

## ЗАСТОСУВАННЯ РЕКУРЕНТНИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ ПОКРАЩЕННЯ ПРОЦЕСУ ПЛАНУВАННЯ SCRUM-СПРИНТІВ

Олекса Гудь<sup>1</sup>, Наталія Кунанець<sup>2</sup>

<sup>1,2</sup> Національний університет “Львівська політехніка”,  
кафедра інформаційних систем та мереж, Львів, Україна

<sup>1</sup> E-mail: oleksa.o.hud@lpnu.ua, ORCID: 0009-0007-4298-7332

<sup>2</sup> E-mail: nataliia.e.kunanets@lpnu.ua, ORCID: 0000-0003-3007-2462

© Гудь О., Кунанець Н., 2024

У дослідженні обґрунтовано доцільність використання технології машинного навчання для вдосконалення процесу планування ітерацій в IT-проектах, що реалізуються з використанням методології Scrum. Постановлено проблему планування продуктивності в командах. Сформовано предмет і об'єкт дослідження. Описано очікувану наукову новизну та практичну значущість результатів дослідження. Розглянуто комплекс можливих проблем, пов'язаних з плануванням задач в IT-проектах, зокрема точність прогнозування продуктивності команд. Виявлено ключові фактори, що впливають на процес планування, та проведено аналіз можливих шляхів їхнього вирішення. Проаналізовано успішність застосування технологій машинного навчання у сфері проєктного менеджменту. Проведено оцінку технологій машинного навчання для прогнозування реалізації задач ітерацій Scrum-проектів. Зосереджено увагу на використанні у цих процесах рекурентних нейронних мереж. Обрано модель Довга Короткочасна Пам'ять (ДКЧП) для прогнозування продуктивності команд IT-проектів. Сформовано мету, цілі та завдання дослідження. Зібрано та проаналізовано історичні дані продуктивності проєкту. Проведено аналіз ефективності роботи розробленої моделі залежно від заданих параметрів та формату вхідних даних. Запропоновано три варіанти моделі з різною кількістю повних запусків процесу навчання алгоритму. Нормалізовано дані для оптимізації моделі. Встановлено, що моделі довгої короткочасної пам'яті здатні точно прогнозувати майбутні показники на основі нормалізованих історичних даних про попередні спринти. Проаналізовано результати прогнозування. Визначено способи подальшого вдосконалення моделі. Доведено доцільність використання підходу рекурентних нейронних мереж у процесі планування спринтів. Запропоновано методи використання рекурентних нейронних мереж для планування задач IT-проекту. Визначено обмеження такого підходу. Запропоновано альтернативний варіант застосування рекурентних нейронних мереж у разі невідповідності умовам обмежень. Наведено перспективи подальших досліджень. Зроблено висновки про процес та результати проведеного дослідження.

**Ключові слова:** прогнозування; машинне навчання; LSTM; Scrum; рекурентні нейронні мережі.

### Вступ

Управління проєктами в умовах сучасного програмного забезпечення часто здійснюється за допомогою гнучких методологій, як-от Scrum. Однією з ключових складових цього підходу є планування спринтів, яке забезпечує команду чітким баченням цілей і завдань на короткий термін.

Проте у процесі планування спринтів можуть виникати труднощі, пов'язані з точною оцінкою робочих завдань, розподілом ресурсів та адаптацією до змін у вимогах.

Застосування інтелектуальних технологій, як-от рекурентні нейронні мережі (RNN), може значно покращити процес планування Scrum-спринтів. Завдяки здатності цих мереж до опрацювання часових рядів і прогнозування, вони можуть надати більш точні прогнози щодо виконання завдань та сприяти оптимізації розподілу ресурсів.

Актуальність цієї статті полягає у дослідженні шляхів покращення процесу планування спринтів проєктів, що реалізуються за Scrum-методологією із застосуванням рекурентних нейронних мереж. Такий підхід забезпечить автоматизацію процесів аналізу даних, прогнозування та прийняття виважених рішень, що сприятиме підвищенню ефективності та гнучкості роботи команди розробників проєкту.

### **Формулювання проблеми**

Ефективне планування спринтів є ключовим аспектом управління проєктами в методології Scrum. Однак часто виникають труднощі з точним прогнозуванням продуктивності команди та розподілом навантаження. Традиційні підходи, як-от оцінка через "Story Points", не завжди дають змогу врахувати всі можливі змінні та динаміку процесів, що ускладнює ефективне планування.

