN з регулярізацію

Результати показують, що модель CNN може вловити деякі локальні шаблони в послідовностях, але важко передбачити довгострокові залежності, що призводить до неоптимальної точності прогнозу (рис. 10) .

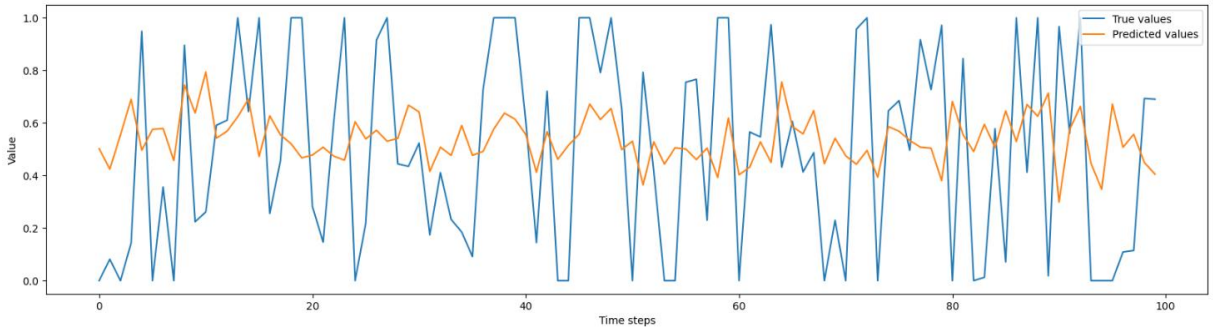



Рис. 11: Результати Deep CNN

LSTM призначені для фіксації довгострокових залежностей у даних часових рядів [28-30]. Було навчено модель LSTM на наборі даних послідовностей PRNG і QRNG і оцінено її продуктивність (рис. 11) .

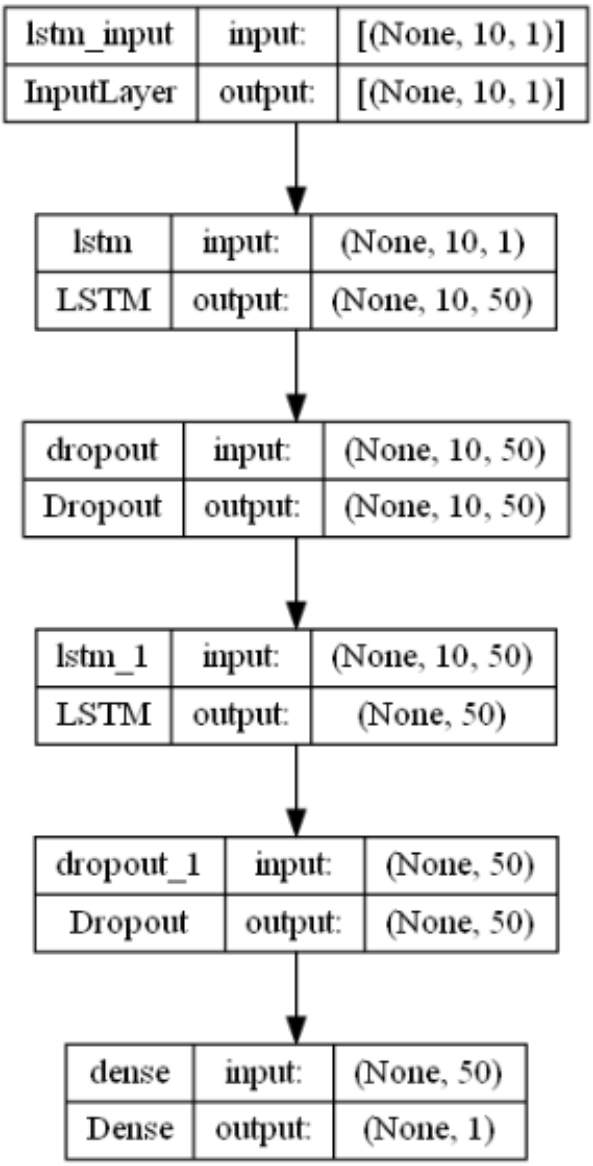



Рис. 12: Архітектура Deep LSTM

Результати показують, що модель LSTM може фіксувати тимчасові залежності в послідовностях, але її продуктивність обмежена відсутністю можливостей вилучення ознак (рис . 12) [31] .

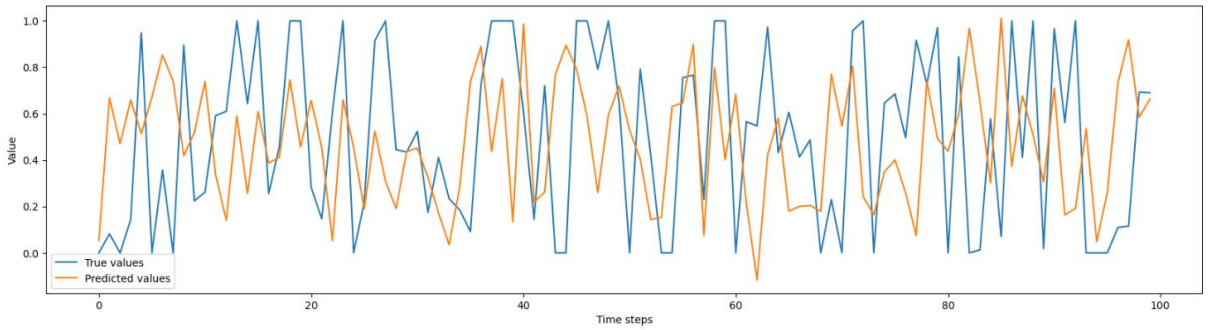



Рис. 13: Результати Deep LSTM

Запропонована гібридна модель поєднує в собі сильні сторони CNN, LSTM і RNN для прогнозування послідовностей PRNG і QRNG (рис. 13) [32, 33] .

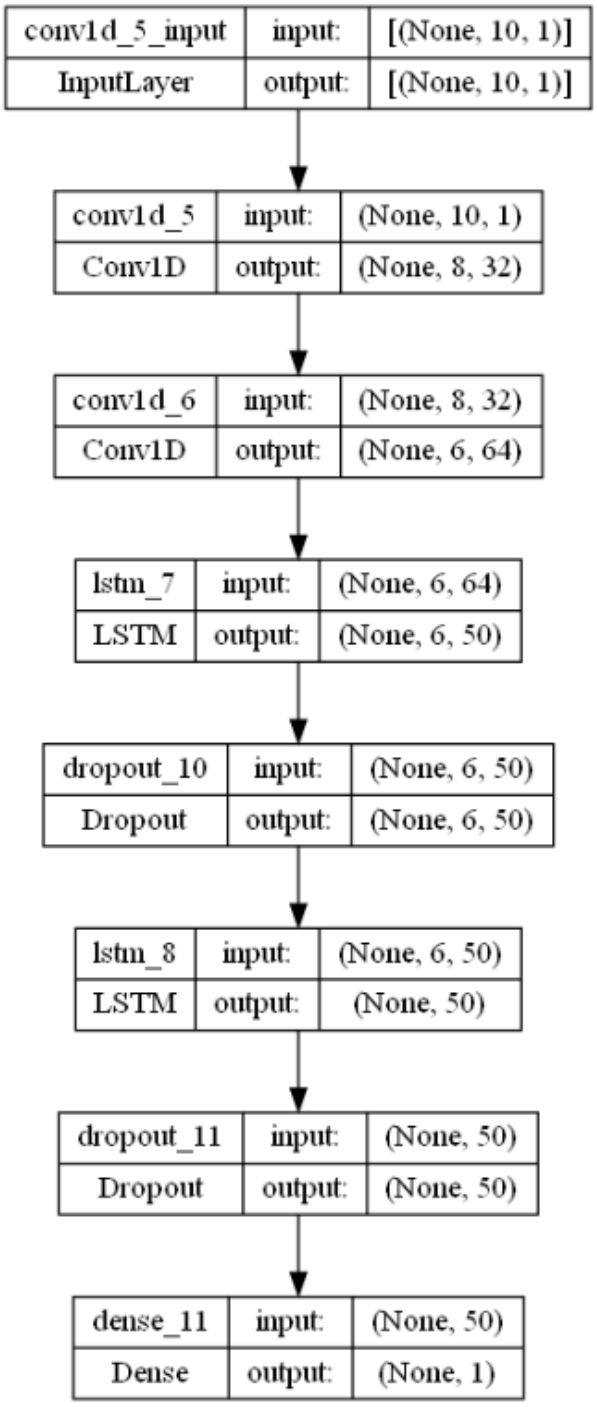



Рис. 14: Результати Deep HNN

Ефективність моделі порівнюється з іншими моделями, і результати показують, що гібридна модель перевершує традиційні моделі CNN, LSTM і RNN, забезпечуючи кращу точність передбачення для послідовностей PRNG і QRNG (рис. 14) .

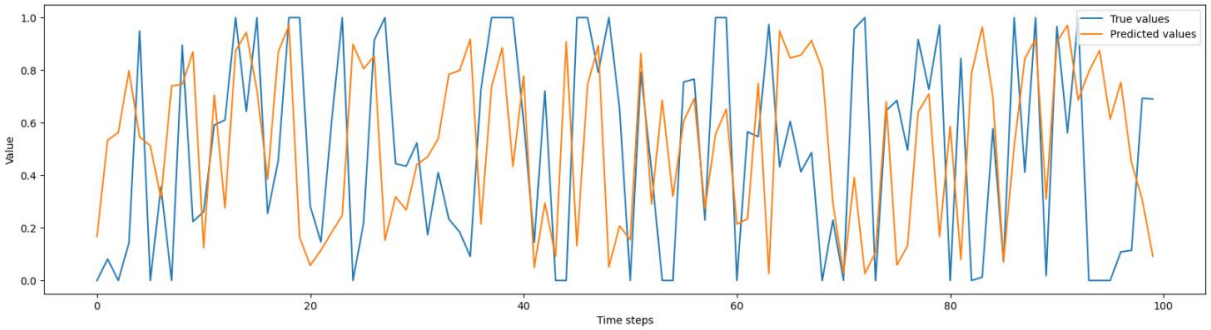



Рис. 15: Результати Deep HNN

Можна спостерігати численні випадки, коли моделі могли передбачити точне значення або дуже близьку тенденцію в послідовностях PRNG і QRNG (рис. 15, 16). Ці приклади демонструють ефективність моделей у розумінні базових закономірностей і залежностей у даних, а також їх здатність узагальнювати та робити точні прогнози на невидимих даних.

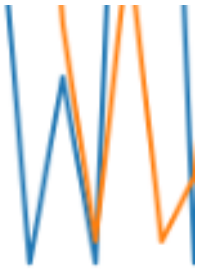


*Рис. 16: Точний постійний збіг*

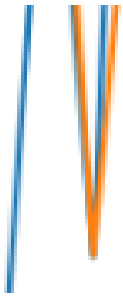


*Рис. 17: Точний збіг*

Крім того, було помічено, що моделі часто здатні передбачити близьку тенденцію в послідовностях, навіть якщо точне значення не було визначено (рис. 17) .

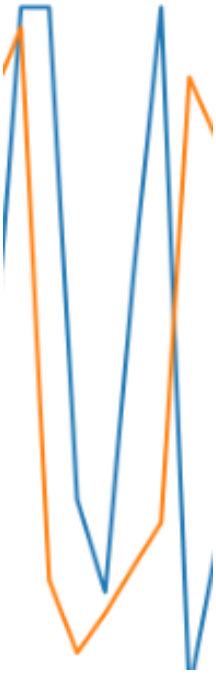


*Рис. 18: Збіг тренду*

Це вказує на те, що моделі добре розуміють загальну динаміку та структуру даних, що дозволяє їм генерувати прогнози, які точно відповідають фактичній траєкторії послідовностей PRNG та QRNG [34-36]. Цей рівень ідентифікації тенденції може виявитися корисним у сценаріях, де розуміння загального напрямку чи шаблону даних важливіше, ніж визначення окремих значень [37].

У середовищі машинного навчання, що постійно розвивається, здатність точно передбачати майбутні події на основі послідовних даних є наріжним каменем численних технологічних досягнень і програм. Від прогнозування тенденцій фондового ринку до декодування людської мови, значення ефективного передбачення послідовності неможливо переоцінити. Центральними в цьому домені є рекурентні нейронні мережі (RNN) і мережі довгострокової короткочасної пам'яті (LSTM), які стали потужними інструментами в арсеналі машинного навчання для обробки послідовних даних.

RNN, відомі своєю унікальною архітектурою, яка дозволяє інформації зберігатися, відіграли важливу роль у моделюванні залежних

від часу даних [38]. Однак їхнє застосування часто затьмарюється проблемами, такими як проблема зникнення градієнта, яка перешкоджає вивченню довгострокових залежностей [39]. Введіть LSTM, особливий вид RNN, розроблений спеціально для подолання цих обмежень. Завдяки своїм складним внутрішнім механізмам LSTM встановили нові стандарти в задачах прогнозування послідовності, демонструючи надзвичайний успіх там, де традиційні RNN дають збитки. [37, 39]

Ця стаття розпочинає комплексне дослідження RNN та LSTM у контексті передбачення послідовності. Ми заглиблюємося в архітектурні тонкощі цих моделей, їхні сильні та слабкі сторони, а також їх продуктивність у різних сценаріях передбачення послідовності. Наше дослідження особливо зосереджено на наборах даних, згенерованих різними генераторами псевдовипадкових чисел (PRNG), пропонуючи унікальну лінзу, через яку можна перевірити та зрозуміти можливості цих моделей. Шляхом ретельних експериментів і аналізу ми прагнемо пролити світло на нюанси передбачення послідовності та надати інформацію, яка може скеровувати майбутні застосування та дослідження в цій захоплюючій галузі машинного навчання.

### 2.4. Верифікація та тестування моделей

Нейронні мережі — це моделі штучного інтелекту, які імітують роботу людського мозку. Нейронна мережа з’єднує процесори, подібні до нейронів, а не маніпулює нулями та одиницями, як це робить цифрова модель. Результат залежить від того, як зв'язки організовані та зважені. Нейронні мережі — це алгоритми, створені за зразком людського мозку, які розпізнають шаблони. Сенсорні дані інтерпретуються за допомогою машинного сприйняття, яке позначає або кластеризує необроблену інформацію. Вони розпізнають числові шаблони у векторах, які мають бути перетворені в дані реального світу, такі як зображення, звуки, текст і часові ряди [57]. Штучні нейронні мережі (ШНМ) — це обчислювальні системи, створені за зразком біологічних нейронних систем, у тому числі людського мозку. ШНМ в основному складаються з великої кількості

взаємопов’язаних обчислювальних вузлів (відомих як нейрони), які працюють разом у розподіленій формі, щоб спільно навчатися на виході з метою оптимізації кінцевого виходу [58]

Згорткові нейронні мережі (CNN) подібні до стандартних штучних нейронних мереж (ANN) тим, що вони використовують нейрони для самовдосконалення шляхом навчання [58]. CNN досягли видатних досягнень. Зараз ця нейронна мережа широко використовується в глибокому навчанні. Згорткові нейронні мережі революціонізували комп’ютерне бачення, уможлививши такі раніше немислимі досягнення, як розпізнавання обличчя, безпілотні автомобілі, супермаркети самообслуговування та інтелектуальні медичні процедури. CNN також відрізняються від типових ШНМ тим, що зосереджуються на розпізнаванні образів зображень. Це дозволяє нам кодувати специфічні для зображення властивості в архітектуру, роблячи мережу краще пристосованою для завдань, орієнтованих на зображення, а також зменшуючи кількість параметрів, необхідних для налаштування моделі [58, 59]:

$$y_t^{(k)} = \sigma\left(\sum_{i=0}^{W-1}\sum_{j=0}^{F-1} w_{ij}^{(k)} x_{t+ij} + b^{(k)}\right)$$

Гібридні нейронні мережі (HNN), які об’єднують сильні сторони багатьох нейронних мереж, стають все більш популярними в програмах комп’ютерного бачення, включаючи субтитри до зображень та ідентифікацію дій. Проте було проведено обмежені дослідження щодо ефективного використання гібридних архітектур для даних часових рядів, особливо для цілей прогнозування трендів [60]. HNN використовують свою внутрішню структуру, щоб обмежити взаємодію між змінними процесу з метою узгодження з фізичними моделями. Порівняно зі звичайними нейронними мережами, пов’язані моделі є більш точними, надійними та узагальненими [61]:

$$y_t = \sigma\left(W_y h_t + c\right)$$

Повторювані нейронні мережі (RNN) являють собою зміну парадигми в нейронних мережах, спеціально розроблених для розпізнавання шаблонів у послідовностях даних [45]. На відміну від традиційних нейронних мереж прямого зв’язку, RNN мають унікальну особливість: вихідні дані попереднього кроку повертаються на вихідні дані поточного кроку. Цей циклічний механізм дозволяє RNN підтримувати внутрішній стан, який фіксує інформацію про послідовність, яку вони обробили до цього моменту, що робить їх ідеальними для таких завдань, як розпізнавання мови, моделювання мови та прогнозування часових рядів [39, 45].

Основна архітектура RNN включає прихований рівень, де активація на певному кроці часу є функцією виходу на тому ж кроці та активації прихованого рівня на попередньому кроці [46]. Така повторювана природа дозволяє мережі підтримувати певну форму пам’яті [45]. Однак RNN часто стикаються з довгостроковими залежностями через такі проблеми, як зникнення та збільшення градієнтів під час зворотного поширення [37, 39], коли мережа стає нездатною вивчати та зберігати інформацію з попередніх часових кроків у послідовності [41].

Мережі довготривалої короткочасної пам’яті, особливий вид RNN, були розроблені, щоб подолати обмеження традиційних RNN. LSTM вміють вивчати довгострокові залежності завдяки своїй унікальній внутрішній структурі [37]. На відміну від стандартних RNN, LSTM мають складну архітектуру з серією вентилів: вентиль забуття, вихідний вентиль і вихідний вентиль [37, 41]. Ці ворота регулюють потік інформації в клітину та з неї, вирішуючи, що зберігати в пам’яті, а що відкинути, таким чином вирішуючи проблему зникнення градієнта [41].

- Forget Gate: Визначає, яка інформація скидається зі стану комірки [40, 41].
- Output Gate: оновлює стан комірки новою інформацією з поточного виводу [40].

- Output Gate: Визначає наступний прихований стан і вихід на основі поточного виводу та оновленого стану комірки [40].

Ця архітектура дозволяє LSTM приймати більш точні рішення про те, яку інформацію зберігати, змінювати та виводити. Як наслідок, LSTM були успішно застосовані в різних складних задачах моделювання послідовності, включаючи машинний переклад, синтез мови та навіть у генеративних моделях для композиції музики [37, 40, 41].

Хоча і RNN, і LSTM розроблені для обробки послідовностей, ключова відмінність полягає в їхній здатності обробляти довгострокові залежності. Стандартні RNN, хоч і простіші та менш інтенсивні з точки зору обчислень, мають проблеми зі збереженням інформації в довших послідовностях. LSTM з їх складним механізмом стримування відмінно підходять у сценаріях, коли розуміння довгострокової контекстної інформації має вирішальне значення.

Вибір між RNN і LSTM часто зводиться до конкретних вимог поставленого завдання, складності задіяних послідовностей і доступних обчислювальних ресурсів. LSTM, як правило, є кращими для більш складних завдань із довшими послідовностями [37], тоді як RNN може бути достатньо для більш простих завдань із коротшими часовими залежностями [38].

У нашому дослідженні ми використовували набори даних, створені чотирма різними алгоритмами PRNG, кожен з яких пропонує унікальні проблеми та характеристики для прогнозування послідовності за допомогою моделей RNN і LSTM. Ці набори даних служать тестовим полігоном для оцінки та порівняння продуктивності різних архітектур нейронних мереж у задачах прогнозування послідовності.

### 2.4.1 LCG

Опис: LCG є одним із найстаріших і найпростіших алгоритмів PRNG [49]. Він генерує випадкові числа за допомогою лінійного рівняння [49]. Простота його алгоритму робить його хорошою базою для оцінки

прогностичних можливостей моделей RNN і LSTM.

Характеристики: послідовність, згенерована LCG, може демонструвати закономірності через свою лінійну природу. Ці закономірності, хоч і не відразу очевидні, можна дізнатися з часом, що робить їх цікавим випадком для моделей передбачення послідовності [49]. Незважаючи на їхні потенційні статистичні проблеми, LCG мають перевагу, пропонуючи всі допоміжні якості, такі як можливість пошуку, численні потоки та k-вимірний рівномірний розподіл [49].

### 2.4.2 MT

Опис: Mersenne Twister, зокрема варіант MT19937, відомий своїм довгим терміном роботи та високою якістю. Він широко використовується в різних сферах застосування завдяки своїй надійності та швидкості [51].

Характеристики: MT генерує послідовності, які набагато складніші та менш передбачувані, ніж LCG [49]. Ця складність створює складний сценарій для RNN і LSTM, перевіряючи їхню здатність моделювати та передбачати більш складні та, здавалося б, випадкові послідовності. На додаток до своєї нездатності створювати повністю нульовий стан, Мерсенну Твістеру також важко діяти випадково у своєму майже повністю нульовому стані [49].

### 2.4.3 XS

Опис: Xorshift — це клас PRNG, який працює з використанням операцій XOR (виключне або) та операцій зсуву бітів [49]. Він відомий своєю простотою та швидкістю, часто використовується в сценаріях, де швидкість генерації випадкових чисел є критичною [49].

Характеристики: незважаючи на свою простоту, Xorshift може виробляти високоякісні випадкові послідовності [49]. Нелінійний характер його операцій робить його цікавим випадком для вивчення того, наскільки добре моделі нейронних мереж можуть адаптуватися до нелінійних алгоритмів і прогнозувати їх результати [54]. Порозрядна операція xor — це тип перестановки, який передбачає перевертання певних бітів у

цільовому файлі. Його можна повторити, щоб усунути ефекти [49]. Традиційне розуміння Xorshift порадило б нам зосередитися на подовженні періоду бітів [49].

### 2.4.4 MS

Опис. Метод середнього квадрата — це старіший метод PRNG, який генерує випадкові числа шляхом зведення числа в квадрат і виділення середніх цифр результату [54]. Сьогодні він використовується рідше через певні обмеження.

Характеристики: цей метод схильний до швидкого переходу до повторюваних циклів або нулів, особливо з певними вихідними значеннями. Передбачуваність і потенційне повторення в послідовностях роблять його унікальним набором даних для перевірки здатності моделей виявляти та адаптуватися до менш складних і потенційно дегенеративних моделей. Сфера інформатики почалася з винаходу середнього квадрата. Завдяки сучасній 64-розрядній архітектурі можна розробити життєздатну версію з досить тривалим періодом (264 кожен потік). Найшвидші RNG можна порівняти за швидкістю обробки. Цей генератор добре працює для паралельної обробки завдяки своїй можливості простого потоку. Оскільки квадрат нелінійний, він забезпечує цьому генератору перевагу над лінійними генераторами з точки зору якості даних [49].