Машинне навчання, особливо рекурентні нейронні мережі (LSTM), має великий потенціал для покращення прогнозування в Scrum-середовищах. Завдяки здатності моделювати послідовні дані, LSTM може допомогти краще передбачати результати спринтів, що дасть змогу оптимізувати управління ресурсами й підвищити ефективність команди.

Основна проблема дослідження полягає в тому, чи можуть моделі LSTM забезпечити більш точне прогнозування ітерацій порівняно з традиційними методами, що використовуються в Scrum.

Об'єктом дослідження є процес планування спринтів у межах Scrum-методології управління проєктами.

Предметом дослідження є метод прогнозування на основі рекурентних нейронних мереж для покращення процесу планування Scrum-спринтів через прогнозування виконання завдань, оптимізації розподілу ресурсів та підвищення точності оцінки результатів.

Науковою новизною результатів дослідження є розроблення нового підходу до оптимізації процесу планування ітерацій в межах методології Scrum, а також емпірично підтверджена доцільність застосування рекурентних нейронних мереж для покращення управлінських процесів у проєктах, що сприятиме підвищенню точності прогнозування та ефективності командної роботи.

Практична значущість результатів дослідження – впровадження запропонованого методу прогнозування продуктивності команд у процесах планування ітерацій у методології Scrum. У роботі потрібні наведені необхідні умови та інструменти для успішного застосування вказаного підходу, що може дозволити проєктним менеджерам ефективніше розподіляти ресурси та коригувати плани.

### **Аналіз останніх досліджень та публікацій**

Продуктивність роботи команди безпосередньо впливає на успішність проєктів, що й зумовлює актуальність досліджень на цю тему та стимулює розвиток методології Scrum. Метою таких досліджень нерідко стає аналіз та оцінка впливу різних факторів на результат ітерацій. Так, у дослідженні [1] було виявлено два основних типи факторів, що впливають на швидкість виконання завдань у Scrum-спринтах, – з непрямым (INDIRECT) та прямим (DIRECT) впливом на продуктивність команди. До факторів із непрямым впливом належить великий обсяг незавершених робіт, прихована складність задач, поява дрібних проблем, нечітко визначені запити на функціонал, невраховані зусилля в процесах розроблення програмних продуктів та неочікуване переривання

роботи. Фактори з прямим впливом передбачають динаміку змін складу команди та необхідність переробки певних обсягів роботи, інтенсивні періоди тестування, а також потреби в навчанні учасників команди. Порівняння результатів огляду публікацій та кейс-дослідження показало, що більшість цих факторів мають вагомий вплив на продуктивність, що підтверджує їхню значущість у контексті прогнозування швидкості роботи команд у Scrum-ітераціях. Однак не всі перераховані фактори можна використати для задач прогнозування, адже для цього вони повинні бути вимірювальними, чого складно досягти не змінюючи основних конфігурацій проєкту.

Фактор “невраховані зусилля в процесах розроблення програмних продуктів” також зазвичай входить у групу невимірювальних через свою абстрактність, проте ситуація змінюється, якщо розглядати його в контексті методології Agile. У дослідженні [2] відзначено, що планування спринту та ретроспективи за методологією Agile може значно вплинути на швидкість виконання завдань та передбачуваність роботи команди проєкту. Зокрема, вони допомагають ідентифікувати та частково виміряти “невраховані зусилля в процесах розроблення програмних продуктів” через регулярне оновлення пріоритетів і комунікацію в межах команди. Це дослідження підтверджує, що належне використання процедур Agile-методології може покращити вимірюваність таких абстрактних факторів, що сприяє їхньому прогнозуванню.

У роботах [3] і [4] зосереджено увагу на вплив м’яких навичок розробників на продуктивність команд у проєктах, які реалізуються за Scrum-методологією. У першій роботі особливу увагу приділено важливості формування метрик оцінки ефективності роботи команд, зокрема таких аспектів, як-от: комунікація, адаптивність та швидкість прийняття рішень, які безпосередньо впливають на прогрес у спринтах проєкту. У другій роботі досліджуються навички членів команди та рівень їхньої компетентності, що впливає на здатність команд досягати поставлених цілей. Обидві роботи вказують на те, що людський фактор може стати ключовим у визначенні продуктивності команд, враховуючи не тільки технічні показники, але й м’які навички, як-от здатність до навчання та командна співпраця.

Дослідження [5] доповнює ці висновки, зосереджуючись на тому, як склад команди та її розмір впливають на її загальну продуктивність у спринтах. У роботі зазначено, що різноманітність навичок і баланс між досвідом та новими членами команди можуть відігравати ключову роль у досягненні стабільної продуктивності. Виявлено, що команди з чітко розподіленими ролями та добре налагодженою комунікацією значно швидше адаптуються до змін та ефективніше працюють під час спринтів. Цей фактор також є важливим у прогнозуванні продуктивності, оскільки вивчення складу команди може дати змогу передбачити в процесі планування потенційні ризики або успіхи.

Технології машинного навчання досліджуються в контексті прогнозування, завдяки здатності виявляти приховані зв’язки між компонентами вхідних даних та можливості комплексного аналізу. Як зазначено в роботі [6], якісна підготовка даних значно покращує результати роботи моделей. У дослідженні [7] розглянуто важливість вибору моделі та методу її оцінки, що підтверджено результатами порівняння методів [8].

Серед методів, застосованих у більшості досліджень в контексті прогнозування, часто фігурують методи екстремального градієнтного підсилення (XGboost), що діє на основі дерев рішень, та метод ДКЧП – довгої-короткочасної пам’яті (LSTM – Long Short Term Memory). У дослідженні [9] було показано, що метод XGBoost здатен знаходити складні залежності між змінними та генерувати точні прогнози. У роботі [10] підтверджується доцільність використання ДКЧП для часових рядів через здатність зберігати інформацію з попередніх даних.

Наразі існує дуже обмежена кількість досліджень, які б безпосередньо пов’язували модель Long Short-Term Memory (LSTM) з методологією Scrum. Однак різні наукові роботи демонструють ефективність цієї моделі машинного навчання в інших сферах, як-от прогнозування ринку акцій та аналіз часових рядів.

Наприклад, у дослідженні [11] було продемонстровано застосування ДКЧП для прогнозування цін на акції на фінансових ринках. Автори показали, що ДКЧП, на відміну від інших моделей, може захоплювати довготривалі залежності в даних, що робить модель ефективним інструментом для передбачення майбутніх значень активів. У цьому дослідженні було побудовано модель для прогнозування цін на акції компаній Google і Nike на основі відкритих даних про їхні щоденні зміни на Нью-Йоркській фондовій біржі. Модель демонструвала велику точність у відслідковуванні динаміки акцій та показала кращі результати порівняно з іншими підходами, зокрема такими, як ARIMA.

У дослідженні [12] автори порівнюють оптимізовану ДКЧП-модель з іншими методами машинного навчання для прогнозування необхідних зусиль на розробку програмного забезпечення. Використовуючи кілька наборів даних і метрик для оцінки точності прогнозів, доведено, що ДКЧП демонструє вищу точність, ніж такі методи, як K- найближчих сусідів, дерев рішень та випадкового лісу.

Ці результати можуть бути використані як основа для припущення, що ДКЧП може бути застосована для аналізу продуктивності команд у Scrum-середовищах для прогнозування різних аспектів планування спринтів на основі історичних даних.

### Формулювання цілі статті

Мета роботи – підтвердити доцільність застосування методів прогнозування на основі рекурентних нейронних мереж з метою підвищення продуктивності під час планування Scrum спринтів.

Для досягнення зазначеної мети визначено такі основні завдання дослідження:

1. Проаналізувати останні публікації на теми, що стосуються цього дослідження, як-от: “Прогнозування в плануванні Scrum-спринтів”, “Рекурентні нейронні мережі”, “Моделі машинного навчання для прогнозування”, “Машинне навчання в методології Scrum”.
2. Визначити та обґрунтувати вибір моделі машинного навчання, яка буде використовуватися для оцінки результатів прогнозування в процесі планування Scrum-спринтів.
3. Провести експеримент з обраною моделлю машинного навчання на реальних або симуляційних даних та визначити точність прогнозування.
4. Порівняти прогнозовані значення, отримані за допомогою моделі, з реальними результатами Scrum-спринтів, використовуючи показники точності.
5. Оцінити перспективу використання запропонованої моделі для планування Scrum-спринтів, зробити висновок про доцільність застосування конкретної моделі машинного навчання для прогнозування в даному контексті.