У нашому дослідженні «Прогнозування виходу PRNG за допомогою послідовного аналізу» ми ретельно підготували набір даних для аналізу передбачуваності різних генераторів псевдовипадкових чисел (PRNG), зосередившись на чотирьох широко визнаних алгоритмах: лінійному конгруентному генераторі (LCG), MiddleSquare, Xorshift і Mersenne Twister (MT). Кожне з цих PRNG було обрано за його унікальний підхід до генерації послідовностей псевдовипадкових чисел, забезпечуючи різноманітний тестовий стенд для наших прогнозних моделей.

Набір даних було створено з використанням таких параметрів, щоб забезпечити узгодженість у всіх PRNG:

Розмір вибірки: кожен PRNG використовувався для створення послідовності з 10 000 чисел з n = 10 000, щоб створити значний набір даних для навчання та оцінки. Початкове значення: загальне початкове значення 8956482 було застосовано для ініціалізації кожного PRNG, забезпечуючи узгодженість початкової точки псевдовипадкової послідовності для різних генераторів. Розмір слова: для PRNG, де це застосовно, наприклад MiddleSquare, було обрано розмір слова 8 біт, врівноважуючи потребу в обчислювальній ефективності з бажанням складності послідовності. Довжина послідовності: вихідні дані були сегментовані на послідовності довжиною 10, які потім використовувалися як окремі точки даних для наступного аналізу. Цю довжину послідовності було обрано, щоб надати достатньо даних для розпізнавання патернів без перевантаження аналітичних моделей.

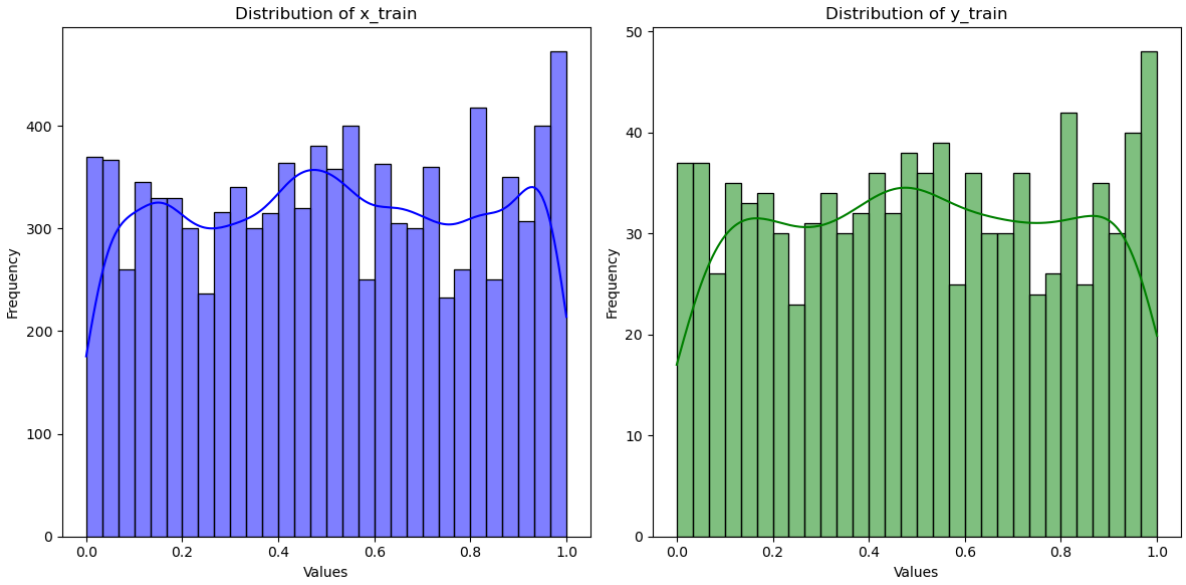



*Рис. 19: Набір даних для тренування від LCG*

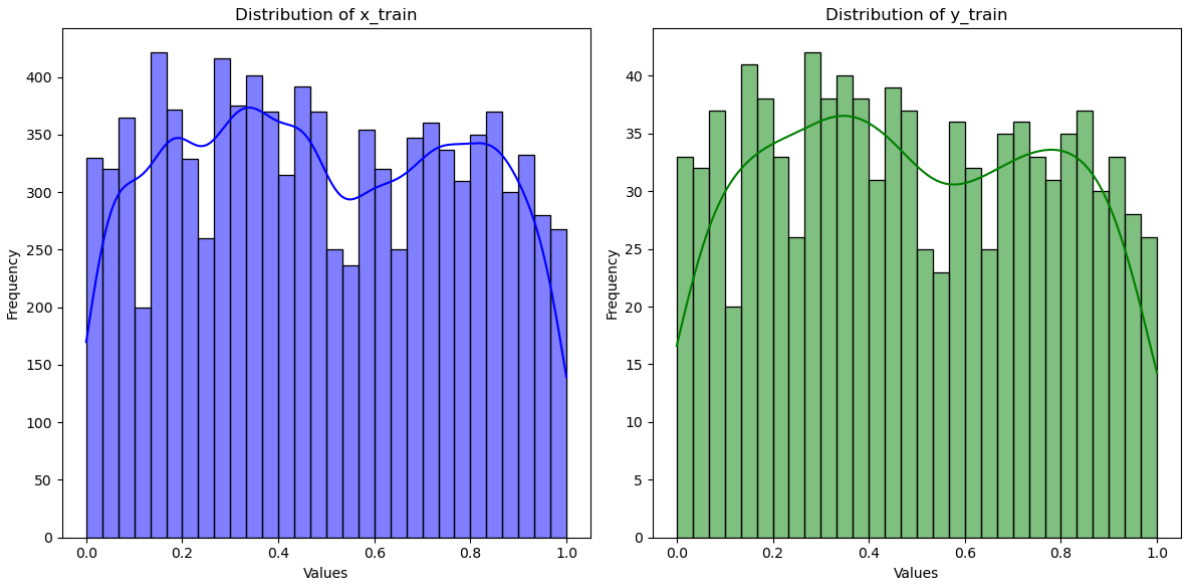



*Рис. 20: Набір даних для тренування від MT*

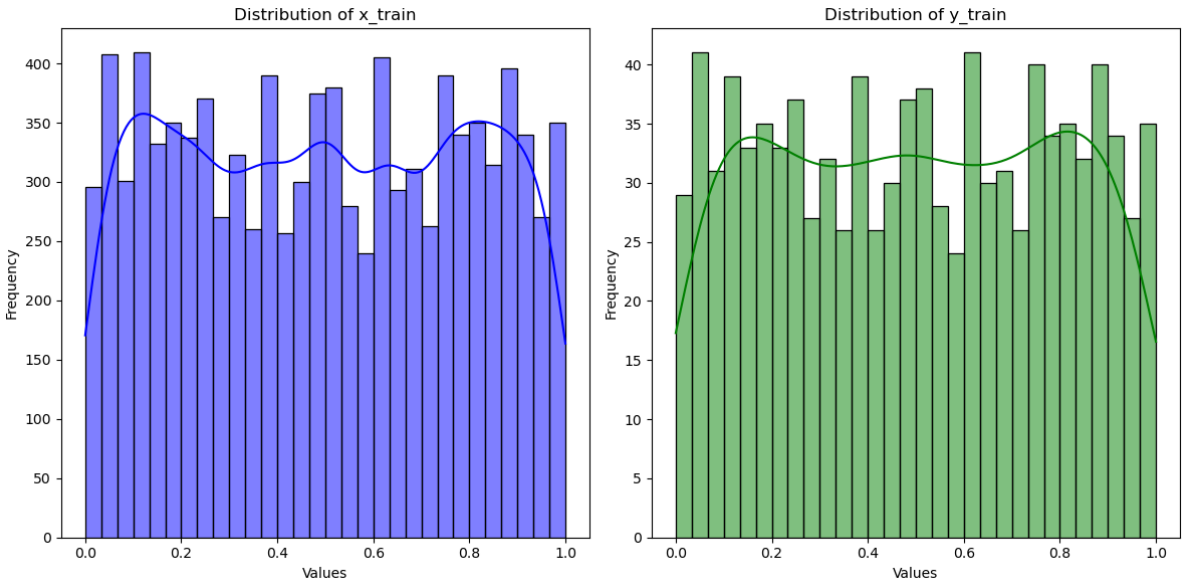



*Рис. 21: Набір даних для тренування від XS*

Після створення набір даних було розділено на три окремі набори, щоб полегшити навчання, тестування та перевірку наших прогнозних моделей:

Навчальний набір: використовується для навчання моделей, дозволяючи їм навчатися та адаптуватися до шаблонів, властивих псевдовипадковим послідовностям, створеним кожним PRNG.

Тестовий набір: використовується для оцінки ефективності моделей на невидимих даних, забезпечуючи неупереджену оцінку їхніх прогнозних можливостей.

Набір перевірки: використовується на етапі налаштування моделі для точного налаштування параметрів і запобігання переобладнанню, гарантуючи, що моделі добре узагальнюються для нових даних.

Ця ретельна підготовка та розділення набору даних були критично важливими для створення надійної основи для нашого дослідження передбачуваності вихідних даних PRNG за допомогою послідовного аналізу. Стандартизувавши параметри генерації та продумано розділивши дані, ми прагнули створити справедливе та узгоджене середовище тестування для кожної моделі прогнозування, застосованої в нашому дослідженні.

У нашому дослідженні «Прогнозування виходу PRNG за допомогою послідовного аналізу» ми застосували комплексний підхід, використовуючи різні архітектури нейронних мереж. Кожну модель було обрано на основі її здатності обробляти послідовні дані, що є основною характеристикою вихідних даних PRNG. Наш аналіз включав згорткові нейронні мережі (CNN), рекурентні нейронні мережі (RNN), мережі довгострокової короткочасної пам’яті (LSTM) і спеціальну гібридну модель, кожна з яких призначена для обробки даних, згенерованих за допомогою PRNG, різними способами.

### 2.4.5 Згорткові нейронні мережі (CNN)

Застосування: використовуються в основному для прогнозування виходу з одним значенням, CNN вміють ідентифікувати шаблони у вікні послідовності фіксованого розміру. Ця модель чудово вловлює локальні залежності та просторову ієрархію в даних, що робить її придатною для аналізу окремих сегментів вихідних даних PRNG.

### 2.4.6 Повторювані нейронні мережі (RNN)

Застосування: RNN використовувалися як для однозначних, так і для безперервних значень вихідних прогнозів. На відміну від CNN, RNN мають механізм пам’яті, який дозволяє їм обробляти цілі послідовності даних, що робить їх ідеальними для розуміння часової динаміки та

залежностей у вихідних даних PRNG.

### 2.4.7 Мережі довготривалої короткочасної пам’яті (LSTM)

Застосування: як і RNN, LSTM використовувалися як для однозначних, так і для безперервних значень вихідних прогнозів. LSTM — це особливий вид RNN, здатний вивчати довготривалі залежності. Вони особливо ефективні в уникненні проблеми зникнення градієнта, дозволяючи їм захоплювати шаблони в більш довгих послідовностях виходів PRNG.

### 2.4.8 Гібридна модель

Конфігурація: гібридна модель представляє інноваційний підхід, об’єднуючи сильні сторони CNN і LSTM в єдину архітектуру. Він містить:

Рівень CNN: для вилучення локальних особливостей у підпослідовності виводу PRNG.

Рівень LSTM: для фіксації довгострокових залежностей і часових шаблонів у даних на основі функцій, отриманих рівнем CNN.

Щільний рівень: слугуючи вихідним рівнем, він синтезує інформацію, оброблену рівнями CNN і LSTM, щоб робити прогнози.

Застосування: розроблена для універсальності, гібридна модель обладнана для обробки як однозначних, так і безперервних виходів, пропонуючи надійне рішення для прогнозування виходів PRNG шляхом використання взаємодоповнюючих переваг згорткових і рекурентних шарів.

Стратегічний вибір і налаштування цих моделей лежать в основі нашої аналітичної методології. Використовуючи різноманітні архітектури, кожна зі своїми унікальними перевагами, наше дослідження має на меті комплексно оцінити передбачуваність вихідних даних PRNG. Гібридна модель підкреслює нашу відданість інноваціям, інтегруючи численні парадигми нейронних мереж для підвищення точності прогнозування та розуміння послідовного характеру даних, створених за допомогою PRNG.

### 2.4.9 Метрики оцінювання

Щоб ретельно оцінити ефективність наших моделей у прогнозуванні результатів PRNG, ми використали набір комплексних оціночних показників. Ці показники мають вирішальне значення для кількісної оцінки точності наших прогнозів і полегшення прямого порівняння між різними архітектурами нейронних мереж, які використовуються в нашому дослідженні. Наша система оцінювання зосереджена навколо середньоквадратичної помилки (MSE) і спеціально розробленого показника ефективності моделі.

### 2.4.10 Середня квадратична помилка (MSE)

MSE є наріжним каменем нашої стратегії оцінювання. Він обчислює середню квадратичну різницю між фактичними та прогнозованими значеннями, пропонуючи точне вимірювання величини помилки передбачення. Зводячи помилки в квадрат, MSE надає більшої ваги більшим помилкам, що робить його особливо чутливим до викидів і значних неточностей прогнозу.

У контексті прогнозування вихідних даних PRNG MSE забезпечує чітке та пряме вимірювання того, наскільки точно прогнози моделі узгоджуються з фактичною послідовністю чисел, згенерованих PRNG. Нижчий MSE вказує на вищу точність передбачення, що відображає здатність моделі ефективно фіксувати та відтворювати базові шаблони послідовності PRNG.

### 2.4.11 Оцінка продуктивності моделі

Визнаючи потребу в стандартизованому показнику, який дає змогу інтуїтивно зрозуміти продуктивність моделі, ми запровадили оцінку ефективності моделі. Цей показник нормалізує MSE до шкали від 0 до 1, де 0 означає найнижчу продуктивність (найвищий MSE), а 1 означає ідеальну точність передбачення (нульовий MSE).

Оцінка продуктивності моделі обчислюється шляхом зворотного масштабування MSE проти попередньо визначеного максимального

порогового значення помилки. Цей підхід гарантує, що оцінка ефективності коригується відповідно до масштабу даних і очікуваної варіації точності прогнозу, що дозволяє чесно порівнювати різні моделі та набори даних.

Ця нормалізована оцінка спрощує інтерпретацію наших результатів, надаючи просту метрику для вимірювання ефективності моделі. Це дозволяє зацікавленим сторонам швидко оцінити відносну продуктивність кожної моделі в прогнозуванні результатів PRNG, не заглиблюючись у складність необроблених значень MSE.

Разом ці метрики оцінювання складають основу нашого аналітичного підходу, що дозволяє детально аналізувати ефективність моделі. MSE пропонує детальне уявлення про точність передбачення, тоді як оцінка ефективності моделі забезпечує високорівневу порівняльну перспективу. Включаючи обидва показники, наше дослідження забезпечує збалансовану та всебічну оцінку того, наскільки добре кожна архітектура нейронної мережі може передбачити, здавалося б, непередбачуване: вихідні дані генераторів псевдовипадкових чисел.

### 2.4.12 Експериментальні змінні та спостереження

Ми провели велику серію експериментів, щоб оцінити передбачувані можливості різних конфігурацій нейронної мережі. Ці експерименти були ретельно розроблені, щоб дослідити вплив різних параметрів моделі на точність вихідних прогнозів PRNG. Нижче ми детально описуємо змінні, задіяні в цих експериментах, і висвітлюємо деякі критичні спостереження, пов’язані з продуктивністю моделі.

### 2.4.13 Експериментальні змінні

Щоб систематично оцінити вплив різних гіперпараметрів на продуктивність моделі, ми перевірили широкий спектр комбінацій, що охоплюють:

Функції активації: ми експериментували з двома популярними функціями активації, ReLU (випрямлена лінійна одиниця) і tanh

(гіперболічний тангенс). Ці функції були обрані через їхні відмінні характеристики в обробці нелінійності в даних.

Кількість нейронів. Випробувана кількість нейронів становила 8, 16 і 32. Цей діапазон дозволив нам дослідити здатність моделей навчатися й узагальнювати дані, врівноважуючи складність із обчислювальною ефективністю.

Епохи: усі моделі були навчені для [1000] епох, надаючи широкі можливості для навчання та конвергенції.

Шари моделі: ми змінювали глибину моделей, тестуючи конфігурації з [38, 39, 41] шарами. Цей варіант мав на меті зрозуміти, як глибина моделі впливає на навчання та точність прогнозування.

Вихідні довжини: для безперервного прогнозування значення було протестовано вихідні довжини [37, 38, 39, 41]. Цей діапазон було обрано для оцінки здатності моделей прогнозувати кілька кроків вперед у послідовності PRNG.

### 2.4.14 Вплив вибуття та регуляризації L2

Одним із найпомітніших висновків наших експериментів був вплив методів відсіву та регулярізації рівня L2 на здатність до вивчення моделі. Всупереч звичайній практиці машинного навчання, де ці методи використовуються для покращення узагальнення моделі та запобігання переобладнанню, наші експерименти показали, що:

Моделі без відсіву та регулярізації L2 продемонстрували чудову продуктивність у навчанні та прогнозуванні вихідних даних PRNG. Запровадження цих методів регуляризації призвело до моделей, які не змогли належним чином навчатися з навчальних даних і, отже, не змогли точно передбачити.

Це спостереження свідчить про унікальний аспект прогнозування вихідних даних PRNG: дані, згенеровані PRNG, хоч і виглядають випадковими, але відповідають детермінованим алгоритмам. Додавання

методів регуляризації, які призначені для введення випадковості та обмежень у процес навчання, може перешкодити здатності моделей охоплювати базові детерміновані моделі послідовностей PRNG.

Результати цих експериментів дають цінну інформацію про дизайн та оптимізацію моделей нейронних мереж для прогнозування вихідних даних PRNG. Зокрема, вони підкреслюють важливість адаптації конфігурацій моделі до конкретних характеристик даних і поставленого завдання. У контексті прогнозування PRNG мінімізація зовнішніх джерел випадковості та обмежень (наприклад, через відсівання та регулярізацію рівня L2) виявляється критично важливою для того, щоб дозволити моделям вивчати та відтворювати детерміновані шаблони, які керують поведінкою PRNG.

### 2.4.15 Аналіз результатів експерименту

Наше вичерпне дослідження прогнозування виходу PRNG за допомогою послідовного аналізу дало переконливі висновки, з'ясовані за допомогою аналізу найефективніших моделей для кожного PRNG. Тут ми детально описуємо важливі результати як для сценаріїв з одним виходом, так і для безперервного виходу в різних PRNG: Xorshift, MT (Mersenne Twister), LCG (Linear Congruential Generator) і MiddleSquare.

### 2.4.16 Аналіз сценарію з одним виходом

Для прогнозів з одним виходом наші експерименти мають такі результати.

Xorshift: модель RNN із 32 нейронами, 5 шарами та функцією активації ReLU стала найефективнішою, отримавши середній бал 0,9898 (таблиця 1). Проте як гібридна, так і CNN-моделі наблизилися до однакового рівня успішності, що свідчить про те, що характеристики послідовності Xorshift не особливо складно зафіксувати.

*Таблиця 1*

Xorshift, один вихідний результат

| Сценарій | Тип моделі | нейрон | Функція активації | Епохи | Шари | Середній бал |
|---|---|---|---|---|---|---|
| Xorshift | RNN | 32 | relu | 1000 | 5 | 0,989848 |
| Xorshift | CNN | 32 | relu | 1000 | 3 | 0,983555 |
| Xorshift | Гібрид | 32 | relu | 1000 | 2 | 0,98293 |
| Xorshift | CNN | 16 | relu | 1000 | 2 | 0,981882 |
| Xorshift | Гібрид | 8 | relu | 1000 | 2 | 0,980837 |
| Xorshift | CNN | 32 | relu | 1000 | 5 | 0,979213 |
| Xorshift | CNN | 8 | relu | 1000 | 2 | 0,978671 |
| Xorshift | CNN | 32 | relu | 1000 | 2 | 0,977584 |
| Xorshift | CNN | 16 | relu | 1000 | 5 | 0,977577 |

50% усіх моделей змогли досягти 90% порогів успішності.

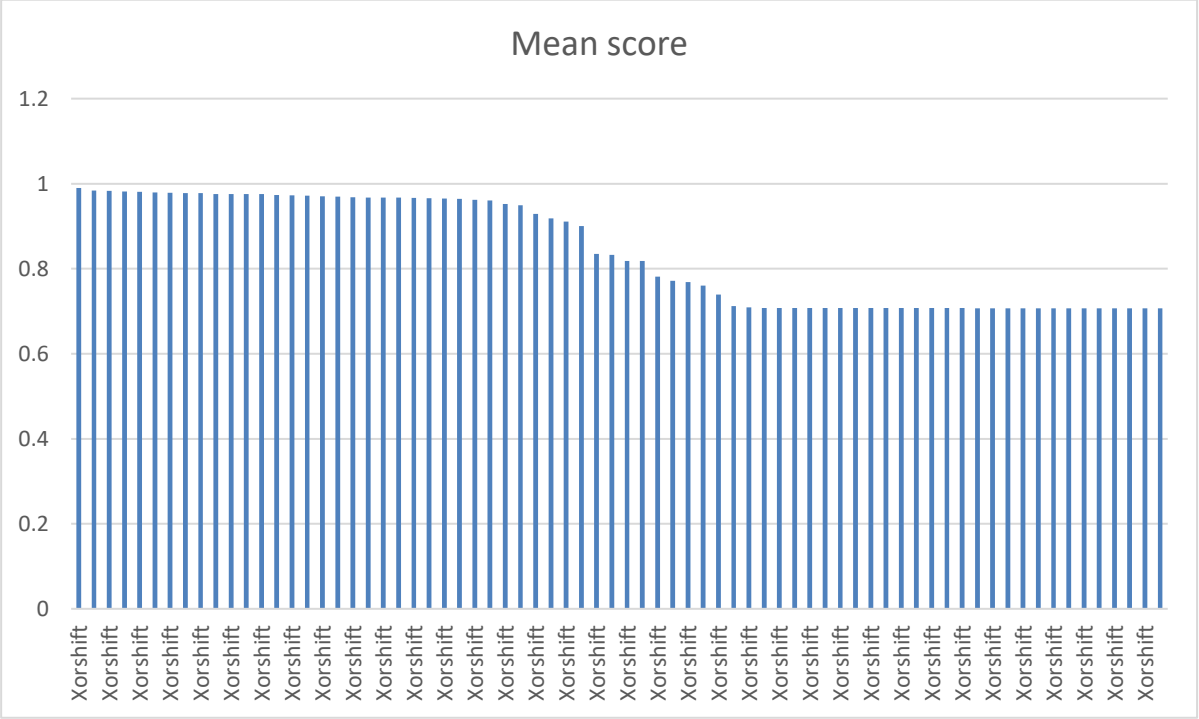



Рис. 22: Результати XS

МТ: модель CNN із 8 нейронами, 3 шарами та функцією активації ReLU лідирувала в групі із середнім балом 0,9832 (таблиця 2), що вказує на те, що можливості вилучення функцій CNN є ефективними для декодування вихідних шаблонів МТ.