### Виклад основного матеріалу

Реалізація проектів із створення програмних продуктів за методологією “Скрам” (англ. Scrum) набуває дедалі більшої популярності в сучасній галузі інформаційних технологій (ІТ). Цей фреймворк для управління ІТ-проектами, який належить до гнучких методологій (Agile), дає змогу командам ефективно працювати над проектами завдяки ітеративному підходу. Цей підхід дає можливість швидко реагувати на зміни вимог і випускати продукт невеликими, але частими інкрементами. Суть інкременту доволі проста: у команди розробників програмного продукту є певний відрізок часу “Спринт” (англ. Sprint) та вичерпна кількість завдань, яку потрібно виконати протягом спринта. Для того, щоб правильно визначити завдання, обрати пріоритети, окреслити обсяг робіт, а також забезпечити взаємодію та розподілити відповідальності в команді, необхідно оцінити доступні людські ресурси та розподілити завдання так, щоб забезпечити максимально ефективне виконання завдань. В основі цього алгоритму лежить низка факторів та невизначеностей. Так, недоступ-

ність учасників команди, поява незапланованої роботи можуть впливати на продуктивність роботи команди з розроблення програмного продукту, що негативно позначається на показниках успіху проєкту. Нехтуючи такими чинниками при плануванні, менеджер проєкту безумовно піддається ризику отримати небажаний результат ітерації, що надалі може знизити якість виконання проєкту загалом у вигляді наростаючого технічного боргу. Однак через складність прогнозування таких чинників та порівняно малий відрізок часу ітерації, відповідальність за оцінку зусиль та терміну, які необхідні для виконання одного або комплексу завдань, перекладається на інтуїтивні здогадки та практичний досвід учасників команди з розроблення програмного продукту. Такий підхід надає можливість удосконалення процедури планування шляхом використання фактичних та аналітичних даних, а дані дають змогу вдосконалити процес планування ітерацій шляхом врахування результатів прогнозування.

Завдяки високому попиту на опрацювання великих обсягів даних технології машинного навчання стрімко розвиваються, зокрема методи прогнозування. Такі методи виходять за межі простої екстраполяції та мають можливість враховувати динаміку, аномалії, зовнішні фактори та отримувати результат. На сьогодні методами машинного навчання створюють прогнози для широкого спектру явищ і подій залежно від доступних даних та обраних моделей: фінансові, маркетингові, ресурсні, медичні, логістичні тощо. Існує понад десяток моделей машинного навчання, які здатні розв'язувати задачі прогнозування, проте зазвичай для цього використовують один із двох таких типів – моделі на основі деревах рішень та нейронні мережі. Звісно, що інші моделі, як-от ансамблеві чи баєсові, також використовуються для прогнозування, проте вибір саме таких моделей переважно зумовлений незвичною специфікою середовища або факторів об'єкта прогнозування. Така варіативність впливає із переваг та недоліків кожної моделі, наприклад, модель градієнтно-підсиленої машини на основі дерев рішень має схильність до перенавчання, нейронні мережі потребують послідовності даних для створення якісних прогнозів, а баєс припускає незалежність між ознаками (факторами), що не завжди відповідає реальності в складних проєктах.

Зважаючи на ці виклики, сучасні дослідження акцентують увагу на важливості впровадження інтелектуальних систем для покращення планування та управління проєктами. Машинне навчання стає невід'ємною частиною цього процесу, забезпечуючи можливість автоматичного аналізу великих обсягів даних та створення більш точних прогнозів, що враховують різноманітні фактори впливу. Це своєю чергою дає змогу зменшити невизначеність при плануванні спринтів у методології Scrum, підвищуючи точність оцінок та ефективність роботи команди. Однак існує проблема вибору оптимальної моделі машинного навчання, яка б найкраще відповідала специфіці конкретного проєкту. Оцінка доцільності використання моделі на основі нейронних мереж є актуальним питанням, яке вимагає дослідження та аналізу, що забезпечить ефективність та успішність проєкту в умовах постійно змінюваних вимог і середовища.