*Таблиця 2*

MT, один вихідний результат

| Сценарій | Тип моделі | нейрон | Функція активації | Епохи | Шари | Середній бал |
|---|---|---|---|---|---|---|
| MT | CNN | 8 | relu | 1000 | 3 | 0,983227 |
| MT | CNN | 32 | relu | 1000 | 3 | 0,980932 |
| MT | RNN | 32 | tanh | 1000 | 5 | 0,978619 |
| MT | CNN | 32 | relu | 1000 | 2 | 0,978589 |
| MT | CNN | 16 | relu | 1000 | 2 | 0,977052 |
| MT | LSTM | 32 | relu | 1000 | 5 | 0,976916 |
| MT | RNN | 32 | relu | 1000 | 5 | 0,973991 |
| MT | CNN | 32 | tanh | 1000 | 2 | 0,97303 |
| MT | CNN | 32 | relu | 1000 | 5 | 0,972651 |
| MT | CNN | 16 | relu | 1000 | 5 | 0,972521 |

28% усіх моделей змогли досягти 90% порогів успішності.

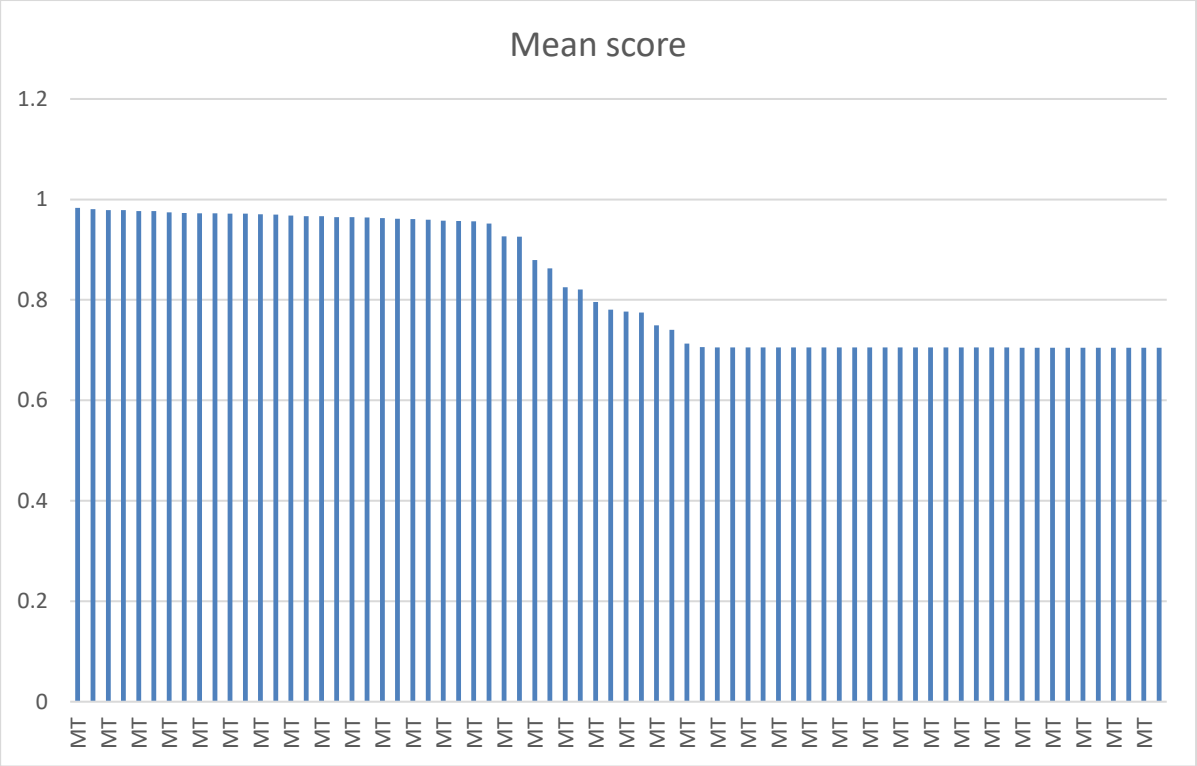



Рис. 23: Результати MT

LCG: гібридна модель, що поєднує архітектури CNN і LSTM з

активацією tanh, продемонструвала чудову продуктивність, особливо з 32 нейронами та 5 шарами, досягнувши середнього балу 0,9831 (таблиця 3). Це підкреслює надійність гібридної моделі в охопленні як локальних, так і далеких залежностей у послідовностях LCG.

*Таблиця 3*

LCG, результати з одним виходом

| Сценарій | Тип моделі | нейрон | Функція активації | Епохи | Шари | Середній бал |
|---|---|---|---|---|---|---|
| LCG | Гібрид | 32 | tanh | 1000 | 5 | 0,983155 |
| LCG | Гібрид | 8 | tanh | 1000 | 2 | 0,982143 |
| LCG | Гібрид | 8 | tanh | 1000 | 3 | 0,980809 |
| LCG | Гібрид | 32 | tanh | 1000 | 2 | 0,980579 |
| LCG | Гібрид | 16 | tanh | 1000 | 2 | 0,979036 |
| LCG | CNN | 8 | relu | 1000 | 2 | 0,978362 |
| LCG | Гібрид | 16 | tanh | 1000 | 3 | 0,978311 |
| LCG | Гібрид | 32 | tanh | 1000 | 3 | 0,97749 |
| LCG | RNN | 32 | tanh | 1000 | 3 | 0,976433 |
| LCG | CNN | 32 | relu | 1000 | 5 | 0,975615 |

60% усіх моделей змогли досягти 90% порогів успішності.

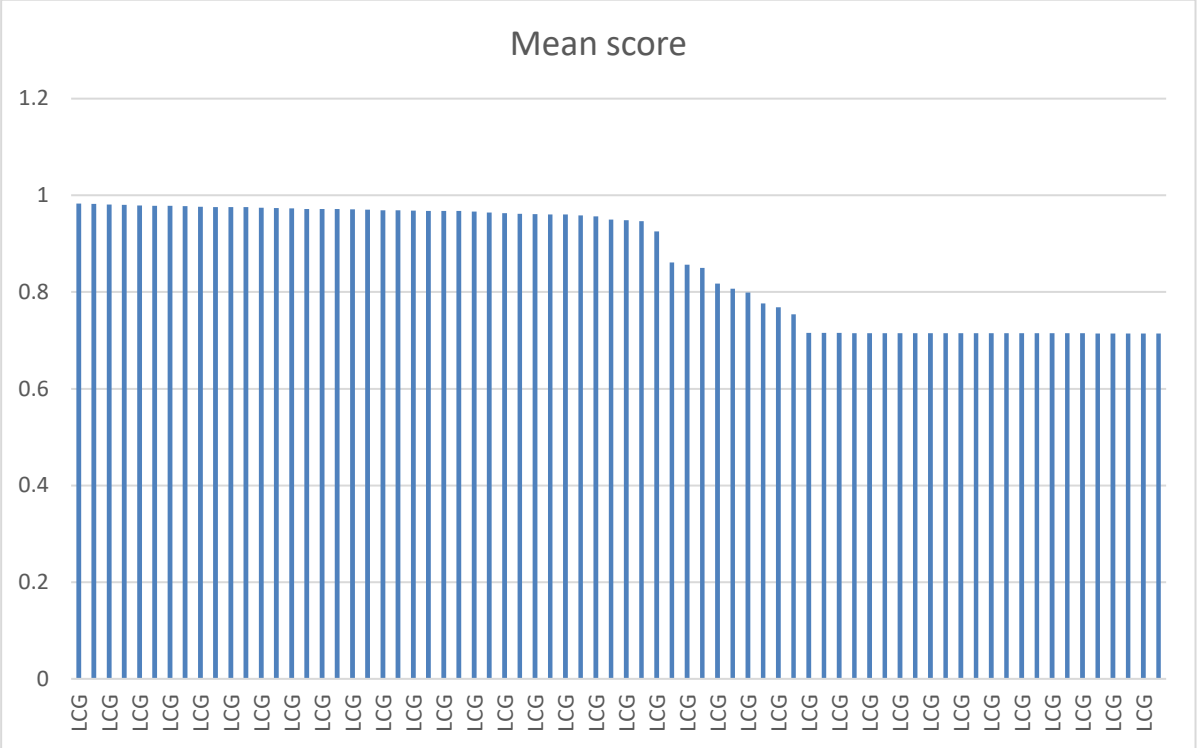



Рис. 24: Результати LCG

MiddleSquare: гібридна модель із активацією tanh, 16 нейронами та 3 шарами виділялася із середнім балом 0,9883 (таблиця 4), що підкреслює ефективність моделі в навігації складними квадратичними обчисленнями, властивими алгоритму MiddleSquare.

*Таблиця 4*

MiddleSquare, один вихідний результат

| Сценарій | Тип моделі | нейрон | Функція активації | Епохи | Шари | Середній бал |
|---|---|---|---|---|---|---|
| MiddleSquare | Гібрид | 16 | tanh | 1000 | 3 | 0,988377 |
| MiddleSquare | CNN | 32 | relu | 1000 | 3 | 0,987493 |
| MiddleSquare | CNN | 32 | relu | 1000 | 2 | 0,986276 |
| MiddleSquare | Гібрид | 32 | tanh | 1000 | 3 | 0,983554 |

| | | | | | | |
|---|---|---|---|---|---|---|
| MiddleSquare | Гібрид | 16 | tanh | 1000 | 5 | 0,983326 |
| MiddleSquare | Гібрид | 8 | tanh | 1000 | 2 | 0,982954 |
| MiddleSquare | CNN | 32 | relu | 1000 | 5 | 0,980597 |
| MiddleSquare | Гібрид | 32 | tanh | 1000 | 5 | 0,98045 |
| MiddleSquare | CNN | 8 | relu | 1000 | 5 | 0,98044 |
| MiddleSquare | Гібрид | 16 | tanh | 1000 | 2 | 0,980428 |

64% усіх моделей змогли досягти 90% порогів успішності.

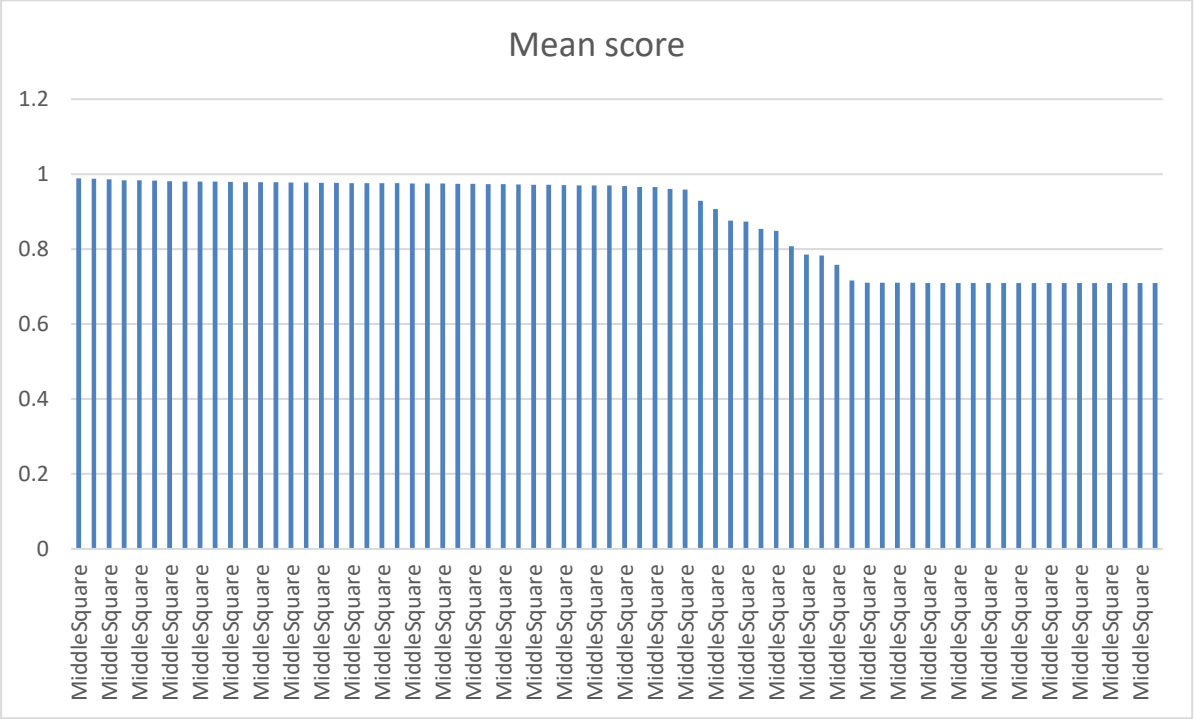



Рис. 25: Результати MS

Ці результати підкреслюють нюанси зв’язку між алгоритмами PRNG та архітектурами нейронних мереж, що свідчить про те, що жодна архітектура однієї моделі не є універсально кращою. Натомість оптимальний вибір залежить від конкретних характеристик і механізмів

прогнозованого PRNG. Незважаючи на те, що моделі були налаштовані для досягнення високої точності прогнозування, графічний аналіз показує, що існує невід'ємне обмеження точності цих прогнозів.

Наступний графік продуктивності ілюструє кореляцію між прогнозованими та фактичними значеннями послідовності PRNG. Майже ідеальне лінійне вирівнювання вздовж лінії 45 градусів свідчить про те, що прогнози моделі сильно корелюють із фактичними виходами PRNG. Щільне групування точок навколо цієї лінії демонструє ефективність моделі в охопленні основного шаблону послідовності PRNG. Однак невелике відхилення точок від лінії означає, що хоча модель може передбачити загальну тенденцію та розподіл вихідних сигналів PRNG, вона не може відтворити послідовність з абсолютною точністю.

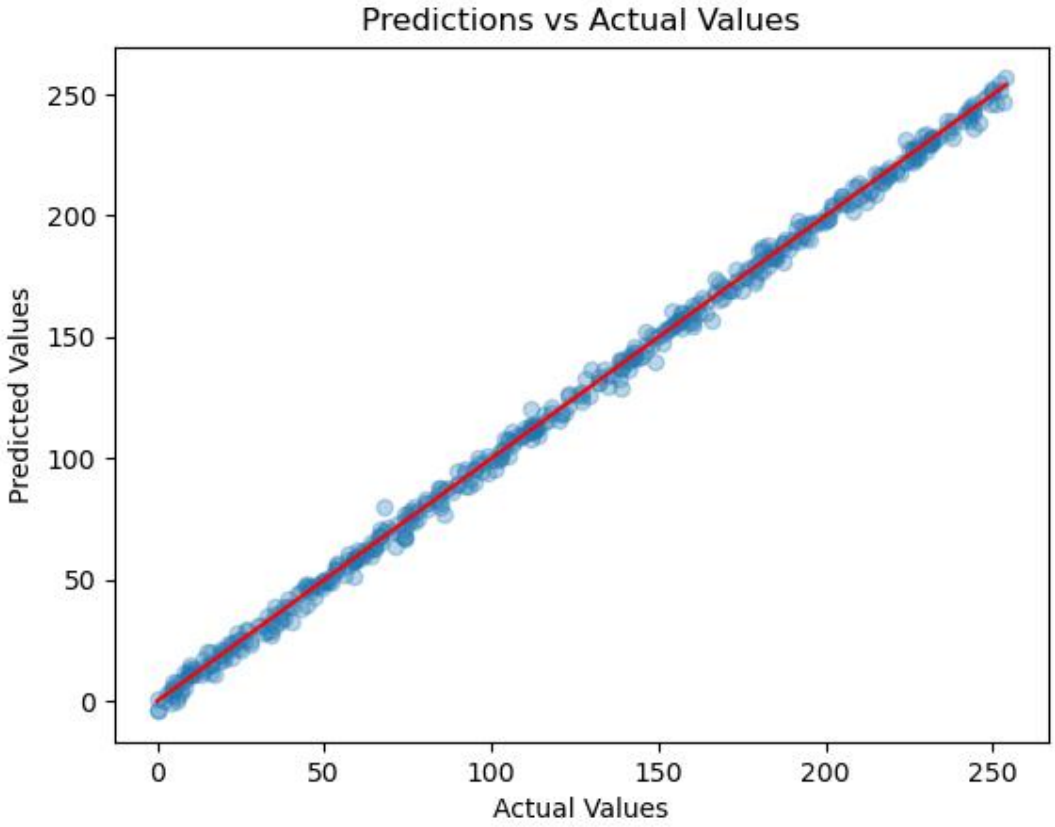



Рис. 26: Результати XS

Діаграма розсіювання, що показує прогнози та фактичні значення для гібридної моделі з використанням активації tanh, 16 нейронів і 3 шарів, показує близьку відповідність між прогнозованими та фактичними

значеннями. Однак дисперсія точок далеко від лінії ідеальної відповідності (де прогнозовані значення дорівнюють фактичним значенням) свідчить про те, що хоча модель може наблизити вихід PRNG з високою точністю, вона не може досягти повної точності. Відхилення від лінії ідеального передбачення можна віднести до детермінованої, але складної природи PRNG, яка за своєю суттю обмежує передбачуваність навіть із складними моделями.

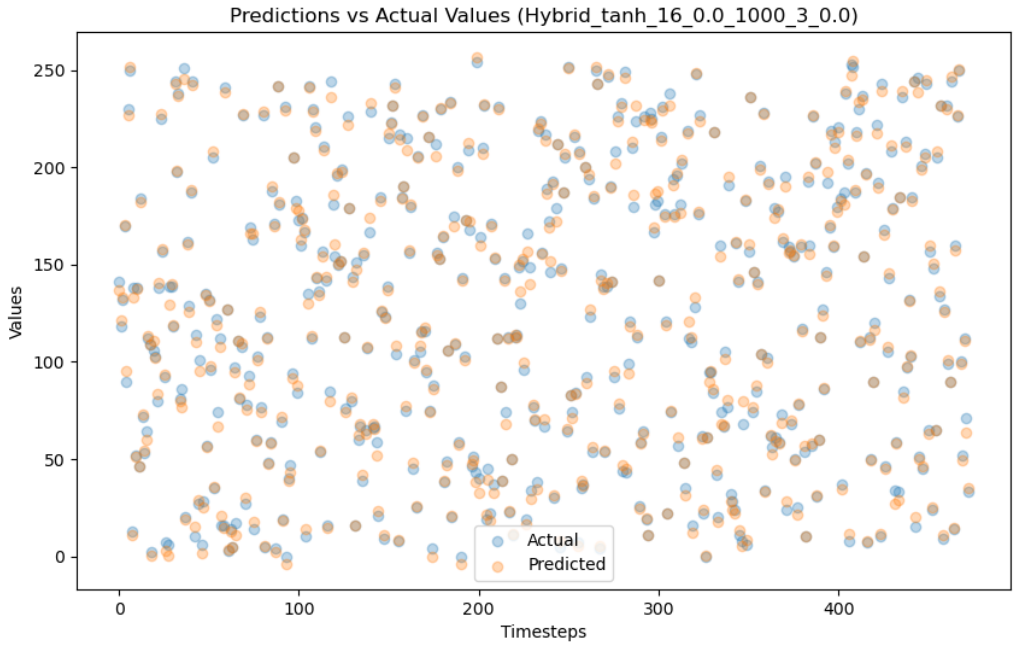



Рис. 27: Результати XS

Моделі безперервного виходу продемонстрували ще вищу точність прогнозування, а гібридна модель, налаштована на безперервне прогнозування (Hybrid-C), досягла надзвичайного успіху.

Для MiddleSquare PRNG модель Hybrid-C з активацією tanh, 16 нейронами, 3 шарами та вихідною довжиною 3 досягла майже ідеальної середньої оцінки 0,9955.

*Таблиця 5*

MiddleSquare, безперервний вихід результатів

| Сценарій | Тип | нейро | Функці | Епохи | Шари | Вихідн | Середн |
|---|---|---|---|---|---|---|---|

| | моделі | н | я активації | | | а довжина | ій бал |
|---|---|---|---|---|---|---|---|
| MiddleSquare | Гібрид-С | 16 | tanh | 1000 | 3 | 3 | 0,995479 |
| MiddleSquare | Гібрид-С | 16 | tanh | 1000 | 2 | 2 | 0,992588 |
| MiddleSquare | Гібрид-С | 8 | tanh | 1000 | 5 | 2 | 0,990003 |
| MiddleSquare | Гібрид-С | 8 | relu | 1000 | 5 | 1 | 0,988989 |
| MiddleSquare | Гібрид-С | 32 | relu | 1000 | 5 | 3 | 0,988566 |
| MiddleSquare | Гібрид-С | 8 | tanh | 1000 | 3 | 1 | 0,988066 |
| MiddleSquare | Гібрид-С | 16 | relu | 1000 | 2 | 2 | 0,986359 |
| MiddleSquare | Гібрид-С | 16 | tanh | 1000 | 5 | 3 | 0,985923 |
| MiddleSquare | Гібрид-С | 16 | tanh | 1000 | 5 | 2 | 0,985797 |
| MiddleSquare | Гібрид-С | 8 | tanh | 1000 | 5 | 1 | 0,984813 |

Лише 29% усіх моделей змогли подолати рубіж успіху в 90%.

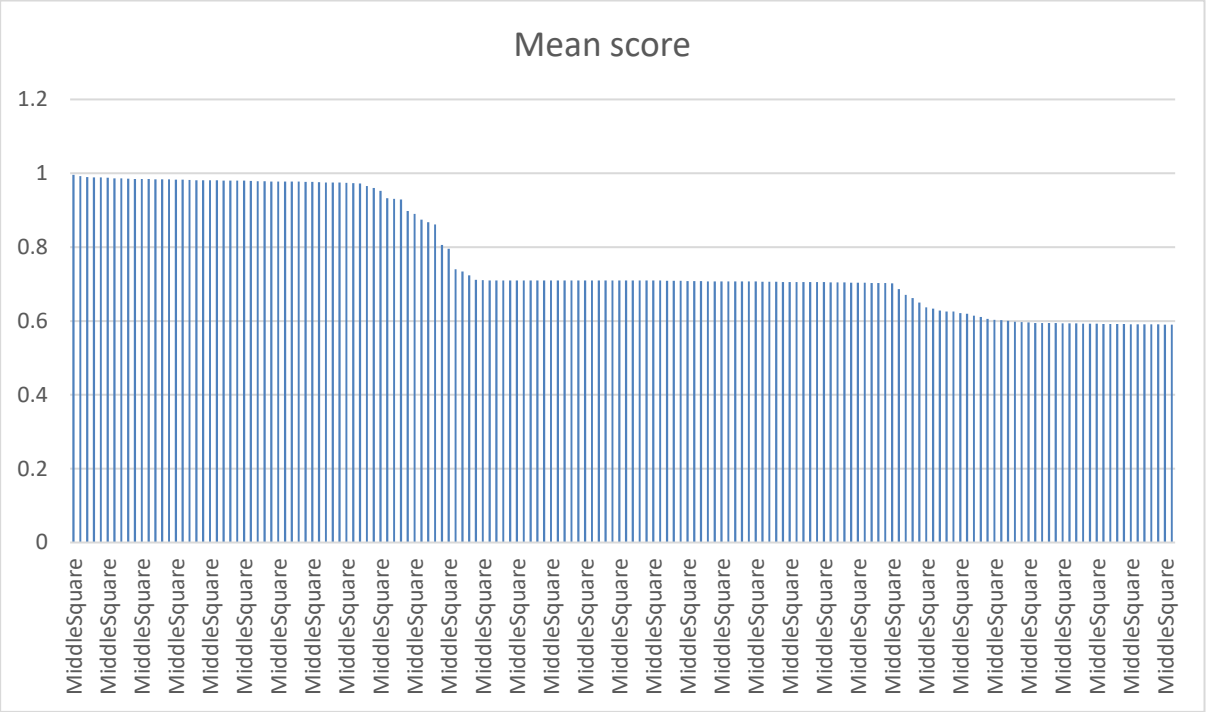



Рис. 28: Результати MS

Для LCG PRNG модель Hybrid-C з активацією tanh, 8 нейронами, 5 шарами та вихідною довжиною 2 досягла майже ідеального середнього балу 0,992055.

*Таблиця 6*

LCG, результати безперервного виведення

| Сценарій | Тип моделі | нейрон | Функція активації | Епохи | Шари | Вихідна довжина | Середній бал |
|---|---|---|---|---|---|---|---|
| LCG | Гібрид-C | 8 | tanh | 1000 | 5 | 2 | 0,992055 |
| LCG | Гібрид-C | 8 | tanh | 1000 | 5 | 3 | 0,98973 |
| LCG | Гібрид-C | 16 | tanh | 1000 | 5 | 5 | 0,987614 |
| LCG | Гібрид-C | 8 | tanh | 1000 | 3 | 5 | 0,986818 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| LCG | Гібрид -C | 8 | tanh | 1000 | 2 | 5 | 0,985174 |
| LCG | Гібрид -C | 16 | tanh | 1000 | 3 | 2 | 0,984808 |
| LCG | Гібрид -C | 32 | tanh | 1000 | 3 | 2 | 0,984462 |
| LCG | Гібрид -C | 16 | tanh | 1000 | 2 | 1 | 0,984411 |
| LCG | Гібрид -C | 32 | tanh | 1000 | 3 | 1 | 0,983866 |
| LCG | Гібрид -C | 32 | tanh | 1000 | 2 | 1 | 0,983706 |

20% усіх моделей змогли подолати 90%-ий поріг успішності.

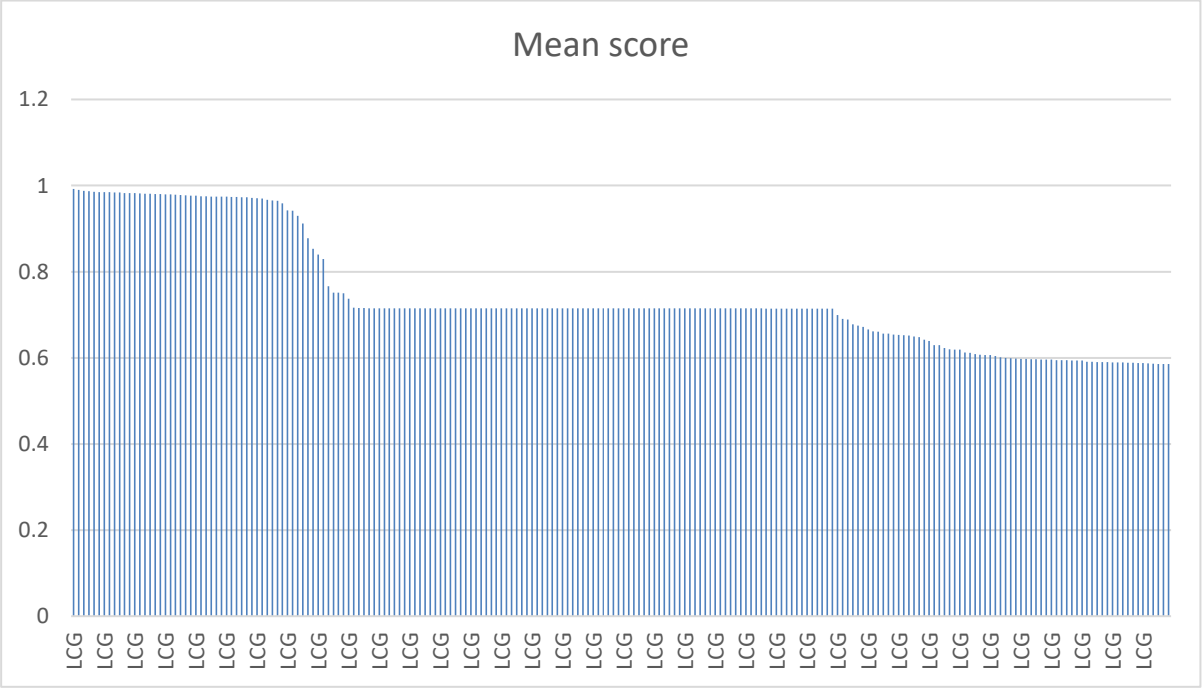



Рис. 29: Результати LCG

Для Xorshift PRNG модель Hybrid-C з активацією relu, 16 нейронами, 2 шарами та вихідною довжиною 2 досягла майже ідеального середнього балу 0,987906 (таблиця 7).

*Таблиця 7*

Xorshift, результати безперервного виведення

| Сценарій | Тип моделі | нейрон | Функція активації | Епохи | Шари | Вихідна довжина | Середній бал |
|---|---|---|---|---|---|---|---|
| Xorshift | Гібрид-С | 16 | relu | 1000 | 2 | 2 | 0,987906 |
| Xorshift | Гібрид-С | 16 | relu | 1000 | 2 | 5 | 0,985753 |
| Xorshift | Гібрид-С | 8 | relu | 1000 | 5 | 5 | 0,985715 |
| Xorshift | Гібрид-С | 8 | relu | 1000 | 2 | 2 | 0,984238 |
| Xorshift | Гібрид-С | 32 | relu | 1000 | 2 | 2 | 0,983437 |
| Xorshift | Гібрид-С | 16 | relu | 1000 | 2 | 3 | 0,981247 |
| Xorshift | Гібрид-С | 8 | relu | 1000 | 2 | 1 | 0,980434 |
| Xorshift | Гібрид-С | 32 | relu | 1000 | 2 | 1 | 0,97893 |
| Xorshift | РНН-С | 32 | tanh | 1000 | 3 | 1 | 0,977685 |
| Xorshift | Гібрид-С | 32 | relu | 1000 | 2 | 3 | 0,976177 |

15% усіх моделей змогли подолати 90% поріг успішності.

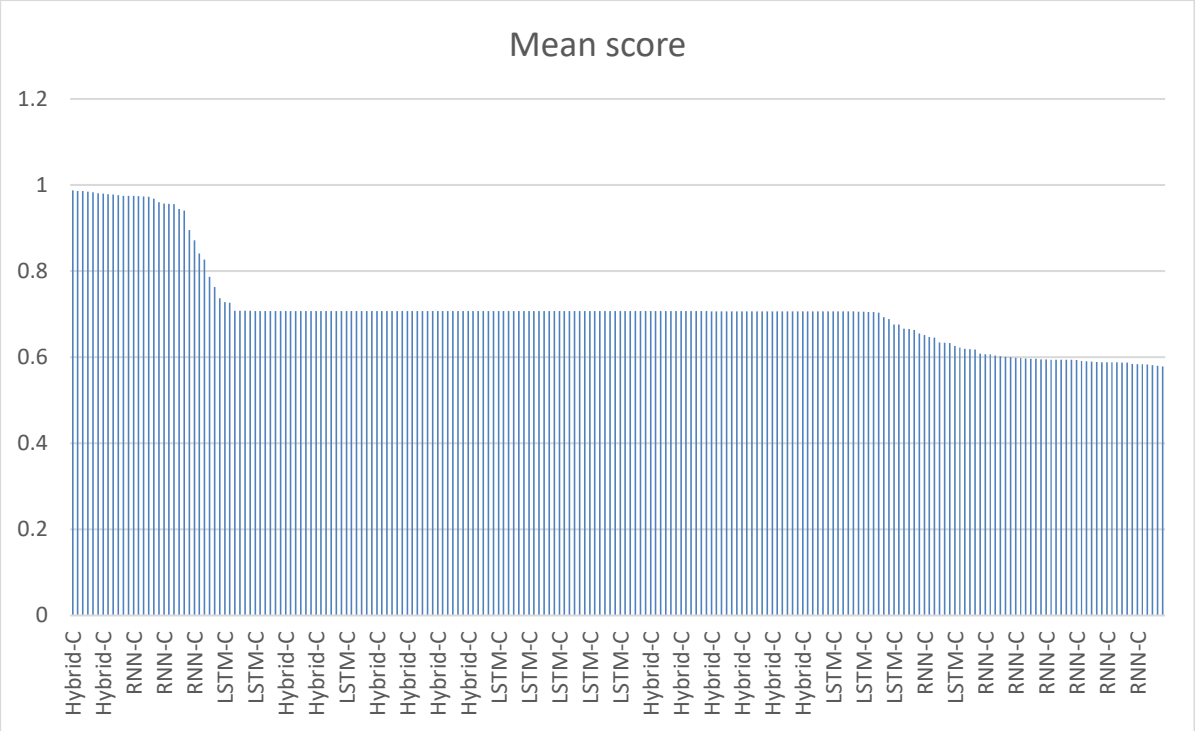



Рис. 30: Результати XS

Для MT PRNG модель Hybrid-C з активацією relu, 32 нейронами, 2 шарами та вихідною довжиною 2 досягла майже ідеального середнього балу 0,985006 (таблиця 8).

*Таблиця 8*

MT, результати безперервного виведення

| Сценарій | Тип моделі | нейрон | Функція активації | Епохи | Шари | Вихідна довжина | Середній бал |
|---|---|---|---|---|---|---|---|
| MT | Гібрид-С | 32 | relu | 1000 | 2 | 2 | 0,985006 |
| MT | РНН-С | 32 | relu | 1000 | 5 | 1 | 0,981523 |
| MT | РНН-С | 32 | tanh | 1000 | 5 | 1 | 0,980135 |
| MT | Гібрид-С | 16 | tanh | 1000 | 2 | 3 | 0,976324 |
| MT | ЛСТМ-С | 32 | relu | 1000 | 5 | 1 | 0,976245 |
| MT | Гібрид- | 8 | tanh | 1000 | 2 | 1 | 0,975258 |

| | C | | | | | | |
|---|---|---|---|---|---|---|---|
| MT | PHH-C | 32 | tanh | 1000 | 3 | 1 | 0,97421 |
| MT | PHH-C | 32 | relu | 1000 | 3 | 1 | 0,972664 |
| MT | PHH-C | 32 | relu | 1000 | 2 | 1 | 0,965829 |
| MT | PHH-C | 32 | tanh | 1000 | 2 | 1 | 0,963914 |

12% всіх моделей змогли подолати 90% поріг успішності.

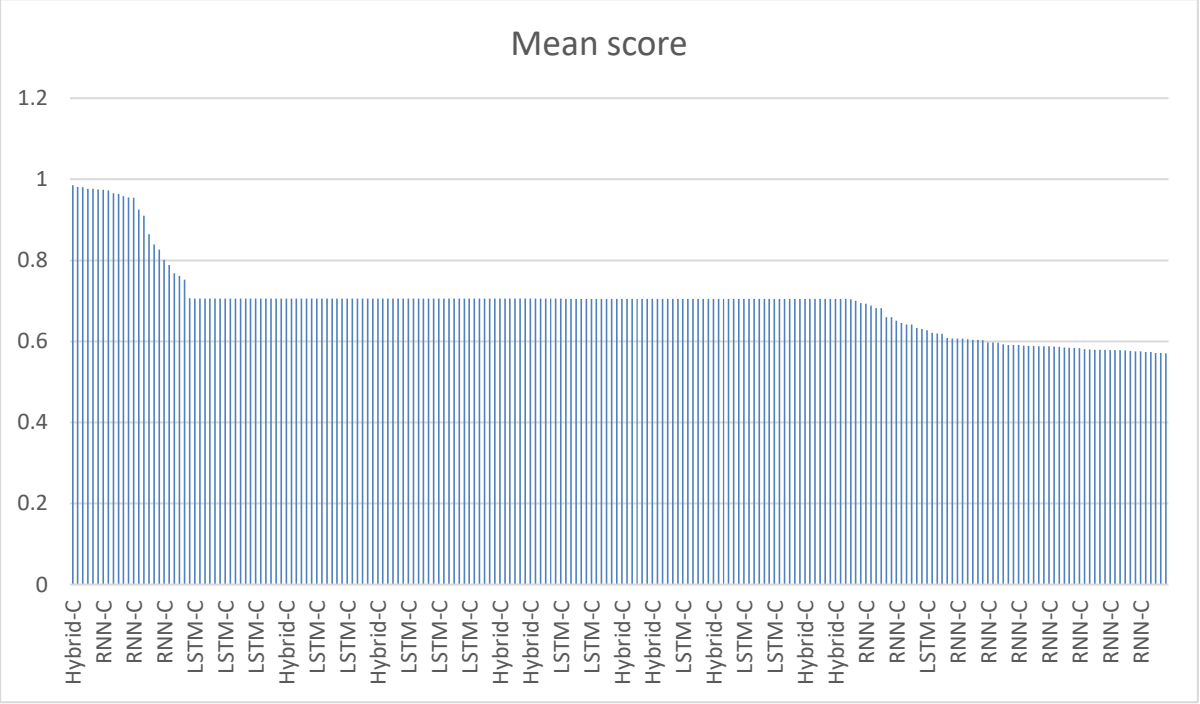



Рис. 31: Результати MT

Дослідження моделей безперервного виходу показує помітне покращення продуктивності прогнозування порівняно з моделями з одним виходом. Це особливо очевидно в контексті передбачення послідовностей, створених MiddleSquare PRNG.

Діаграма продуктивності, яка ілюструє кореляцію між прогнозованими та фактичними значеннями для моделі безперервного виходу, демонструє навіть більш жорстке лінійне вирівнювання, ніж модель з одним виходом. Ця майже ідеальна кореляція разом із високим показником успішності 0,9955 відображає виняткову точність прогнозування моделі. Щільне групування точок уздовж діагоналі свідчить про те, що модель може надійно передбачити вихід MiddleSquare PRNG з високою впевненістю, і така точність свідчить про здатність

моделі охоплювати як безпосередні, так і контекстні залежності всередині послідовності PRNG.

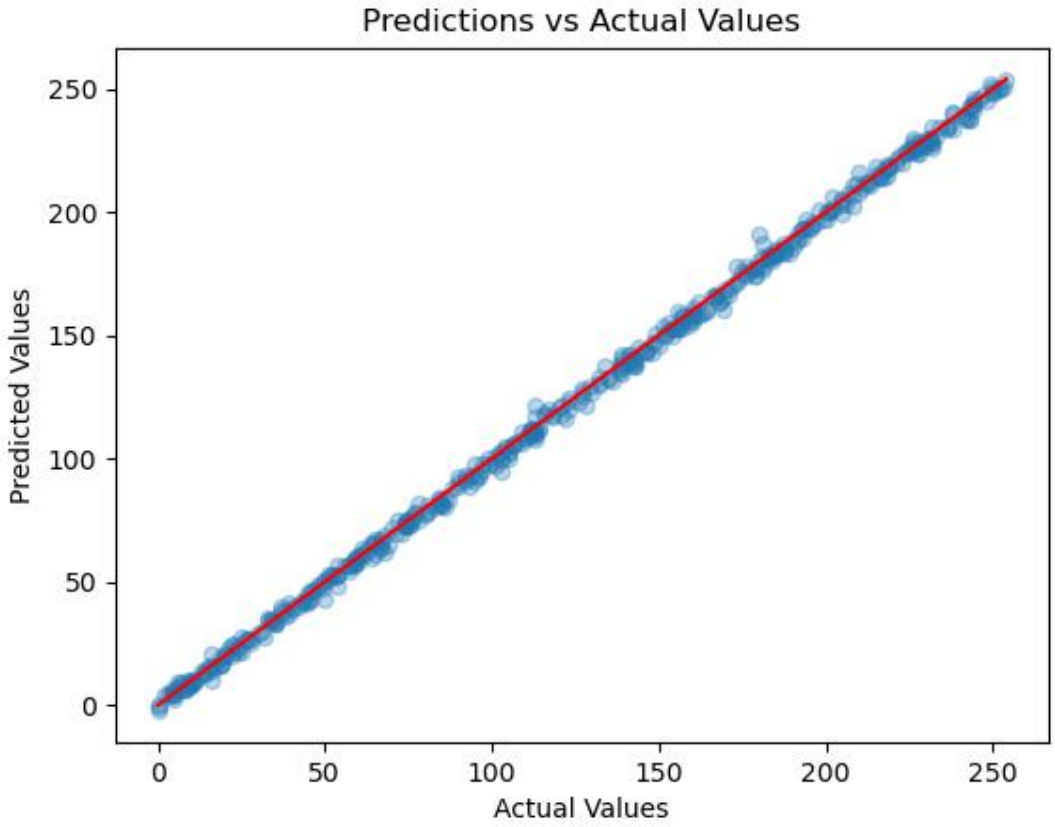



Рис. 32: Результати MS

Діаграма розсіювання для гібридної моделі з безперервним виходом, яка об’єднує архітектури CNN і LSTM (Hybrid-C), демонструє значну концентрацію точок, тісно вирівняних з лінією ідеального прогнозу. Модель, яка використовує активацію tanh із 16 нейронами на 3 шарах, демонструє чудову здатність відстежувати фактичні значення в усій послідовності. Ця тісна кластеризація вказує на суттєве зменшення помилок передбачення та сильне узгодження зі справжньою послідовністю PRNG, що свідчить про глибше розуміння моделлю базових закономірностей.

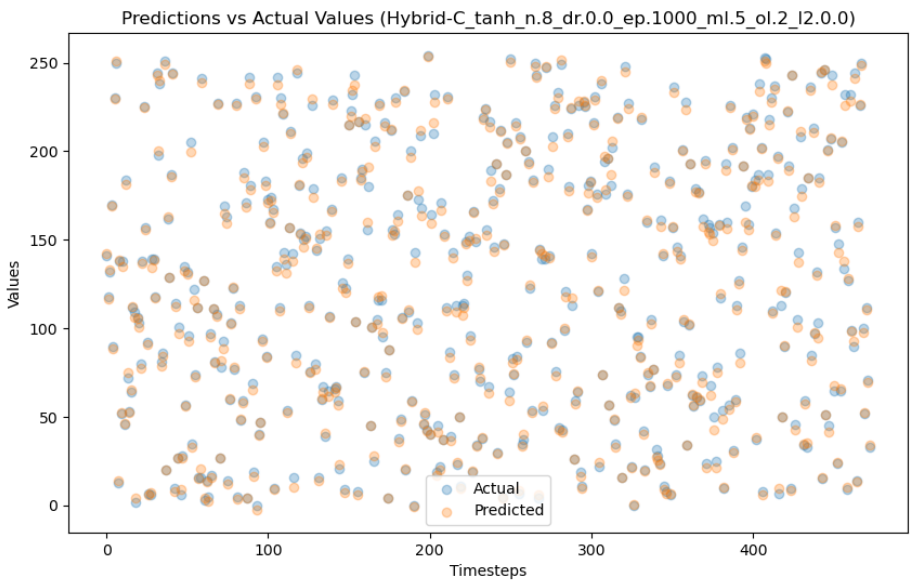



Рис. 33: Результати MS

Висока продуктивність моделі безперервного виводу, про що свідчить ближча близькість прогнозованих до фактичних значень і вищий показник успішності, підкреслює переваги використання послідовного контексту в прогнозуванні виходу PRNG. Здатність прогнозувати послідовність із показником успішності, що досягає 0,9955, є важливою віхою, яка свідчить про те, що моделі, що включають історію послідовності, можуть ефективніше декодувати детерміновану, але складну структуру виходів PRNG.

Цей аналіз означає, що моделі безперервного виходу є перспективними для додатків, де точність прогнозування послідовностей є критичною. Уявлення, отримані в результаті цього дослідження, можуть допомогти розробити більш безпечні PRNG, здатні витримувати складний послідовний аналіз. Майбутня робота, ймовірно, досліджуватиме розширення цього підходу до більш складних і більш вимірних послідовностей, потенційно інтегруючи додаткові рівні складності та вивчаючи вплив на продуктивність моделі.

Висновки нашого дослідження підкреслюють нюанси прогнозування виходу PRNG, коли різні моделі перевершують для конкретних генераторів. Ця варіація підкреслює важливість вибору моделі,

адаптованої до характеристик PRNG, що аналізується. Наприклад, найефективніша модель для генератора Xorshift може використовувати його унікальні операції XOR і зсуву, тоді як оптимальна модель для Mersenne Twister (MT) повинна враховувати його складні методи маніпулювання бітами та відпустки.

Примітно, що моделі з одним виходом послідовно досягали 98% успіху в різних PRNG, демонструючи високий рівень точності в прогнозуванні наступного вихідного значення виключно на основі одного попереднього значення. Цей рівень успіху вказує на здатність моделей розшифровувати базові детерміновані шаблони, які керують виходами PRNG.

Ще більш вражаюче те, що модель безперервного виходу, яка використовує послідовності значень для прогнозування наступних виходів, досягла 99% успіху. Це вдосконалення свідчить про те, що включення більшої кількості контексту у формі безперервних вихідних послідовностей дає змогу моделям краще охоплювати внутрішні алгоритми PRNG, що призводить до більш точних прогнозів.

Успіх наших моделей у прогнозуванні вихідних даних PRNG з такою високою точністю має глибокі наслідки для криптографії та генерації випадкових чисел. Хоча PRNG розроблені для створення послідовностей, які важко передбачити, наші результати свідчать про те, що вдосконалені моделі нейронних мереж можуть розкривати та використовувати приховані шаблони в цих послідовностях. Цей висновок вимагає постійних зусиль для підвищення непередбачуваності та безпеки PRNG, гарантуючи, що вони залишаються надійними проти складних аналітичних методів.

**Висновки**

Це дослідження заглиблюється в передбачуваність генераторів псевдовипадкових чисел (PRNG) з використанням сучасних моделей нейронних мереж. Наше дослідження демонструє, що протестовані

архітектури мають чудову здатність передбачати виходи різних PRNG, з підвищеною точністю, що спостерігається в сценаріях безперервного прогнозування виходу, демонструючи чудову продуктивність у захопленні довготривалих залежностей у послідовностях PRNG, підтверджуючи їхню придатність для складних завдань прогнозування послідовності.

Наші висновки висвітлюють тонку динаміку передбачуваності PRNG і потенційні вразливості, притаманні генераторам, які зазвичай використовуються. Використовуючи нейронні мережі, ми не тільки розкриваємо детерміновані шаблони, замасковані під випадковістю, але й розширюємо межі розуміння криптографічної безпеки та генерації випадкових чисел.

Майбутні дослідження мають вивчити інтеграцію більш складних нейронних архітектур і застосування цих знахідок у сценаріях реального світу, таких як безпечний зв’язок і генерація криптографічних ключів. Наслідки нашої роботи свідчать про кардинальний зсув до більш безпечних і непередбачуваних проектів PRNG, зміцнення захисту від конкурентних прогнозів і підвищення цілісності криптографічних систем.