Дослідження проводилося в таких умовах: команда здійснює підтримку віртуалізованої платформи, яка охоплює два ключові напрями діяльності – розроблення автоматизації (Dev) та обслуговування і ремонт апаратного забезпечення (Ops). Незважаючи на важливість обох напрямів, методологія Scrum застосовується виключно для Dev-процесів, у межах яких інженери займаються автоматизацією. До того ж Ops-завдання часто виникають непередбачено, що призводить до відволікання інженерів від основних Dev-завдань, негативно впливаючи на їхню продуктивність у межах спринтів.

Для оцінювання складності завдання методологія Scrum передбачає використання так званих "сторі поінтів" (англ. Story Points). Story Points ґрунтуються на оцінці розміру та складності завдань, а не їхній тривалості. Вони не виражаються у фіксованих одиницях виміру, як-от години чи дні. Однією з ключових характеристик Story Points є їхня гнучкість і здатність адаптуватися відповідно до зміни обставин, що відрізняє їх від традиційних часових одиниць. Це дає змогу команді більш

точно оцінювати зусилля, не обмежуючись лінійним виміром часу. Кількість витрачених зусиль було розглянуто як основний показник продуктивності команди. Цей самий показник було взято за об'єкт прогнозування, як і дані, використані в дослідженні, – з реального проекту і помножено на сталий коефіцієнт у форматі кількості витрачених зусиль за останні чотирнадцять спринтів. Кожен спринт становив стандартний термін ітерації, а саме: два тижні або десять робочих днів.

Під час дослідження виконувалася трансформація даних з метою покращення процесу навчання моделі, проводилася оцінка оптимальної кількості повних запусків навчального процесу в алгоритмі (епох), а також врахування ресурсних витрат на запуск рішення.

Для початку побудовано модель з урахуванням лише історичних даних, погрупованих у показники продуктивності спринтів.

Відповідні дані наведено в табл. 1.

Таблиця 1

### Історичні дані продуктивності

Спринт	Кількість витрачених зусиль (Story Points)
1	104
2	266
3	32
4	96
5	56
6	96
7	48
8	126
9	20
10	174
11	228
12	94
13	170
14	204

У табл. 1 наведено повні дані, що використовувалися на момент початку дослідження.

Для прогнозування цих показників застосовувалася модель довгої короткочасної пам'яті (ДКЧП) як методу для прогнозування ключових показників продуктивності команди в Scrum-плануванні. ДКЧП є різновидом рекурентної нейронної мережі, що має здатність враховувати довготривалі залежності в даних завдяки використанню "пам'яті", яка дає змогу моделі зберігати інформацію про попередні стани та використовувати її для покращення прогнозування.

У дослідженні було застосовано стандартну архітектуру ДКЧП, що залучає кілька шарів: вхідний шар для прийому початкових даних, приховані шари з ДКЧП-комірками для опрацювання послідовностей даних та вихідний шар для генерації прогнозованих значень. Ця модель здатна ефективно розпізнавати патерни в часових рядах та враховувати попередні зміни, що дає змогу отримувати більш точні прогнози щодо продуктивності роботи команди під час спринтів.

Для тренування моделі використовувалася наведена вибірка історичних даних про кількість виконаних задач в попередніх спринтах. Модель навчалася на цих даних із застосуванням алгоритму оптимізації даних, що давало змогу зменшити похибку прогнозування. Для оцінки якості прогнозів використовувалися метрика, як-от кореневе середнє квадратичне відхилення (RMSE), що дозволило визначити, наскільки точно модель може передбачити майбутні показники.

Математично модель ДКЧП описується набором рівнянь для кожної комірки мережі, який управляє станом пам'яті та виходом для кожної ітерації в часі. Основними компонентами ДКЧП є:

1. Стан комірки (cell state,  $C_t$ ):

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \quad (1)$$

де  $C_t$  – стан комірки;  $f_t$  – блок забуття;  $\odot$  – поелементне множення, яке застосовується для комбінування інформації з блоків,  $i_t$  – вхідний блок (англ. input gate) та  $\tilde{C}_t$  – кандидат на новий стан комірки, який представляє нові дані, що потенційно можуть бути додані до стану комірки.

Стан комірки – це “лінійна” частина, яка допомагає передавати інформацію з одного кроку до наступного з мінімальними змінами. Стан комірки модифікується за допомогою сигмоїдних функцій.