Майбутні напрямки досліджень

Ці висновки значно покращили наше розуміння можливостей і обмежень сучасних технологій PRNG під впливом передових моделей прогнозування на основі нейронної мережі. Високі показники успішності, досягнуті цими моделями, зокрема 99% успіху з моделями безперервного виведення, не тільки демонструють можливість прогнозування вихідних даних PRNG, але й підкреслюють складні закономірності, які генерують детерміновані алгоритми – закономірності, які можуть розкрити складні моделі.

Це дослідження відкриває кілька шляхів для майбутніх досліджень, спрямованих як на вдосконалення дизайну PRNG, так і на розробку більш просунутих прогнозних моделей:

Удосконалені алгоритми PRNG: існує явна потреба в розробці

нових алгоритмів PRNG, які включають механізми, спеціально розроблені для протидії можливостям моделей прогнозування на основі нейронної мережі. Майбутні дослідження повинні зосередитися на вивченні алгоритмічних складнощів, які можуть більш ефективно приховувати детерміновані шаблони.

Удосконалення нейронних мереж. Наше дослідження показало, що певні архітектури нейронних мереж краще передбачають вихідні дані PRNG, ніж інші. Дослідження розробки нових моделей нейронних мереж або гібридних архітектур, які можуть більш ефективно обробляти та прогнозувати складні послідовності, є захоплюючим рубежем. Це включає дослідження глибших мереж, механізмів уваги та інших розширених функцій, які можуть ще більше підвищити точність передбачення.

Міждисциплінарні підходи: поєднання ідей криптографії, машинного навчання та теорії складності може дати інноваційні підходи як до проектування PRNG, так і до прогнозного моделювання. Міждисциплінарні дослідження можуть відкрити нові принципи створення послідовностей, які за своєю суттю складніше передбачити, а також моделі, які краще розуміють складні закономірності.

Реальні сценарії застосування: застосування наших висновків до реальних сценаріїв, де PRNG використовуються з різними обмеженнями та для різних цілей, буде важливим. Це включає тестування PRNG у середовищах із високими вимогами до безпеки, наприклад у технологіях блокчейн, захищених комунікацій та цифрових підписів.

Етичні міркування та наслідки для безпеки: у міру прогресу досліджень у прогнозуванні вихідних даних PRNG вкрай важливо враховувати етичні наслідки та потенційні ризики для безпеки, пов’язані з розповсюдженням передових прогнозних моделей. Розробка вказівок і найкращих практик для відповідального дослідження та застосування в цій галузі має вирішальне значення.

Підвищення безпеки PRNG: здатність нейронних мереж

передбачати результати PRNG з такою точністю підкреслює нагальну потребу криптографічного співтовариства переоцінити та покращити дизайн і впровадження PRNG. Забезпечення того, що PRNG можуть витримувати аналіз передовими прогнозними моделями, має вирішальне значення для підтримки безпеки та цілісності криптографічних систем, які значною мірою залежать від непередбачуваності цих генераторів.

# РОЗДІЛ 3. МЕТОД ОЦІНЮВАННЯ ЯКОСТІ ПОСЛІДОВНОСТЕЙ ПСЕВДОВИПАДКОВИХ ЧИСЕЛ З ВИКОРИСТАННЯМ АЛГОРИТМІВ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ЗАСТОСУВАНЬ В ГАЛУЗІ КОМП'ЮТЕРНИХ

Послідовності випадкових чисел необхідні в широкому діапазоні застосувань, включаючи криптографію, моделювання та статистичну вибірку. Забезпечення випадковості цих послідовностей має вирішальне значення для підтримки надійності та безпеки цих програм. Традиційні статистичні тести, такі як тест хі-квадрат, широко використовуються для оцінки випадковості. Тим не менш, нещодавні досягнення в машинному навчанні та штучному інтелекті, особливо нейронні мережі, відкривають нові можливості для оцінки випадковості. Це дослідження спрямоване на порівняння ефективності тесту хі-квадрат і методів на основі нейронної мережі для визначення випадковості числових послідовностей.

## 3.1. Визначення критеріїв якості послідовностей псевдовипадкових чисел

Критерій хі-квадрат є широко використовуваним тестом статистичної гіпотези, який визначає, чи існує значна розбіжність між спостережуваними частотами у вибірці та очікуваними частотами за нульовою гіпотезою (тобто дані виявляють випадковість) [65]. У контексті оцінки випадковості критерій хі-квадрат допомагає визначити, чи рівномірний розподіл чисел у послідовності.

Тест хі-квадрат має кілька переваг:

- Простота: тест відносно простий для розуміння та реалізації [65].
- Універсальність: його можна застосовувати до різних типів даних, включаючи категоріальні та числові дані [65].
- Надійність: Тест є непараметричним, тобто він не покладається на припущення щодо базового розподілу даних [65].

Однак тест хі-квадрат має певні недоліки та обмеження:

• Чутливість до розміру вибірки: здатність тесту виявляти невипадковість може бути обмежена в малих вибірках, тоді як він може бути надто чутливим до незначних відхилень від однорідності у великих вибірках [65].

• Залежність від групування: результати тесту можуть коливатися залежно від розподілу даних на контейнери або категорії. Довільне групування може призвести до оманливих висновків [65].

• Нездатність ідентифікувати конкретні шаблони: тест в першу чергу виявляє відхилення від однорідності та може не ідентифікувати інші типи невипадковості, такі як автокореляція, коли значення числа в послідовності залежить від сусідніх значень [65].

• Застосовність до дискретних даних. Критерій хі-квадрат призначений для категоричних або дискретних даних, а при застосуванні до безперервних даних необхідна дискретизація, що потенційно може призвести до втрати інформації [65].

• Припущення незалежності: Тест хі-квадрат передбачає, що спостереження є незалежними. Якщо це припущення порушується, результати тесту можуть бути ненадійними [65].

Незважаючи на ці обмеження, тест хі-квадрат служить цінним інструментом для швидкої оцінки однорідності набору даних або послідовності випадкових чисел.

### 3.2. Вибір та адаптація алгоритмів штучного інтелекту для оцінювання якості

Ми створили дві послідовності: одну абсолютно випадкову, як показано на малюнку 1, і одну з невеликим відхиленням від однорідності (малюнок 2). Потім ми виконали тест хі-квадрат на цих послідовностях з різними розмірами вибірки та вибірками групування:

$$y_t^{(k)} = \sigma\left(\sum_{i=0}^{W-1}\sum_{j=0}^{F-1} w_{ij}^{(k)} x_{t+ij} + b^{(k)}\right)$$

Код обчислює статистику хі-квадрат і значення p для випадкових і зміщених послідовностей із різними розмірами вибірки та різною кількістю категорій. Спостерігаючи за варіаціями в статистиці хі-квадрат і p-значенні для різних розмірів вибірки та кількості категорій, стає очевидним, що ці фактори можуть суттєво впливати на ефективність тесту хі-квадрат.


Рис. 34: Архітектура CNN

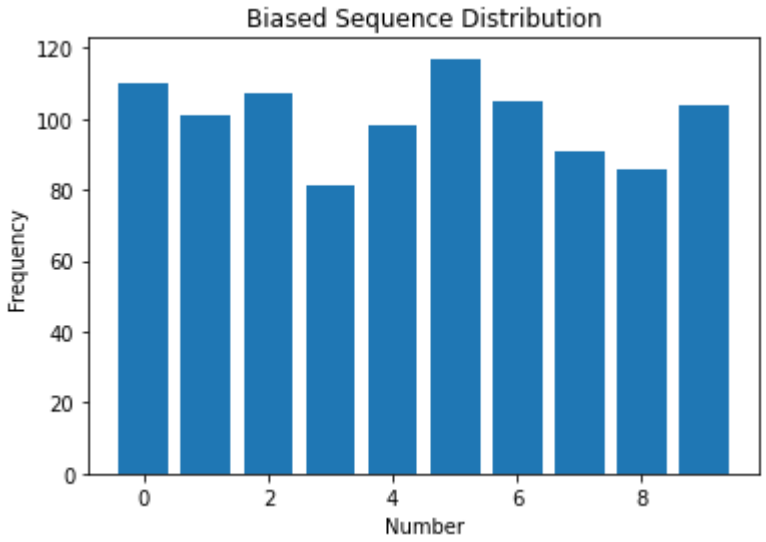



Рис. 35: Розподіл зміщеної послідовності для тесту хі-квадрат

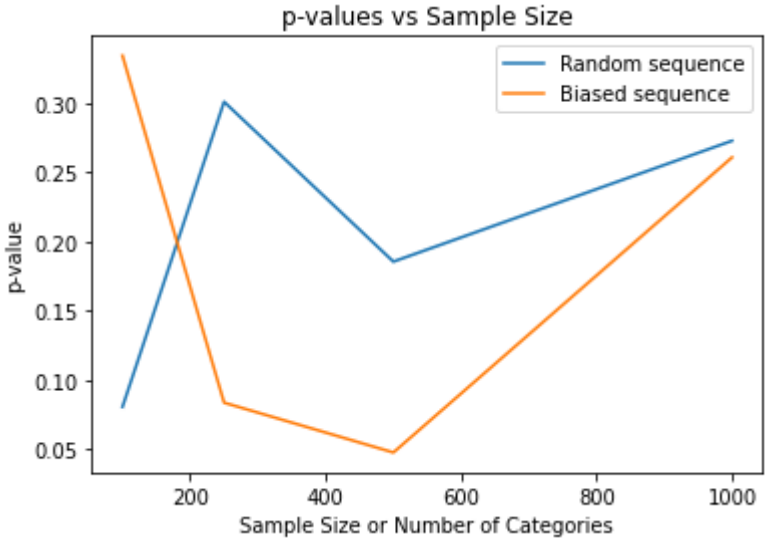



Рис. 36: Чутливість тесту хі-квадрат для випадкових і зміщених наборів даних

Наданий код демонструє чутливість тесту хі-квадрат до розміру вибірки та варіантів групування під час оцінювання випадковості числових послідовностей.

Чутливість до розміру вибірки. Виконуючи тест хі-квадрат на випадкових і зміщених послідовностях із різними розмірами вибірки, код демонструє, як на здатність тесту виявляти невипадковість може вплинути розмір вибірки. Для менших розмірів зразків тест може не виявити відхилення від однорідності, тоді як для більших зразків він може бути надто чутливим до невеликих відхилень.

Чутливість до вибору групування: код також показує, як вибір бінів (кількість категорій) може вплинути на результати тесту хі-квадрат, і показує, що здатність тесту виявляти відхилення від однорідності може змінюватися залежно від того, як дані розподілені на контейнери або категорії (малюнок 3).

### 3.3. Реалізація методу оцінювання

Нейронні мережі продемонстрували потенціал у різних задачах розпізнавання образів і класифікації, включаючи оцінку випадковості числових послідовностей або розрізнення випадкових і невипадкових послідовностей. З цією метою досліджувалися як згорткові нейронні мережі (CNN), так і повторювані нейронні мережі (RNN).

Переваги використання нейронних мереж для оцінки випадковості включають:

Здатність фіксувати складні шаблони: нейронні мережі можуть вивчати складні закономірності та кореляції в даних, які можуть бути пропущені традиційними статистичними тестами [68, 70].

Адаптивність: Нейронні мережі можуть бути точно налаштовані та адаптовані до різних проблемних областей, підвищуючи їх узагальненість [68, 70].

Ємність для роботи з великими наборами даних: Нейронні мережі

можуть ефективно обробляти великі набори даних, дозволяючи їх застосовувати для оцінки випадковості у великих числових послідовностях [68].

Однак є також деякі обмеження, пов’язані з нейронними мережами для оцінки випадковості:

Обчислювально інтенсивні: нейронні мережі, особливо моделі глибокого навчання, можуть бути обчислювально вимогливими, вимагаючи потужного апаратного забезпечення та більш тривалого навчання [68].

Природа чорної скриньки: процес прийняття рішень у нейронних мережах часто важко інтерпретувати, що ускладнює розуміння обґрунтування їх оцінки випадковості [68, 70].

Ризик переналагодження: нейронні мережі можуть бути переналаштовані для навчальних даних, що призводить до поганого узагальнення невидимих даних [68, 70].

Незважаючи на ці проблеми, нейронні мережі пропонують багатообіцяючу альтернативу для оцінки випадковості числових послідовностей, особливо в поєднанні з традиційними статистичними тестами [72, 73]. Поєднуючи сильні сторони обох підходів, дослідники можуть потенційно подолати індивідуальні обмеження кожного методу та забезпечити більш точні та надійні оцінки випадковості.

## 3.4. Експериментальна перевірка ефективності запропонованого методу

Щоб оцінити випадковість у числових послідовностях, ми розглянули можливість використання одновимірної згорткової нейронної мережі (1D-CNN) або рекурентної нейронної мережі (RNN), такої як LSTM або GRU. Хоча обидві архітектури можуть виявляти шаблони в послідовностях, вони мають відмінні переваги.

1D-CNN вміють ідентифікувати локальні шаблони або особливості

у вхідних послідовностях. Вони, як правило, тренуються швидше, ніж RNN, і менш схильні до надмірної фізичної підготовки. Однак їх здатність фіксувати довгострокові залежності обмежена.

RNN, зокрема LSTM і GRU, призначені для обробки послідовностей і можуть фіксувати довгострокові залежності. Вони підтримують прихований стан, який зберігає інформацію з попередніх часових кроків. Однак RNN можуть бути повільнішими для навчання та більш схильними до зникнення або вибуху градієнтів.

Враховуючи нашу мету оцінити випадковість числових послідовностей, 1D-CNN був оптимальним вибором, оскільки він може ефективно виявляти локальні закономірності та потребує менше часу на навчання.

Наданий код створює просту 1D-CNN із двома згортковими шарами, за якими слідує повністю зв’язаний шар і вихідний рівень. Модель скомпільовано з використанням оптимізатора Адама та втрат бінарної кросентропії . Вхідні дані змінюються, щоб відповідати очікуваній формі вхідних даних 1D-CNN, і модель навчається за допомогою даних навчання, тоді як дані перевірки використовуються для моніторингу прогресу навчання та коригування гіперпараметрів, якщо необхідно.

### 3.5. Порівняння з існуючими методами

Наша модель створила малі (50 000 послідовностей) і великі (100 000 послідовностей) набори даних, створила та навчила просту модель 1D-CNN, а також оцінила ефективність тесту хі-квадрат і CNN для обох наборів даних. Кожна послідовність складається з 10 випадково згенерованих цифр і спочатку оцінюється за допомогою тесту хі-квадрат, щоб мати список міток (0 – для випадкових (співвідношення значущості нижче 0,05) і 1 – для невипадкових (співвідношення значущості вище за 0,05)) для навчання та перевірки. Найвища точність і найменші втрати досягнуті через 100 епох (рис. 4).

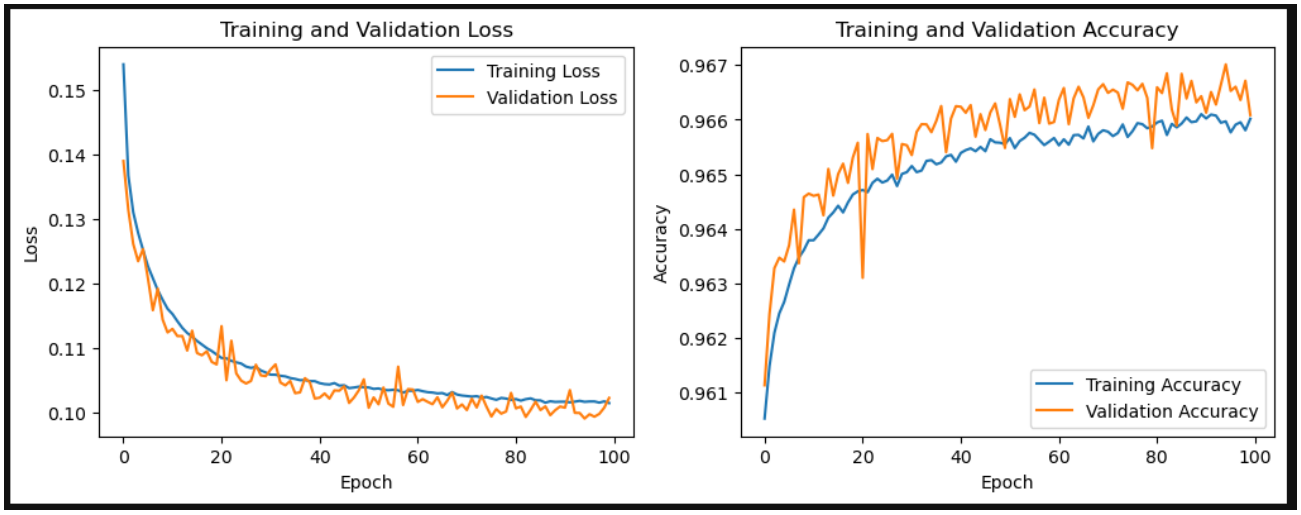



Рис. 37: Втрати та точність навченого режиму 1D-CNN для 10-розрядних послідовностей.

Результати показують, що навчена модель 1D-CNN може оцінити великий набір даних у 7 разів швидше, ніж тест хі-квадрат (малюнок 1) з точністю 96% (малюнок 5).

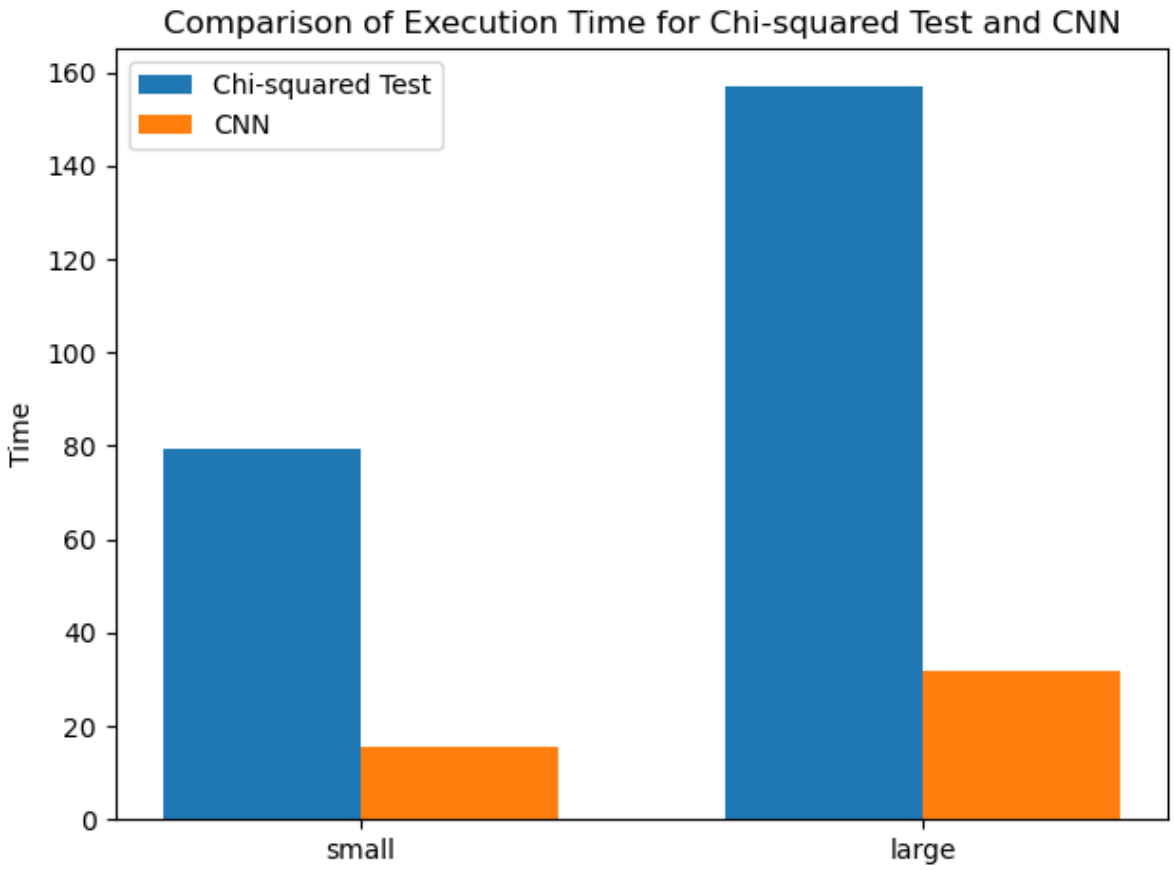



Рис. 38: Перевірка хі-квадрат і час виконання 1D-CNN для доступу до повного набору перевірки 10-значних послідовностей

96% точність показує, що все ще є місце для вдосконалення за рахунок подальшої оптимізації, тестування більшого обсягу даних і додаткового часу на навчання (малюнок 6).

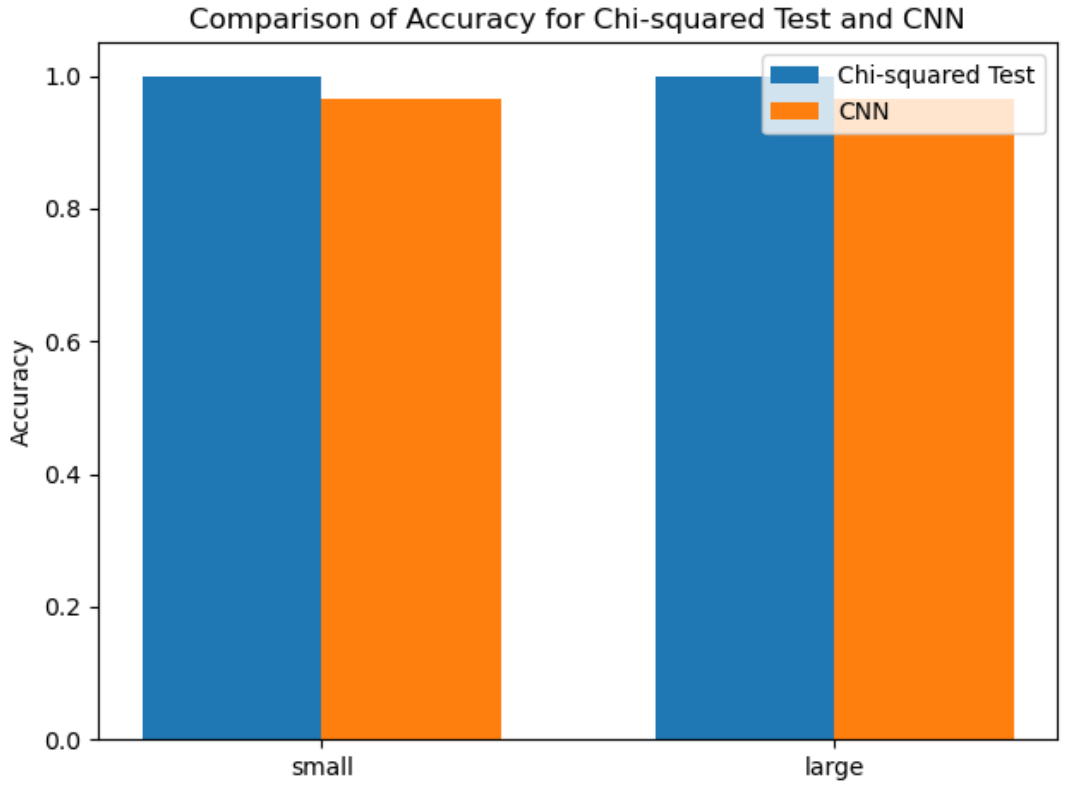



Рис. 39: Критерій хі-квадрат і точність 1D-CNN на основі доступу до повного набору перевірки 10-значних послідовностей

Після збільшення довжини послідовності з 10 до 100 цифр спостерігаються ті ж результати: точність 1D-CNN залишається на 4-5% нижчою, ніж хі-квадрат (малюнок 7), але час виконання на 40% кращий (малюнок 8) . Тим не менш, необхідна додаткова оптимізація для наборів даних з довшими послідовностями, щоб зменшити втрати підтвердження та підвищити точність (малюнок 9).

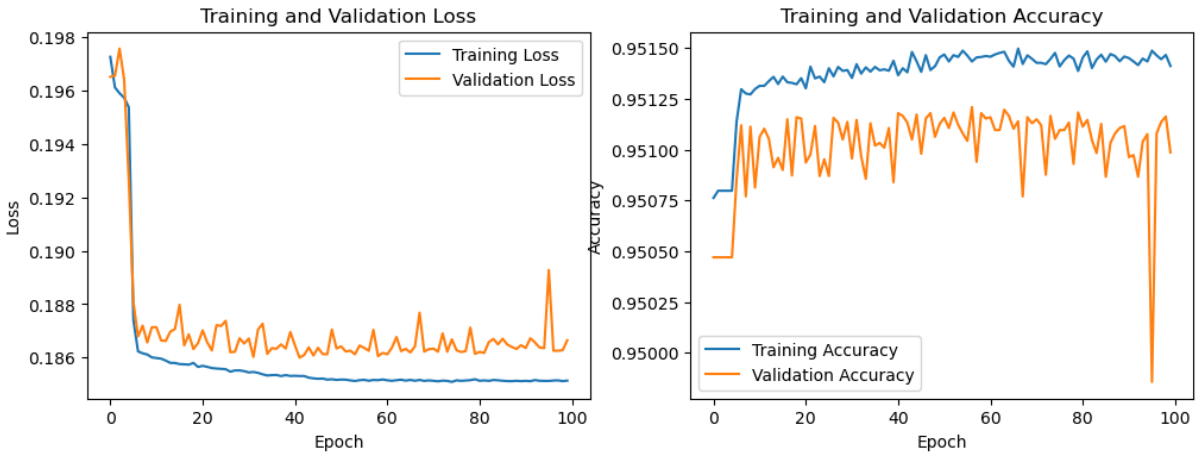



Рис. 40: Втрати та точність навченого режиму 1D-CNN для 100-розрядних послідовностей

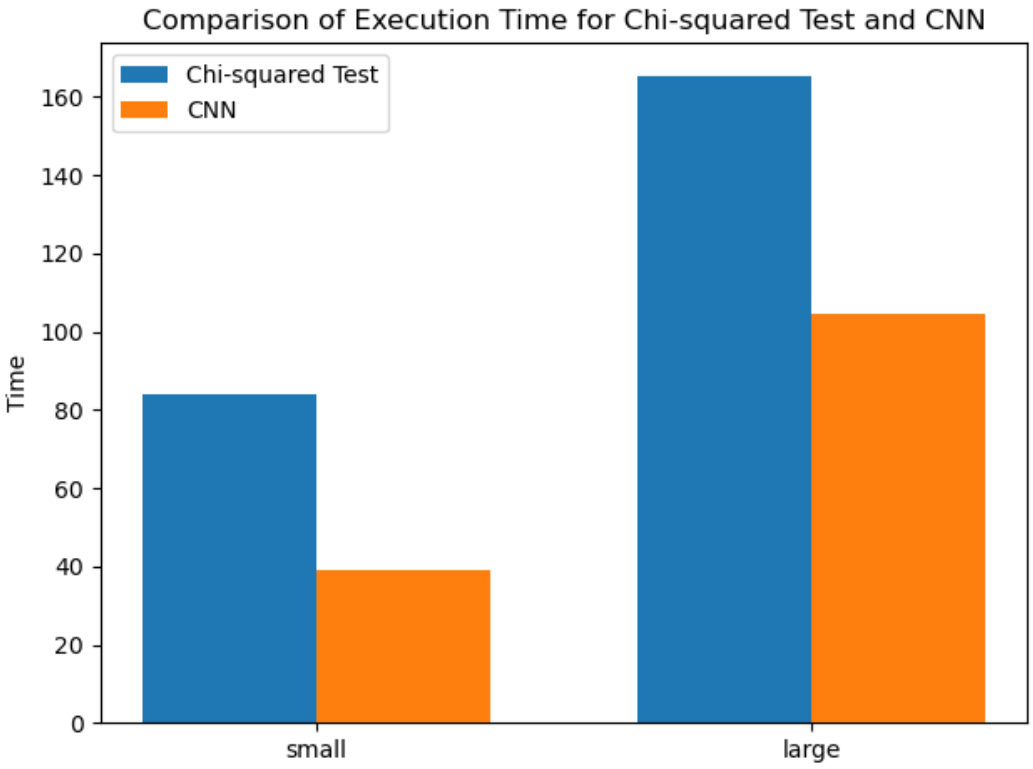



Рис. 41: Тест хі-квадрат і час виконання 1D-CNN для доступу до повного набору перевірки 100-значних послідовностей

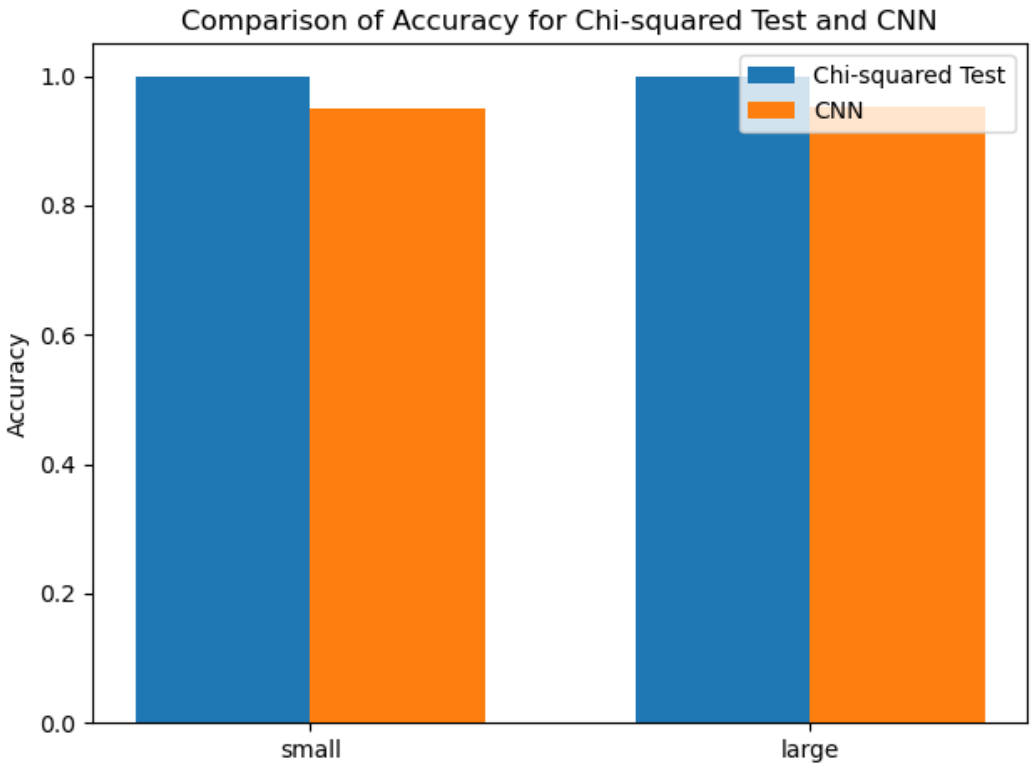



Рис. 42: Тест хі-квадрат і точність 1D-CNN на основі доступу до повного набору перевірки 100-значних послідовностей

На підставі наданого коду та аналізу його результатів можна зробити такі висновки:

Точність: нейронна мережа не демонструє вищої продуктивності з точки зору точності, вона не є більш ефективною для оцінки випадковості послідовності чисел. Необхідні подальші вдосконалення архітектури нейронної мережі та процесу навчання.

Час обчислення: нейронна мережа має вищу продуктивність, особливо під час роботи з великими наборами даних. Процес навчання нейронної мережі може займати багато часу, але після навчання вона потенційно може пропонувати швидші прогнози.

Хоча поточне дослідження дає цінну інформацію про ефективність тесту хі-квадрат і підходів на основі нейронних мереж для оцінки випадковості числових послідовностей, воно має деякі обмеження, які вимагають подальших досліджень:

Обмежене дослідження архітектур нейронних мереж. Дослідження було зосереджено насамперед на 1D-CNN, і подальші дослідження мають вивчити продуктивність інших архітектур, таких як RNN, LSTM та більш просунуті варіанти CNN.

Гіперпараметрична оптимізація: у поточному дослідженні використовувалася відносно проста архітектура нейронної мережі та процес навчання. Майбутні дослідження повинні досліджувати вплив оптимізації гіперпараметрів, таких як швидкість навчання, розмір партії та кількість епох, на продуктивність нейронної мережі.

Неконтрольоване та напівконтрольоване навчання: дослідження покладалося на позначені дані для навчання та оцінювання. Майбутні дослідження можуть вивчити потенціал неконтрольованих або напівконтрольованих методів навчання для оцінки випадковості, не покладаючись на позначені дані.

Дані та додатки реального світу: у дослідженні для оцінки використовувалися синтетичні набори даних, які можуть не повністю відобразити складність даних реального світу. Майбутні дослідження мають оцінити ефективність цих методів з використанням реальних даних і програм, таких як генерація криптографічних ключів або моделювання за методом Монте-Карло.

Комбінація методів: поточне дослідження було зосереджено на незалежному порівнянні тесту хі-квадрат і підходів на основі нейронної мережі. Майбутні дослідження можуть вивчити потенційні переваги поєднання цих методів або використання методів ансамблю для покращення загальної ефективності оцінки випадковості.

Враховуючи попередній характер досліджень нейронних мереж для оцінки випадковості, існує значний потенціал для подальших досліджень і розвитку в цій галузі.

**Висновки**

У цьому дослідженні ми вирішили дослідити ефективність тесту хі-квадрат і методів на основі нейронної мережі в оцінці випадковості числових послідовностей. Наші початкові висновки свідчать про те, що методи на основі нейронних мереж є перспективними в цій області. Однак важливо зазначити, що ці результати можуть бути не остаточними через обмежений обсяг експериментів, проведених у дослідженні. Тим не менш, спостереження дають цінну інформацію про потенційні переваги та недоліки використання нейронної мережі для оцінки випадковості. Наприклад, нейронні мережі можуть запропонувати покращену масштабованість і надійність у порівнянні з тестом хі-квадрат, якщо працювати з великими та різноманітними наборами даних.

У рамках майбутніх досліджень ми рекомендуємо проводити більш масштабні експерименти з більшими наборами даних і різними послідовностями чисел для подальшої перевірки та вдосконалення результатів цього дослідження. Крім того, було б корисно вивчити більш просунуті архітектури нейронних мереж і методи машинного навчання, щоб визначити найбільш ефективний підхід для оцінки випадковості в числових послідовностях.

Крім того, розгляд можливості інтеграції нашого підходу з усталеними наборами тестів на випадковість, такими як тест NIST, може бути цінним. Поєднання сильних сторін статистичних тестів і методів машинного навчання може призвести до більш комплексної та надійної оцінки випадковості, дозволяючи покращити виявлення невипадкових закономірностей і краще зрозуміти їх основні структури.

На завершення це дослідження забезпечує основу для майбутніх досліджень у галузі оцінки випадковості з використанням методів машинного навчання. Ми вважаємо, що наукове співтовариство може ще більше підвищити ефективність і надійність оцінки випадковості числових послідовностей, розширивши попередні висновки та включивши досконаліші методи та набори тестів.

# РОЗДІЛ 4: ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ДЛЯ КОМПЛЕКСНОГО ОЦІНЮВАННЯ ЯКОСТІ ГЕНЕРАТОРІВ ПОСЛІДОВНОСТЕЙ ПСЕВДОВИПАДКОВИХ ЧИСЕЛ В УМОВАХ ОБМЕЖЕНОЇ КІЛЬКОСТІ ВХІДНИХ ДАНИХ

## 4.1. Архітектура інформаційної технології

Інформаційна технологія для оцінювання якості генераторів послідовностей псевдовипадкових чисел (ГППЧ) включає кілька ключових компонентів, які забезпечують комплексний підхід до аналізу. Архітектура системи складається з наступних основних модулів:


Рис. 43: Архітектура

Модуль отримання та форматування даних: відповідає за первинне прийняття послідовностей, їх попередню обробку та підготовку до аналізу.

Модуль ідентифікації джерела послідовності: використовує натреновані на існуючих даних нейронні мережі для ідентифікації джерела послідовності.

Модуль передбачення послідовності: застосовує нетреновані нейронні мережі для передбачення подальших чисел у послідовності та оцінки їхньої якості.

Модуль аналізу послідовності: проводить детальний аналіз послідовності з використанням згорткових нейронних мереж та статистичних методів.

Основний процес роботи системи починається з отримання

послідовності чисел, яка потім проходить через кілька етапів обробки та аналізу. Кожен модуль системи відіграє критичну роль у забезпеченні точності та надійності оцінювання якості ГППЧ. Важливим аспектом є інтеграція всіх компонентів в єдину інформаційну технологію, яка здатна адаптуватися до різних умов та типів даних.

## 4.2. Компоненти системи та їх функції

### 4.2.1. Модуль отримання та форматування даних

Цей модуль відповідає за прийом вхідних послідовностей чисел з різних джерел. Після отримання послідовності, дані форматуються для відповідності вимогам першого шару генеративної нейронної мережі (ГНМ). Цей процес включає нормалізацію даних та підготовку їх до подальшого аналізу. Форматування даних включає перетворення вхідних значень у стандартизований формат, що дозволяє забезпечити коректне функціонування всіх наступних модулів системи.

### 4.2.2. Модуль ідентифікації джерела послідовності

На цьому етапі використовується натренована ГНМ, яка була навчена на існуючих даних з відомих джерел. Цей модуль відповідає за порівняння вхідної послідовності з базою відомих зразків і визначення ймовірності того, що послідовність походить від певного генератора. Якщо точність ідентифікації перевищує 85%, генератор вважається ідентифікованим, а якість визначається як низька. Якщо ж точність нижче цього порогу, послідовність передається до наступного етапу. Модуль ідентифікації також зберігає результати аналізу для подальшого використання у інших модулях.

### 4.2.3. Модуль передбачення послідовності

Цей модуль використовує нетреновану ГНМ для передбачення наступних значень в послідовності. Отримані дані використовуються для тренування, тестування та валідації моделі (80%, 10%, 10% відповідно). Якщо точність передбачення перевищує 85% та значення R-квадрату відповідає вимогам, генератор вважається ідентифікованим, а якість

визначається як низька. Інакше послідовність передається до наступного етапу. Модуль передбачення також здійснює постійний моніторинг та оновлення моделей для підвищення їхньої точності та надійності в умовах зміни вхідних даних.

### 4.2.4. Модуль аналізу послідовності

На цьому етапі використовується натренована згорткова нейронна мережа (ЗНМ) для аналізу послідовності. Аналіз здійснюється шляхом обчислення показника chi-квадрат, який визначає випадковість послідовності. Якщо показник ближче до 1, послідовність вважається невипадковою; якщо ближче до 0 – випадковою. Додатково модуль аналізу може використовувати інші статистичні методи та метрики для більш точного оцінювання якості послідовностей, такі як автокореляція, спектральний аналіз та інші.

## 4.3. Інтеграція моделей оцінювання якості в інформаційну систему

Інтеграція моделей ГНМ та ЗНМ в інформаційну систему передбачає налаштування середовища для тренування моделей, збереження та використання натренованих моделей для аналізу вхідних даних. Цей процес включає встановлення відповідного програмного забезпечення, налаштування апаратних ресурсів, таких як графічні процесори (GPU), та розробку інтерфейсів для взаємодії користувачів із системою. Важливим аспектом є забезпечення зручного інтерфейсу для користувачів, який дозволяє легко завантажувати послідовності, отримувати результати аналізу та рекомендації щодо якості генераторів. Інтерфейс також має включати можливості для візуалізації результатів аналізу, що дозволяє користувачам краще розуміти процес оцінювання та приймати обґрунтовані рішення.

## 4.4. Використання технології в різних застосуваннях

Запропонована інформаційна технологія може бути використана в різних галузях, де важливо забезпечити високу якість генерації

випадкових чисел. Основні галузі застосування включають:

Криптографія: забезпечення надійності шифрування та безпеки даних через генерацію високоякісних випадкових чисел, які неможливо передбачити або відтворити.

Наукові дослідження: використання випадкових чисел для симуляцій та моделювання складних систем, де точність випадковості є критичною для отримання достовірних результатів.

Ігрова індустрія: генерація випадкових подій, результатів та сценаріїв, що підвищує реалістичність та непередбачуваність ігрового процесу.

Фінансові системи: випадковий розподіл ризиків, генерування сценаріїв для моделювання фінансових ринків та аналізу ризиків, що дозволяє краще прогнозувати можливі результати та приймати ефективні рішення.

Технологія також може бути адаптована для інших специфічних застосувань, де потрібна висока якість випадкових чисел, таких як медичні дослідження, телекомунікації, та інші.

### 4.5. Оцінка продуктивності та надійності системи

Оцінка продуктивності системи здійснюється через метрики точності та швидкості аналізу. Важливими показниками є також час обробки послідовностей та ресурсомісткість моделей. Для забезпечення надійності системи проводяться регулярні тести на різних наборах даних для перевірки стійкості до різних типів вхідних послідовностей. Важливим аспектом є також моніторинг та оптимізація використання апаратних ресурсів, щоб забезпечити ефективну роботу системи навіть при великому обсязі даних.

Для оцінки надійності використовуються показники стабільності роботи моделей в умовах змін вхідних даних, а також здатність системи вчасно реагувати на зміни та адаптуватися до нових умов. Важливу роль

відіграє резервне копіювання моделей та даних, що дозволяє відновлювати систему у випадку збоїв або втрати даних.

**Висновки**

Запропонована інформаційна технологія для оцінювання якості генераторів послідовностей псевдовипадкових чисел забезпечує комплексний підхід до аналізу. Вона дозволяє виявляти неякісні, ненадійні та скомпрометовані генератори в умовах обмеженої кількості вхідних даних. Використання технологій машинного навчання, таких як ГНМ та ЗНМ, значно підвищує ефективність і точність оцінювання, роблячи цю систему корисною для широкого спектру застосувань.

Важливою перевагою цієї технології є її здатність адаптуватися до різних умов та вимог, що робить її універсальним інструментом для оцінювання якості генераторів послідовностей. Розробка та впровадження цієї технології сприяє підвищенню безпеки та надійності у багатьох сферах діяльності, де використовуються випадкові числа. Система також може слугувати основою для подальших досліджень та розробок у галузі оцінювання якості випадкових чисел та інших застосувань машинного навчання.

## ВИСНОВКИ

У дисертаційній роботі проведені наукові дослідження, спрямовані на забезпечення швидкого та точного оцінювання генераторів послідовностей псевдовипадкових чисел в умовах обмеженої кількості вхідних даних. Зокрема, отримано такі вагомі наукові та практичні результати:

Проведено аналіз існуючих підходів до генерування та оцінювання якості послідовностей псевдовипадкових чисел. Виявлено, що на сьогодні ефективним і перспективним є використання методів і засобів штучного інтелекту (зокрема, згорткових, рекурентних і гібридних нейронних мереж) в контексті оцінювання якості генераторів послідовностей псевдовипадкових чисел в умовах обмеженої кількості вхідних даних;

Розроблено і досліджено експериментально модель ідентифікації джерела послідовностей псевдовипадкових чисел, що дає можливість виявляти генератори, якими були сформовані послідовностей псевдовипадкових чисел. Використання гібридної нейронної мережі (HNN) дало можливість дослідити 8 реалізованих генераторів та 4 000 згенерованих послідовностей (для навчання та оптимізації) і ідентифікувати на основі нових для HNN даних з усередненою точністю 87,14% джерела послідовностей псевдовипадкових чисел;

Удосконалено та досліджено модель передбачення наступної послідовності псевдовипадкових чисел, яка дозволяє передбачати чергові послідовності для неякісних генераторів. Використання гібридної нейронної мережі (HNN) дало можливість більш точно передбачати наступні послідовності у порівнянні з рекурентною нейронною мережею (RNN) та згортковою нейронною мережею (CNN) – на 2,52% та 1,44% відповідно;

Розвинуто та досліджено експериментально метод оцінювання якості послідовностей псевдовипадкових чисел, який дозволяє більш швидко оцінювати якість генераторів для криптографічних та інших

застосувань в галузі комп’ютерних наук. Реалізація одновимірної згорткової нейронної мережі (1D-CNN) та генераторів псевдовипадкових чисел середовища розробки python підтвердила можливість на 40% швидшого оцінювання якості генераторів у порівнянні з методом chi-квадрат (з незмінним рівнем точності P-value), а також можливість аналізу менших за довжиною послідовностей;

Запропоновано малоресурсну інформаційну технологію, яка дає можливість здійснювати комплексне оцінювання якості генераторів послідовностей псевдовипадкових чисел в умовах обмеженої кількості вхідних даних, виявляти неякісні, ненадійні і скомпроментовані генератори.

Отримані результати будуть корисні для криптографії, стільникових мереж LTE / 5G / 6G, технологій на основі UAV, захисту критичної інформаційної інфраструктури держави. Результати були впроваджені в навчальний процес кафедри комп’ютерних інформаційних технологій Факультету комп’ютерних наук та технологій і в наукову діяльність Науково-дослідної лабораторії протидії кіберзагрозам в авіаційній галузі Національного авіаційного університету і Головного управління розвідки Міністерства оборони України.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Caicedo, J.C.; Lazebnik, S. Active object localization with deep reinforcement learning. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 11–18 December 2015; pp. 2488–2496.

2. Bellver, M.; Giró i Nieto, X.; Marqués, F.; Torres, J. Hierarchical Object Detection with Deep Reinforcement Learning. arXiv 2016, arXiv:1611.03718. Available online: http://arxiv.org/abs/1611.03718 (accessed on 1 February 2018).

3. Dabney, W., Kurth-Nelson, Z., Uchida, N., Starkweather, C. K., Hassabis, D., Munos, R., & Botvinick, M. (2020). A distributional code for value in dopamine-based reinforcement learning. Nature , 577 (7792), 671-675.

4. Jie, Z.; Liang, X.; Feng, J.; Jin, X.; Lu, W.; Yan, S. Tree-structured reinforcement learning for sequential object localization. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 127–135.

5. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In European Conference on Computer Vision; Springer: Amsterdam, The Netherlands, 2016; pp. 21–37.

6. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.

7. Wu, H.; Zhang, H.; Zhang, J.; Xu, F. Fast aircraft detection in satellite images based on convolutional neural networks. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 27–30 September 2015; pp. 4210–4214.

8. Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; Riedmiller, M. Deterministic policy gradient algorithms. In Proceedings of the

31st International Conference on Machine Learning (ICML-14), Beijing, China, 21–26 June 2014; pp. 387–395.

9. Yun, S.; Choi, J.; Yoo, Y.; Yun, K.; Choi, J.Y. Action-Decision Networks for Visual Tracking with Deep Reinforcement Learning. In Proceedings of the 2017 Conference on Computer Vision and Pattern

10. Jayaraman, D.; Grauman, K. Look-ahead before you leap: End-to-end active recognition by forecasting the effect of motion. In European Conference on Computer Vision; Springer: Amsterdam, The Netherlands, 2016; pp. 489–505.

11. Hosang, J.; Benenson, R.; Dollár, P.; Schiele, B. What makes for effective detection proposals? IEEE Trans. Pattern Anal. Mach. Intell. 2016, 38, 814–830.

12. Cheng, M.M.; Zhang, Z.; Lin, W.Y.; Torr, P. BING: Binarized normed gradients for objectness estimation at 300fps. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 3286–3293.

13. Uijlings, J.R.; Van De Sande, K.E.; Gevers, T.; Smeulders, A.W. Selective search for object recognition. Int. J. Comput. Vis. 2013, 104, 154–171. Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1349– 1358.

14. Zitnick, C.L.; Dollár, P. Edge boxes: Locating object proposals from edges. In European Conference on Computer Vision; Springer: Zurich, Switzerland, 2014; pp. 391–405.

15. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. Nature 2015, 518, 529–533.

16. L. Wen, X. Li and L. Gao, "A New Reinforcement Learning Based Learning Rate Scheduler for Convolutional Neural Network in Fault

Classification," in IEEE Transactions on Industrial Electronics, vol. 68, no. 12, pp. 12890-12900, Dec. 2021, doi: 10.1109/TIE.2020.3044808.

17. M. Karimzadeh, A. Esposito, Z. Zhao, T. Braun and S. Sargento, "RL-CNN: Reinforcement Learning-designed Convolutional Neural Network for Urban Traffic Flow Estimation," 2021 International Wireless Communications and Mobile Computing (IWCMC), 2021, pp. 29-34, doi: 10.1109/IWCMC51323.2021.9498948.

18. Li, Y.; Fu, K.; Sun, H.; Sun, X. An Aircraft Detection Framework Based on Reinforcement Learning and Convolutional Neural Networks in Remote Sensing Images. Remote Sens. 2018, 10, 243. https://doi.org/10.3390/rs10020243

**19.** Lee, J.; Kim, J.; Kim, I.; Han, K. Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles. IEEE Access 2019, 7, 165607–165626.

20. Z. Wang, T. Schaul , M. Hessel, H. Van Hasselt, M. Lanctot і N. De Freitas, «Dueling network architectures for deep reinforcement learning», препринт arXiv arXiv:1511.06581, 2015.

21. Collantes, JC Garcia-Escartin, “Quantum random number generators”, Reviews of Modern Physics, Vol. 89, 2017, ст. 015004.

22. І. Гудфеллоу, Ю. Бенгіо, А. Курвіль, «Глибоке навчання», MIT Press, 2016.

23. J. Wang, MJ Pérez-Jiménez, “Forecasting Sunspot Numbers with LSTM”, International Conference on Membrane Computing, Springer, Cham, 2016, pp. 153-166.

A. Vaswani, N. Shazeer , N. Parmar et al, “Attention is all you need”, Advances in neural information processing systems, Vol. 30, 2017, стор. 5998-6008.

24. G. Marsaglia, “Random Number Generators”, Journal of Modern Applied Statistical Methods, Vol. 2, 2003, стор. 2-13.

25. Д. П. Кінгма, Дж. Ба, Адам, «Метод стохастичної оптимізації», препринт arXiv : 1412.6980, 2014.

i. 26 Y. LeCun, Y. Bengio, G. Hinton, “Deep learning”, Nature, Vol. 521, 2015, стор. 436-444.

26. S. Hochreiter, J. Schmidhuber , “Long short-term memory”, Neural computation, Vol. 9, 1997, стор. 1735-1780.

27. Іманбаєв А., Тинимбаєв С., Одарченко Р. та інші, «Дослідження алгоритмів машинного навчання для розробки систем виявлення вторгнень у мобільних мережах 5G і за їх межами», Датчики, 2022, том. 22, випуск 24, ст. 9957.

28. І. Суцкевер , О. Віньялс, К. В. Ле, «Послідовне навчання за допомогою нейронних мереж, досягнення в нейронних системах обробки інформації», том. 27, 2014, стор. 3104-3112.

29. J. Chung, C. Gulcehre , K. Cho, Y. Bengio, «Емпірична оцінка стробованих рекурентних нейронних мереж на моделюванні послідовностей», препринт arXiv arXiv:1412.3555, 2014.

30. Азаров І., Гнатюк С., Олександр М., Азаров І., Мукашева А. «Алгоритми ML у реальному часі для виявлення небезпечних об’єктів у критичних інфраструктурах», Матеріали семінару CEUR, 2023, том. 3373, стор. 217-226.

31. Іашвілі Г., Авкурова З., Явич М., Бауиржан М., Гагнідзе А., Гнатюк С. «Підхід машинного навчання на основі вмісту для системи виявлення вразливостей апаратного забезпечення», Конспект лекцій з інженерії даних та комунікаційних технологій, том. 83, стор. 117-126, 2021.

32. J. Aldama, S. Sarmiento, IH López Grande, S. Signorini, LT Vidarte and V. Pruneri , “Integrated QKD and QRNG Photonic Technologies,” in Journal of Lightwave Technology, vol. 40, вип. 23, стор. 7498-7517, 1 грудня 2022 р.

33. Фор Е., Щерба А., Василіу Ю., Фесенко А. «Метод обміну криптографічним ключем для факторного кодування даних», Матеріали семінару CEUR, 2020, том. 2654, стор. 643 - 653.

34. Р. Куанг, Д. Лу, А. Хе, К. Маккензі та М. Реддінг, «Псевдоквантовий генератор випадкових чисел із квантовою перестановкою», 2021 Міжнародна конференція IEEE з квантових обчислень та інженерії (QCE), Брумфілд, Колорадо, США , 2021, стор. 359-364.

35. М. Гупта та М. Дж. Нене, «Генерація випадкової послідовності з використанням надпровідних кубітів», Третя міжнародна конференція з інтелектуальних комунікаційних технологій і віртуальних мобільних мереж (ICICV) 2021 р., Тірунелвелі, Індія

36. Hochreiter, S., & Schmidhuber , J. (1997). Довга короткочасна пам'ять. Нейронні обчислення, 9(8), 1735-1780.

37. Чо, К., ван Мерріенбоер , Б., Гюльсере , К., Багданау , Д., Бугарес , Ф., Швенк, Х., і Бенгіо, Ю. (2014). Вивчення подання фраз за допомогою кодувальника-декодера RNN для статистичного машинного перекладу. arXiv:1406.1078.

38. Sutskever , I., Vinyals, O., & Le, QV (2014). Послідовне навчання за допомогою нейронних мереж. Досягнення в нейронних системах обробки інформації, 27.

39. Браунлі, Дж. (2018). Глибоке навчання для прогнозування часових рядів: прогнозуйте майбутнє за допомогою MLP, CNN і LSTM на Python. Майстерність машинного навчання.

40. Герс, Ф.А., Шмідхубер , Дж., і Каммінс, Ф. (2000). Навчання забувати: постійне передбачення з LSTM. Нейронні обчислення, 12 (10), 2451-2471.

41. Грейвс, А., Мохамед, А.-Р., і Хінтон, Г. (2013). Розпізнавання мовлення за допомогою глибоких рекуррентних нейронних

мереж. 2013 Міжнародна конференція IEEE з акустики, обробки мови та сигналів.

42. Лекун, Ю., Бенгіо, Ю., і Хінтон, Г. (2015). Глибоке навчання. Природа, 521 (7553), 436-444.

43. Гудфеллоу І., Бенгіо Ю. та Курвіль А. (2016). Глибоке навчання. MIT Press.

44. Vaswani, A., Shazeer , N., Parmar, N., Uszkoreit , J., Jones, L., Gomez, AN, Kaiser, Ł., & Polosukhin , I. (2017). Увага — це все, що вам потрібно. Досягнення нейронних систем обробки інформації, 30.

45. Карпати, А. (2015). Невиправдана ефективність рекурентних нейронних мереж. Блог Андрія Карпати.

46. Desai, V., Patil, R., & Rao, D. (2012). Використання шарової рекурентної нейронної мережі для генерації послідовностей псевдовипадкових чисел. International Journal of Computer Science Issues, 9, 324–334.

47. Окада, К., Ендо, К., Ясуока , К., і Курабаяші , С. (2023). Навчений генератор псевдовипадкових чисел: WGAN-GP для генерації статистично надійних випадкових чисел. PLoS One, 18(6): e0287025. doi:10.1371/journal.pone.0287025. PMID: 37315028; PMCID: PMC10266608.

48. О'Ніл, ME (2014). PCG: сімейство простих швидких статистично ефективних алгоритмів для генерації випадкових чисел.

49. Де Мікко, Л., Антонеллі, М., Россо, О.А. (2021). Від хаотичних систем безперервного часу до генераторів псевдовипадкових чисел: аналіз і узагальнена методологія. Ентропія (Базель), 23(6):671. doi:10.3390/e23060671. PMID: 34073348; PMCID: PMC8229976.

50. Мацумото, М., і Нісімура, Т. (2015). Динамічне створення генераторів псевдовипадкових чисел.

51. Джонсон, Д. (2018). Генератори випадкових чисел - Принципи та практика. ISBN: 9781501506260.

52. Cesa -Bianchi, N., & Lugosi, G. (2006). Прогнозування, навчання та ігри. ISBN: 9781139454827.

53. Відинський, Б. (2017). Середньоквадратична послідовність Вейля RNG.

54. Зеніл, Х. (2011). Випадковість через обчислення: кілька відповідей, більше запитань. ISBN: 9789814327749.

55. Райнер, Б., Колміцер , К., Расс , С., і Шауер, С. (2020). Квантова генерація випадкових чисел: теорія і практика. ISBN: 9783319725949.

56. Іслам, М., Чен, Г., і Джин, С. (2019). Огляд нейронної мережі. Американський журнал нейронних мереж і програм, 5(1), 7-11. doi:10.11648/j.ajnna.20190501.12.

57. О'Ші, К., Неш, Р. (2015). Вступ до згорткових нейронних мереж.

58. Лі З., Ян В., Пен С. та Лю Ф. (2021). Огляд згорткових нейронних мереж: аналіз, застосування та перспективи.

59. Лін Т., Гуо Т. та Аберер К. Гібридні нейронні мережі для вивчення тенденції часових рядів.

60. Y.M. Koukou, S.H. Othman, M. M. Siraj and H. Nkiama. (2016) .Comparative Study of AES, Blowfish, CAST-128 and DES Encryption Algorithm. IOSR Journal of Engineering (IOSRJEN), Vol. 6, Issue 6, 1-7.

61. Nidhi Singhal, J.P.S.Raina. (2011). Comparative Analysis of AES and RC4 Algorithms for Better Utilization. International Journal of Computer Trends and Technology (IJCTT), V1(3), 259-263.

62. B. J. S. Kumar, V. K. R. Raj and A. Nair. (2017). Comparative study on AES and RSA algorithm for medical images. International Conference on Communication and Signal Processing (ICCSP), 501-504.

63. A. Nadeem and M. Y. Javed. (2005). A Performance Comparison of Data Encryption Algorithms. International Conference on Information and Communication Technologies, 84-89.

64. A. K. Mandal, C. Parakash, and A. Tiwari. (2012). A. Performance evaluation of cryptographic algorithms: DES and AES. IEEE Students' Conference on Electrical, Electronics and Computer Science, 1-5.

65. T. Khoei, E. Ghribi, P. Ranganathan, N. Kaabouch. (2021). A performance comparison of encryption/decryption algorithms for UAV swarm communications. Academic Press, V1, 1-5.

66. Usman, M., Amin, R., Aldabbas, H., Alouffi, B. (2022). Lightweight Challenge-Response Authentication in SDN-Based UAVs Using Elliptic Curve Cryptography. Electronics 2022, Vol. 11, 1026. https://doi.org/10.3390/ electronics11071026.

67. Muslum Ozgur Ozmen, Rouzbeh Behnia, Attila A Yavuz. (2019). IoD-crypt: A lightweight cryptographic framework for Internet of drones. arXiv, Vol. 1.

68. F. Syafaat, A. Farhan. (2019). Implementation of AES-128 Cryptography on Unmanned Aerial Vehicle and Ground Control System. Teknik Informatika – Universitas Komputer Indonesia, 10-19.

69. F. Ronaldo, D. Pramadihanto and A. Sudarsono. (2020). Secure Communication System of Drone Service using Hybrid Cryptography over 4G/LTE Network. 2020 International Electronics Symposium (IES), 116-122. doi: 10.1109/IES50839.2020.9231951.

70. Bafandehkar, Mohsen et al. (2013). Comparison of ECC and RSA Algorithm in Resource Constrained Devices. 2013 International Conference on IT Convergence and Security (ICITCS), 1-3.

71. Bansal, Malti et al. (2021). Comparison of ECC and RSA Algorithm with DNA Encoding for IoT Security. 2021 6th International Conference on Inventive Computation Technologies (ICICT), 1340-1343.

72. Nagesh, O Sri and Vankamamidi Srinivasa Naresh. (2020). Comparative Analysis of MOD-ECDH Algorithm and Various Algorithms. International Journal of Industrial Engineering & Production Research, Vol. 31, No. 2, 301-308.

73. Mahto, Dindayal et al. (2016). Security Analysis of Elliptic Curve Cryptography and RSA. Proceedings of the World Congress on Engineering 2016, Vol. I, 1-4.

74. S. Jhajharia, S. Mishra and S. Bali. (2013). Public key cryptography using neural networks and genetic algorithms. 2013 Sixth International Conference on Contemporary Computing (IC3), 137-142.

75. B. Chavali, S. K. Khatri and S. A. Hossain. (2020). AI and Blockchain Integration. 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 548-552.

76. T. Dong and T. Huang. (2020). Neural Cryptography Based on Complex-Valued Neural Network. IEEE Transactions on Neural Networks and Learning Systems, Vol. 31, No. 11, 4999-5004.

77. M. Danziger and M. A. Amaral Henriques. (2014). Improved cryptanalysis combining differential and artificial neural network schemes. 2014 International Telecommunications Symposium (ITS), 1-5.

78. Y. Xiao, Q. Hao and D. D. Yao. (2019). Neural Cryptanalysis: Metrics, Methodology, and Applications in CPS Ciphers. 2019 IEEE Conference on Dependable and Secure Computing (DSC), 1-8.

79. Psichogios, DC, & Ungar, LH. Гібридна нейронна мережа – підхід до моделювання процесів.

80. І. Гудфеллоу, Ю. Бенгіо та А. Курвіль, Глибоке навчання. Кембридж, Массачусетс: MIT Press, 2016.

81. P. L'Ecuyer і R. Simard, "TestU01: Бібліотека AC для емпіричного тестування генераторів випадкових чисел", ACM Trans. математика Програмне забезпечення, вип. 33, вип. 4, стор. 22, 2007.

82. G. Marsaglia, The Marsaglia Random Number CDROM, включаючи Diehard Battery of Tests of Randomness. 1995 рік.

83. Д. Е. Кнут, Мистецтво комп’ютерного програмування, Том 2: Напівчислові алгоритми, 3-є видання. Редінг, Массачусетс: Аддісон-Уеслі, 1997.

84. NIST (Національний інститут стандартів і технологій), спеціальна публікація NIST 800-22: набір статистичних тестів для генераторів випадкових і псевдовипадкових чисел для криптографічних програм. 2010 рік.

85. S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Comput ., vol. 9, № 8, стор. 1735-1780, 1997.

86. Y. Lecun , Y. Bengio та G. Hinton, "Глибоке навчання", Nature, vol. 521, вип. 7553, стор. 436-444, 2015.

87. А. Грейвс, А. Мохамед і Г. Хінтон, «Розпізнавання мовлення за допомогою глибоких рекурентних нейронних мереж», у Proc. IEEE Int. конф. Acoustic., Speech, Signal Process., стор. 6645-6649, IEEE, 2013.

88. Х. Абді, "Праймер нейронної мережі", J. Biol. Syst., том. 2, стор. 247-281, 1994.

89. LE Bassham III, AL Rukhin , J. Soto, JR Nechvatal , ME Smid , EB Barker, SD Leigh, M. Levenson, M. Vangel , DL Banks, et al., " Sp 800-22 rev. 1a. статистичний тест набір для генераторів випадкових і псевдовипадкових чисел для криптографічних програм", 2010.

90. Ryabyy M., Prystavka P., Kinzeryavyy O., Proskurin D. An Advanced Method of Compressing Digital Images as Part of a Video Stream to Pre-Process the Data Before Encrypting, 2022 (прийнято до друку).

91. Puleko I., Svintsytska O., Chumakevych V., Ptashnyk V, Polishchuk Yu., The Scalar Metric of Classification Algorithm Choice in Machine Learning Problems Based on the Scheme of Nonlinear Compromises, CEUR Workshop Proceedings, 2022, vol. 3171, pp. 1066-1075.

92. Gnatyuk S., Kinzeryavyy V., Polishchuk Y., & Horbakha B., Analysis of methods of ensuring data confidentiality transmitted from UAV, CEUR Workshop Proceedings, 2022 (прийнято до друку).

93. Tynymbayev S., Ibraimov M., Namazbayev T., Gnatyuk S. Development of pipelined polynomial multiplier modulo irreducible polynomials for cryptosystems, Eastern-European Journal of Enterprise Technologies, 2022, Vol. 1, Issue 4-115, pp. 37-43.

94. Proskurin D., Gnatyuk S., Bauyrzhan M. Distributive Training Can Improve Neural Network Performance based on RL-CNN Architecture, CEUR Workshop Proceedings, 2022, vol. 3187, pp. 48-57.

95. Рябий М.О., Кінзерявий О.М., Проскурін Д.П., Удосконалений метод стиснення цифрових зображень, як частини відеопотоку для попередньої обробки даних перед їх шифруванням, Проблеми інформатизації та управління, 2022, Т. 2, № 70, с. 200-207.

96. Gnatyuk, S., Kinzeryavyy, V., Polishchuk, Y., Nechyporuk, O., & Horbakha, B., Аналіз методів забезпечення конфіденційності даних, які передаються з БПЛА. Електронне фахове наукове видання "Кібербезпека: освіта, наука, техніка", 1(17), 167-186. https://doi.org/10.28925/2663-4023.2022.17.167186

97. Тынымбаев С., Гнатюк С.А., Шайкулова А.А., Әділбекқызы С. Аппаратное формирование ключей RSA, Вестник АУЭС, 2022, №2, с. 168-176

98. Kinzeryavyy, V., Polishchuk, Y., & Horbakha, B., Датасет криптографічних алгоритмів для забезпечення ефективності захисту інформації безпілотними літальними апаратами. Електронне фахове наукове видання "Кібербезпека: освіта, наука, техніка", 1(17) (прийнято до друку).

99. V. Desai, R. Patil і D. Rao, "Використання шарової рекурентної нейронної мережі для генерації послідовностей псевдовипадкових чисел", Int. J. Comput. Sci. Питання, вип. 9, стор. 324-334, 2012.

100. Дж. М. Хьюз, «Генерація псевдовипадкових чисел за допомогою двійкових рекурентних нейронних мереж», доктор філософії. дис., 2007.

101. М. Де Бернарді та П. Малакарія, «Генерація псевдовипадкових чисел за допомогою генеративних змагальних мереж», у Workshop Proc.

102. В. Мніх, К. Кавукчуоглу , Д. Сільвер, А. Грейвс, І. Антоноглу , Д. Вірстра та М. А. Рідміллер , «Гра в Atari з глибоким навчанням з підкріпленням», ArXiv ! abs/1312.5602, 2013.

103. Т. Шаул, Дж. Куан, І. Антоноглу та Д. Сільвер, «Пріоритетне повторення досвіду», препринт arXiv arXiv:1511.05952, 2015.

104. Дж. Шульман, П. Моріц, С. Левін, М. Джордан і П. Аббіл, «Безперервне багатовимірне керування з використанням узагальненої оцінки переваг», препринт arXiv arXiv:1506.02438, 2015.

105. Дж. Шульман, Ф. Вольскі, П. Дарівал , А. Редфорд і О. Клімов , «Алгоритми оптимізації проксимальної політики», препринт arXiv: arXiv:1707.06347, 2017.

106. Р. Саттон і А. Барто, Навчання з підкріпленням: Вступ, серія «Адаптивні обчислення та машинне навчання». Cambridge, MA: MIT Press, 2018.

107. Грищук, Р. В. (2013). Методика оцінювання рівня небезпеки кібернетичних загроз. Сучасний захист інформації, Спецвипуск, С. 23-28.

108. Грищук, Р., Даник, Ю., (2010) Основи кібернетичної безпеки, ЖНАЕУ, 636 с.

109. С. Саттон, Д. А. МакАллестер, С. П. Сінгх та Ю. Мансур, «Методи градієнта політики для навчання з підкріпленням із наближенням функцій», у Adv. Neural Inf. процес. Syst., стор. 1057–1063, 2000.

110. Proskurin D., Gnatyuk S., Okhrimenko T., Iavich M. ML-Based Cryptographic Keys Quality Assessment for 5G / 6G Networks Privacy and Security, Proceedings of the IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS. 2023. С. 1025-1030.

111. Gnatyuk S., Okhrimenko A., Navrotskyi D., Proskurin D., Horbakha B. Dataset of Cryptographic Algorithms for UAV Image Encryption based on Artificial Neural Networks, CEUR Workshop Proceedings. 2023. Вип. 3504. С. 63-71.

112. Hu Z., Ryabyy M., Prystavka P., Janisz K., Proskurin D. Advanced Method for Compressing Digital Images as a Part of Video Stream to Pre-processing of UAV Data Before Encryption, Lecture Notes on Data Engineering and Communications Technologies. 2023. Вип. 181. С. 371-381.

113. Proskurin D., Gnatyuk S., Okhrimenko T. Predicting Pseudo-Random and Quantum Random Number Sequences using Hybrid Deep Learning Models, CEUR Workshop Proceedings. 2023. Вип. 3426. С. 77-88.

114. Proskurin D., Gnatyuk S., Bauyrzhan M. Distributive Training Can Improve Neural Network Performance based on RL-CNN Architecture, CEUR Workshop Proceedings. 2021. Вип. 3187. С. 48-57.

115. Рябий М., Кінзерявий О., Проскурін Д., Сорокопуд В. An advanced method of compressing digital images as part of a video stream to pre-process the data before encrypting, Проблеми інформатизації та управління. 2023. Т. 1, № 73. С. 128-137.