2. Блок забуття (forget gate,  $f_t$ ):

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f), \quad (2)$$

де  $\sigma$  — сигмоїдна функція активації,  $x_t$  – вхід на кроці  $t$ ,  $h_{t-1}$  – вихід на попередньому кроці,  $W$  – вагова матриця для кожного блоку,  $b$  – вектор зсуву.

Блок забуття визначає, яка частина інформації з попереднього стану повинна бути “забута”. Це робиться через сигмоїдну функцію  $\sigma$ , яка визначає від 0 (забути) до 1 (зберегти).

3. Вхідний блок (input gate,  $i_t$ ):

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i), \quad (3)$$

визначає, яка нова інформація буде додана до стану комірки.

4. Кандидат на нове значення стану комірки (candidate cell state,  $\tilde{C}_t$ ):

$$\tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C), \quad (4)$$

де  $\tanh$  – гіперболічна тангенс-функції, а формула генерує нові значення кандидатів для додавання до стану комірки і обмежується гіперболічною тангенс функцією  $\tanh$ , яка призводить до значень у діапазоні від -1 до 1.

5. Вихідний блок (output gate,  $o_t$ ):

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o), \quad (5)$$

визначає, яку частину стану комірки буде використано для вихідного значення на поточному кроці.

6. Вихід  $h_t$ :

$$h_t = o_t \odot \tanh(C_t). \quad (6)$$

Вихід на поточному кроці визначається як комбінація вхідного сигналу та модифікованого стану комірки через гіперболічну тангенс-функцію.

На кожному кроці модель вирішує, яку частину минулої інформації зберегти (forget gate), яку нову інформацію додати (input gate), і яку частину використовувати для обчислення виходу (output gate). Це дає змогу ДКЧП зберігати важливу інформацію на довгих часових інтервалах і робити точні прогнози на основі довгих послідовностей. Візуальне представлення даного процесу представлено на рис. 1.

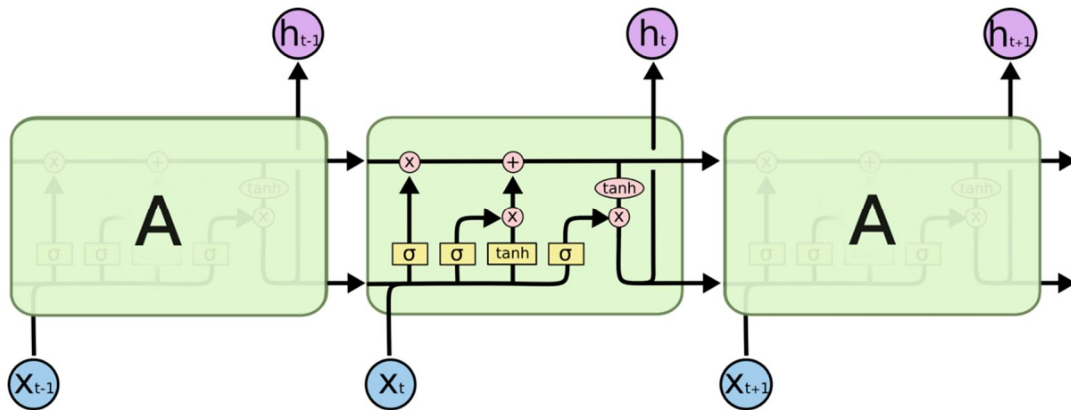


Рис. 1. Діаграма алгоритму ДКЧП

На рис. 1 кожна лінія представляє цілий вектор, що передається від виходу одного вузла до входів інших. Кола позначають поелементні операції, як-от додавання векторів, натомість прямокутники символізують шари нейронної мережі, що навчаються. Лінії, які об'єднуються, вказують на конкатенацію даних, а розгалужені лінії позначають процес копіювання вмісту для подальшого використання в різних місцях.

Для реалізації вказаного методу було використано мову програмування Python 3 із допомогою комплексу пакетів для машинного навчання. Повний перелік пакетів зображено на рис. 2 у вигляді зразка використаного програмного коду.