116. Гнатюк С.О., Поліщук Ю.Я., Кінзерявий В.М., Горбаха Б.М., Проскурін Д.П. Формування датасету криптоалгоритмів для забезпечення конфіденційності даних, які передаються з розвідувально-пошукового БПЛА, Кібербезпека: освіта, наука, техніка. 2023. № 4 (20). С. 205–219.

117. Проскурін Д.П., Явіч М.П., Гнатюк С.О. Модель ідентифікації джерела послідовностей псевдовипадкових чисел на основі гібридної нейронної мережі, Проблеми інформатизації та управління. 2024. Т. 1, № 73. С. 54-62.

118. Проскурін Д.П., Гнатюк С.О. Дистрибутивне навчання покращує роботу нейронних мереж на основі RL-CNN архітектури, ABIA-2021: XVI міжнар. наук.-техн. конф., 20-22 квітня 2021 р.: тези доп. Київ: НАУ, 2021. С. 16.14-16.17.

119. Гнатюк С.О., Проскурін Д.П. Імплементація дистрибутивного навчання покращує роботу RL-CNN архітектури для ідентифікації об'єктів на зображеннях // Всеукраїнська науково-практична інтернет-конференція здобувачів вищої освіти і молодих учених «Інформаційно-комп’ютерні технології: стан, досягнення та перспективи розвитку», 25-26 листопада 2021, Житомир, Україна.

120. Proskurin D. P. Assessing Randomness in Number Sequences in Cryptography: A Comparative Study of the Chi-Squared Test and Neural Network-Based Approaches, EEML 2023: Eastern European Machine Learning Conference, June 2023.

121. Проскурін Д.П., Гнатюк С.О. Підхід до оцінювання рівня випадковості бінарних послідовностей на основі одновимірної згорткової нейронної мережі, Information, Communication, Society (ICS-2023), 18-20 травня 2023 р., Зозулі (Львівська область), Україна.

122. Проскурін Д.П., Гнатюк С.О. Оцінювання випадковості бінарних послідовностей на основі одновимірної згорткової нейронної мережі 1D-CNN для криптографічних застосувань // Новітні дослідження культури і мистецтва: пошуки, проблеми, перспективи : матеріали Всеукр. наук.-практ. конф. / М-во культ. України та інформ. політики ; Нац. акад. кер. кадрів культ. і мистец. ; Наук. тов. студ., асп., доктор. і молод. вч. (Київ, 18 травня 2023 р.). Київ : НАКККіМ, 2023. С. 27-32.

# ДОДАТОК А

LCG генератор

```
# Linear Congruential Generator
def lcg(seed, a, c, m, n):
    numbers = []
    for _ in range(n):
        seed = (a * seed + c) % m
        numbers.append(seed)
    return numbers
```

# ДОДАТОК Б

MT генератор

```
# Mersenne Twister
def mersenne_twister(seed, n):
    np.random.seed(seed)
    # Adjust the upper bound to fit within the int32 range
    return np.random.randint(0, 2**31 - 1, n).tolist()
```

# ДОДАТОК В

XS генератор

```
# Xorshift
def xorshift(seed, n):
    numbers = []
    for _ in range(n):
        seed ^= (seed << 13)
        seed ^= (seed >> 17)
        seed ^= (seed << 5)
        numbers.append(seed & (2**32 - 1))
    return numbers
```

# ДОДАТОК Г

MS генератор

```
# Middle Square Method
def middle_square(seed, n, w):
    numbers = []
    for _ in range(n):
        seed = int(str(seed * seed).zfill(2 * w)[w//2:-w//2])
        if seed == 0:
            break  # Optionally break if it converges to zero
        numbers.append(seed % (2**w))  # Optional: keep it within w
digits
    return numbers
```

# ДОДАТОК Д

LFSR генератор

```
# Linear Feedback Shift Register
def lfsr(seed, taps, n):
    numbers = []
    for _ in range(n):
        bit = sum([seed >> i & 1 for i in taps]) % 2
        seed = (seed >> 1) | (bit << 31)
        numbers.append(seed)
    return numbers
```

# ДОДАТОК Е

ACORN генератор

```
# Additive Congruential Random Number Generator
def acorn(seed, a, n):
    numbers = [seed]
    for _ in range(1, n):
        seed = (numbers[-1] + a) % 2**32
        numbers.append(seed)
    return numbers
```

## ДОДАТОК Є

BBS генератор

```
# Blum Blum Shub
def blum_blum_shub(seed, n, p=383, q=503):
    m = p * q
    numbers = []
    for _ in range(n):
        seed = pow(seed, 2, m)
        numbers.append(seed)
    return numbers
```

# ДОДАТОК Ж

ССЗ0 генератор

```
# ChaCha20
def chacha20_random(n):

    key = os.urandom(32) # 256-bit key
    nonce = os.urandom(16) # 96-bit nonce, unique for each operation
    algorithm = algorithms.ChaCha20(key, nonce)
    cipher = Cipher(algorithm, mode=None, backend=default_backend())

    encryptor = cipher.encryptor()
    plaintext = b'\x00' * n # Assuming you want n bytes of random data
    random_data = encryptor.update(plaintext)
    return random_data
```

# ДОДАТОК З

Підготовка даних

```
def create_single_input_datasets(sequence_length):
    datasets = {
        "LCG": create_normalized_sequences(lcg(seed, lcg_a, lcg_c, lcg_m, n),
sequence_length),
        "MT": create_normalized_sequences(mersenne_twister(seed, n), sequence_length),
        "XS": create_normalized_sequences(xorshift(seed, n), sequence_length),
        "MS": create_normalized_sequences(middle_square(seed, n, mdq_w), sequence_length),
        "LFSR": create_normalized_sequences(lfsr(seed, lfsr_taps, n), sequence_length),
        "ACORN": create_normalized_sequences(acorn(seed, acorn_a, n), sequence_length),
        "BBS": create_normalized_sequences(blum_blum_shub(seed, n, bbs_p, bbs_q),
sequence_length)
    }

    # Modify the handling of the chacha20_random output
    chacha20_data = chacha20_random(n)  # Assuming this is where you get the
binary/hexadecimal data
    chacha20_data_integers = binary_data_to_integers(chacha20_data)
    datasets["CC20"] = create_normalized_sequences(chacha20_data_integers,
sequence_length)

    return datasets
```

## ДОДАТОК И

Дані для тренування, тестування та валідації

```
def transform_dataset_split_data(datasets, prng):
    x, y = datasets[prng]
    num_samples = x.shape[0]

    # Calculate the split indices
    train_end = int(0.7 * num_samples)
    val_end = train_end + int(0.2 * num_samples)

    # Split the data
    x_train = x[:train_end]
    y_train = y[:train_end]

    x_val = x[train_end:val_end]
    y_val = y[train_end:val_end]

    x_test = x[val_end:]
    y_test = y[val_end:]

    return x_train, x_val, x_test, y_train, y_val, y_test
```

# ДОДАТОК І

Оптимізація даних

```
def transform_dataset_prng_prediction(prng_datasets, sequence_length, test_size=0.2,
validation_size=0.1):
    combined_data = []
    combined_labels = []

    # Function to split sequences into chunks
    def chunk_sequence(seq, chunk_size):
        return [seq[i:i + chunk_size] for i in range(0, len(seq), chunk_size) if len(seq[i:i +
chunk_size]) == chunk_size]

    # Combine all sequences and labels, with sequences broken into chunks
    for label, sequences in prng_datasets.items():
        for seq in sequences:
            # Break each sequence into chunks of size 'sequence_length'
            chunks = chunk_sequence(seq, sequence_length)
            for chunk in chunks:
                combined_data.append(chunk)
                combined_labels.append(label)

    label_to_id = {label: idx for idx, label in enumerate(prng_datasets.keys())}
    categorical_labels = np.array([label_to_id[label] for label in combined_labels])

    # Adjust sizes based on the new dataset size
    smallest_class_size = min([len(sequences) // sequence_length for sequences in
prng_datasets.values()])
    test_size = max(1, int(len(categorical_labels) * test_size))
    validation_size = max(1, int(len(categorical_labels) * validation_size))

    # Split the data into training and test sets
    X_train, X_temp, y_train, y_temp = train_test_split(
        combined_data, categorical_labels, test_size=test_size + validation_size,
stratify=categorical_labels
    )

    # Further split the test set into validation and test sets
    X_val, X_test, y_val, y_test = train_test_split(
        X_temp, y_temp, test_size=test_size / (test_size + validation_size), stratify=y_temp
    )

    return standardize_prng_dataset(X_train, sequence_length),
standardize_prng_dataset(X_val, sequence_length), standardize_prng_dataset(X_test,
sequence_length), np.array(y_train), np.array(y_val), np.array(y_test)
```

# ДОДАТОК Ї

Згорткова нейронна мережа

```
def create_cnn_model(neurons, dropout_rate, activation, input_shape, num_conv_layers=3,
conv_filters=None, kernel_size=2, pool_size=2):
    model = Sequential()

    # Define the number of filters for each convolutional layer
    if conv_filters is None:
        conv_filters = [neurons // (2 ** i) for i in range(num_conv_layers)]

    # Input layer
    model.add(Conv1D(filters=conv_filters[0], kernel_size=kernel_size, activation=activation,
input_shape=input_shape))
    model.add(MaxPooling1D(pool_size=pool_size))

    # Additional convolutional layers
    for filters in conv_filters[1:]:
        # Check to prevent too much downsampling
        if model.output_shape[1] <= pool_size:
            break
        model.add(Conv1D(filters=filters, kernel_size=kernel_size, activation=activation))
        model.add(MaxPooling1D(pool_size=pool_size))

    # Flatten and Dense layers
    model.add(Flatten())
    model.add(Dense(neurons, activation=activation))
    model.add(Dropout(dropout_rate))
    model.add(Dense(1))  # Output layer for single input prediction

    model.compile(optimizer='adam', loss='mse')
    return model
```

# ДОДАТОК Й

Рекурентна нейронна мережа

```
def create_rnn_lstm_model(model_type, neurons, dropout_rate, activation, input_shape, num_layers=3):
    model = Sequential()

    # Choose the correct layer type based on model_type
    LayerType = SimpleRNN if model_type == 'RNN' else LSTM

    # Input layer
    model.add(LayerType(neurons, activation=activation, return_sequences=num_layers > 1, input_shape=input_shape))

    # Additional RNN or LSTM layers
    for i in range(1, num_layers):
        model.add(LayerType(neurons, activation=activation, return_sequences=i < num_layers - 1))

    # Dense layers
    model.add(Flatten())  # Flatten the output for the Dense layer if the last RNN/LSTM layer has return_sequences=True
    model.add(Dense(neurons, activation=activation))
    model.add(Dropout(dropout_rate))
    model.add(Dense(1))  # Output layer for single input prediction

    model.compile(optimizer='adam', loss='mse')
    return model
```

## ДОДАТОК К

Гібридна нейронна мережа

```
def create_hybrid_cnn_lstm_model(neurons, dropout_rate, activation, input_shape, num_cnn_layers, num_lstm_layers, cnn_filters, kernel_size, pool_size, lstm_units):
    model = Sequential()

    # Adjust input_shape for TimeDistributed layer
    # input_shape should be (timesteps, subsequences, features)
    adjusted_input_shape = (input_shape[0], 1, input_shape[1])

    # CNN layers within TimeDistributed wrapper for sequence data
    model.add(TimeDistributed(Conv1D(filters=cnn_filters, kernel_size=kernel_size, activation=activation, padding='same'), input_shape=adjusted_input_shape))

    # Check if the sequence length is sufficient for pooling
    if adjusted_input_shape[1] // pool_size > 0:
        model.add(TimeDistributed(MaxPooling1D(pool_size=min(pool_size, adjusted_input_shape[1]))))  # Adjusted pooling size
        adjusted_input_shape = (adjusted_input_shape[0], adjusted_input_shape[1] // pool_size, adjusted_input_shape[2])
    else:
        pool_size = 1
        model.add(TimeDistributed(MaxPooling1D(pool_size=pool_size)))
        adjusted_input_shape = (adjusted_input_shape[0], adjusted_input_shape[1] // pool_size, adjusted_input_shape[2])

    # Additional CNN layers
    for _ in range(1, num_cnn_layers):
        model.add(TimeDistributed(Conv1D(filters=cnn_filters, kernel_size=kernel_size, activation=activation, padding='same')))
        # Check if the sequence length is sufficient for pooling
        if adjusted_input_shape[1] // pool_size > 0:
            model.add(TimeDistributed(MaxPooling1D(pool_size=min(pool_size, adjusted_input_shape[1]))))  # Adjusted pooling size
            adjusted_input_shape = (adjusted_input_shape[0], adjusted_input_shape[1] // pool_size, adjusted_input_shape[2])
        else:
            break  # Break the loop if the sequence length is too small for further pooling

    # Flatten the output of CNN layers before feeding it into LSTM layers
    model.add(TimeDistributed(Flatten()))

    # LSTM layers
    for i in range(num_lstm_layers):
        model.add(LSTM(lstm_units, activation=activation, return_sequences=True if i < num_lstm_layers - 1 else False))

    # Dense layers
    model.add(Dense(neurons, activation=activation))
    model.add(Dropout(dropout_rate))
    model.add(Dense(1))  # Adjusted for single output, change if necessary

    model.compile(optimizer='adam', loss='mse')
    return model
```

# ДОДАТОК Л

Тренування гібридної нейронної мережі з одиничним виходом

```
    # Hybrid
    hybrid_model = create_hybrid_cnn_lstm_model(
        neuron,
        dropout_rate=0.0,
        activation=activation,
        input_shape=(x_train.shape[1], 1),
        num_cnn_layers=model_layer,
        num_lstm_layers=model_layer,
        cnn_filters=64,  # Number of filters in the Conv1D layers
        kernel_size=3,   # Kernel size for the Conv1D layers
        pool_size=2,     # Pool size for the MaxPooling1D layers
        lstm_units=64
    )

    early_stopping = prepare_early_stopping()
    hybrid_history = hybrid_model.fit(x_train, y_train, epochs=epoch, validation_data=(x_val,
y_val), callbacks=[early_stopping])
    process_and_log_results('Hybrid', hybrid_model, hybrid_history, x_test, y_test, scaler,
graph_title+f'_ep.{early_stopping.stopped_epoch}', False)
    hybrid_model.save(f'{model_save_prefix}Hybrid_{graph_title}.h5')
```

# ДОДАТОК М

Тренування  згорткової нейронної мережі з одиничним виходом

```
    # CNN
    cnn_model = create_cnn_model(
        neuron,
        dropout_rate=0.0,
        activation=activation,
        input_shape=(x_train.shape[1], 1),
        num_conv_layers=model_layer,
        conv_filters=[100, 80, 60, 40, 20],
        kernel_size=3,
        pool_size=2
    )

    early_stopping = prepare_early_stopping()
    cnn_history = cnn_model.fit(x_train, y_train, epochs=epoch, validation_data=(x_val, y_val),
callbacks=[early_stopping])
    process_and_log_results('CNN', cnn_model, cnn_history, x_test, y_test, scaler,
graph_title+f'_ep.{early_stopping.stopped_epoch}', False)
    cnn_model.save(f'{model_save_prefix}CNN_{graph_title}.h5')
```

# ДОДАТОК Н

Тренування рекурентної нейронної мережі з одиничним виходом

```
    # RNN
    rnn_model = create_rnn_lstm_model(
        'RNN',
        neuron,
        dropout_rate=0.0,
        activation=activation,
        input_shape=(x_train.shape[1], 1),
        num_layers=model_layer
    )

    early_stopping = prepare_early_stopping()
    rnn_history = rnn_model.fit(x_train, y_train, epochs=epoch, validation_data=(x_val, y_val),
callbacks=[early_stopping])
    process_and_log_results('RNN', rnn_model, rnn_history, x_test, y_test, scaler,
graph_title+f'_ep.{early_stopping.stopped_epoch}', False)
    rnn_model.save(f'{model_save_prefix}RNN_{graph_title}.h5')

    # LSTM
    lstm_model = create_rnn_lstm_model(
        'LSTM',
        neuron,
        dropout_rate=0.0,
        activation=activation,
        input_shape=(x_train.shape[1], 1),
        num_layers=model_layer
    )

    early_stopping = prepare_early_stopping()
    lstm_history = lstm_model.fit(x_train, y_train, epochs=epoch, validation_data=(x_val,
y_val), callbacks=[early_stopping])
    process_and_log_results('LSTM', lstm_model, lstm_history, x_test, y_test, scaler,
graph_title+f'_ep.{early_stopping.stopped_epoch}', False)
    lstm_model.save(f'{model_save_prefix}LSTM_{graph_title}.h5')
```

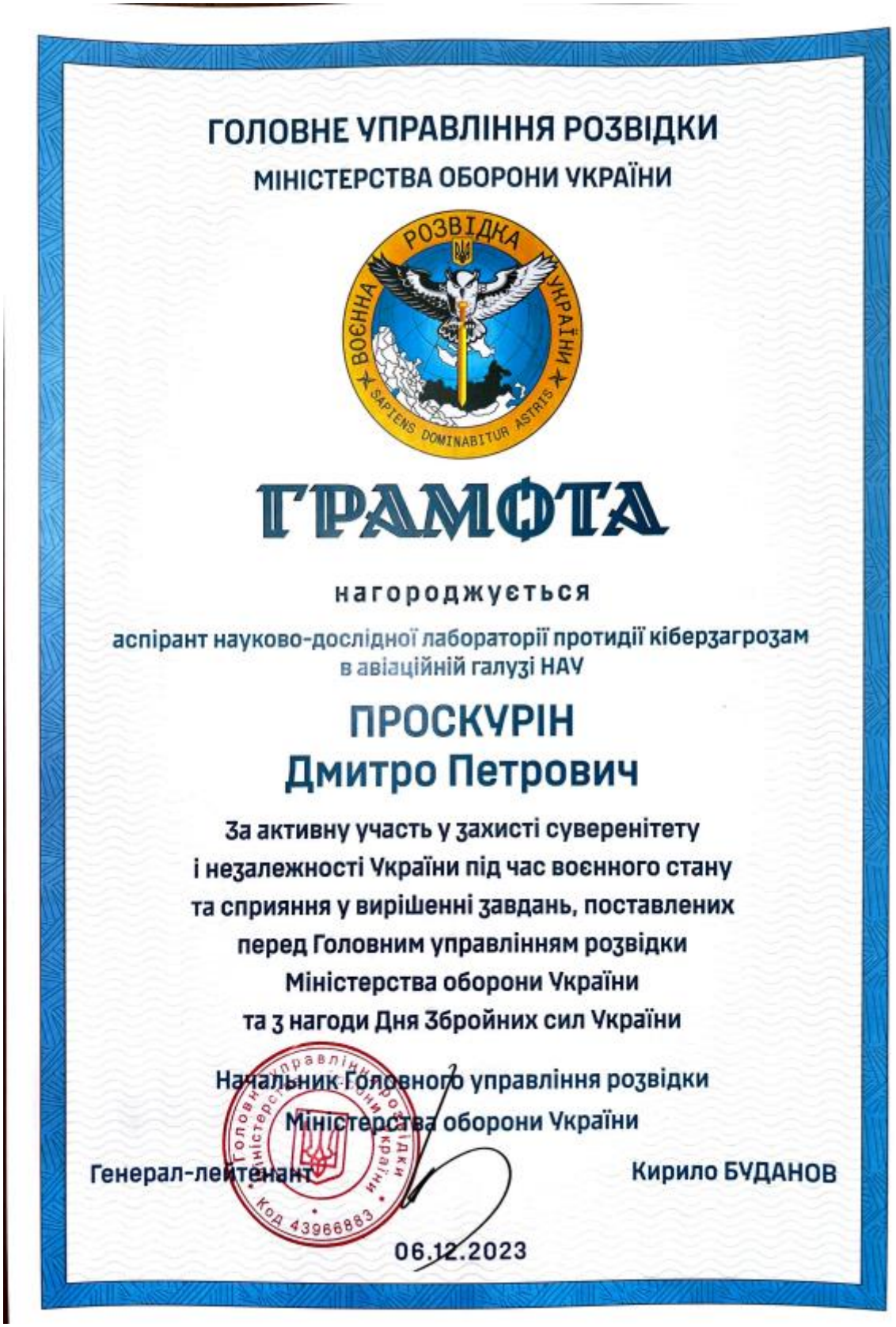
# ДОДАТОК О

Приклад згенерованих послідовностей

| | | |
|---|---|---|
| 87628868 | 1845187042 | 3336926330 |
| 71072467 | 1675701733 | 3833858479 |
| 2332836374 | 1358822685 | 2691347863 |
| 2726892157 | 561383553 | 2301620635 |
| 3908547000 | 789925284 | 3911280821 |
| 483019191 | 170765737 | 1003192174 |
| 2129828778 | 878579710 | 3888331485 |
| 2355140353 | 1402032510 | 3622373553 |
| 2560230508 | 290876773 | 2178099726 |
| 3364893915 | 137773602 | 2143326375 |
| 171172990 | 410361373 | 3716714477 |
| 3194601925 | 1959836417 | 3370611511 |
| 4148119648 | 1995074678 | 2590543511 |
| 316399679 | 1983958385 | 3016540682 |
| 3004788882 | 657890619 | 3670208076 |
| 1976948425 | 1819941518 | 133207273 |
| 1702883732 | 1069045865 | 2249749951 |
| 4121112547 | 1605979227 | 817327336 |
| 1744294886 | 659577591 | 3809463773 |
| 4092090893 | 665605494 | 1107980391 |
| 4267815944 | 1063926992 | 1220729572 |
| 2850572231 | 1685103474 | 706328139 |
| 1816322682 | 1981297353 | 4010564675 |
| 2147159953 | 115061003 | 697650721 |
| 1233879996 | 36027469 | 620097321 |
| 1818069995 | 1251993226 | 1568012161 |
| 2900471886 | 457175121 | 3117739555 |
| 667024213 | 1712592594 | 823007563 |
| 381260976 | 1282922662 | 3093302104 |
| 3662258255 | 1319794951 | 880847997 |

**ДОДАТОК П**

Грамота ГУР



ГОЛОВНЕ УПРАВЛІННЯ РОЗВІДКИ
МІНІСТЕРСТВА ОБОРОНИ УКРАЇНИ

РОЗВІДКА ВОЄННА УКРАЇНИ
SAPIENS DOMINABITUR ASTRIS

ГРАМОТА

нагороджується

аспірант науково-дослідної лабораторії протидії кіберзагрозам
в авіаційній галузі НАУ

ПРОСКУРІН
Дмитро Петрович

За активну участь у захисті суверенітету
і незалежності України під час воєнного стану
та сприяння у вирішенні завдань, поставлених
перед Головним управлінням розвідки
Міністерства оборони України
та з нагоди Дня Збройних сил України

Начальник Головного управління розвідки
Міністерства оборони України

Генерал-лейтенант Кирило БУДАНОВ

Головне управління розвідки Міністерства оборони України
Код 43966883

06.12.2023

**ДОДАТОК Р**

Нагорода EEML 2023