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from pandas import read_csv
4 import math
5 import tensorflow as tf
6 from tensorflow.keras.models import Sequential
7 from tensorflow.keras.layers import Dense
8 from tensorflow.keras.layers import LSTM
9 from sklearn.preprocessing import MinMaxScaler
10 from sklearn.metrics import mean_squared_error

```

Рис. 2. Список використаних пакетів під час дослідження

Вхідні дані зібрано та збережено у файлі CSV, та опрацьовано інтерпретатором. Дані було розділено на два набори – тестовий та тренувальний.

Тренувальний набір використовується для навчання моделі ДКЧП. Він містить послідовності вхідних даних та їхні відповідні вихідні значення. Модель ДКЧП оптимізує свої ваги на основі тренувальних даних за допомогою зворотного поширення помилки та оновлення параметрів. Завдяки своїй архітектурі ДКЧП може навчатися як короткотривалим, так і довготривалим залежностям у часі, використовуючи цей набір даних.

Тестовий набір використовується для оцінки точності моделі після її навчання. Він містить дані, які не використовувалися під час навчання, і дає змогу визначити, наскільки добре модель узагальнює результати на нових, невідомих даних. Після завершення навчання модель тестується на цьому наборі. Її продуктивність вимірювалася за допомогою метрики, як-от кореневе середньоквадратичне відхилення (RMSE).



У дослідженні масивом даних, що становить тренувальний набір, було обрано дві третини всіх даних, або 67 %. З точки зору загального масиву даних це показники спринтів від номера “1” до номера “10”.

Створюється порожня модель типу “Послідовна”. Це означає, що шари моделі додаються послідовно, один за другим. У цьому разі модель має декілька шарів: ДКЧП і “щільний”.

Шар ДКЧП складається з чотирьох нейронів, на кожен крок подається один параметр, а кількість попередніх кроків використовуються для прогнозу – 1.

До моделі додається щільний шар з одним нейроном на виході. Це означає, що модель виводитиме лише одне значення на основі вхідних даних (наприклад, прогноз для наступного кроку).

Для компіляції моделі застосовано низку параметрів. Встановлено функцію втрат, яку використовують для оптимізації моделі. В такому разі це середньоквадратична помилка (MSE), яка є популярною для завдань регресії. Також оптимізатор Adam, який використовується для навчання моделі, автоматично налаштовує коефіцієнти навчання для кращої ефективності.

Модель тренується на тренувальному наборі даних, і буде тренуватися 10 епох (повний перегляд набору даних 10 разів). Ця кількість епох буде мінятися протягом дослідження, щоб оцінити оптимальність моделі. Модель буде оцінювати масштаб навчання після кожного зразка.

Для аналізу продуктивності за 14 спринтів, незалежно від кількості епох, стає очевидним, що такого обсягу даних замало. Результати експерименту представлено у табл. 2.

Таблиця 2

**Результати експерименту з неопрацьованими вхідними даними**

Спроба	Епохи	Кількість часу (с)	RMSE тренувального набору	RMSE тестового набору
1	10	1,69	93,44	118,34
2	50	2,97	75,09	87,9
3	100	3,82	70,11	86,71

Модель вважають успішною і придатною для прогнозування у разі, якщо значення її тестового RMSE становить до 20 % від середнього значення використаних даних. В цій ітерації дослідження середнє значення використаних даних дорівнює 122,4. Найкращий результат RMSE тестового набору – 86,71, що становить 70,8 % від середнього значення. Такі показники кореневого середньоквадратичного відхилення знаходяться поза межами норми і свідчать про непридатність моделі для подальшого прогнозування. Враховуючи динаміку зміни RMSE, незалежно від зміни кількості епох модель прогнозування в такій її конфігурації не задовольняє базові потреби дослідження. Це підтверджує візуальне представлення спроби № 3, зображене на рис. 3.

Синя лінія на графіку зображає фактичні значення спостереження, оранжева позначає результати тренувального етапу, а зелена – тестового. Не витримана геометрія, розриви на відрізках вказують на непридатність даних до прогнозування.

Причиною для такої невідповідності результату може бути замалий обсяг даних. Проте доволі велика частина IT-проектів тривають до шести місяців, і просте збільшення обсягу даних, залучаючи до нього показники попередніх спринтів, пропорційно зменшуватиме практичну значущість моделі. Для цього дані було перетворено та обраховано кількість витрачених зусиль по днях. Зразок оновлених даних подано у вигляді табл. 3.

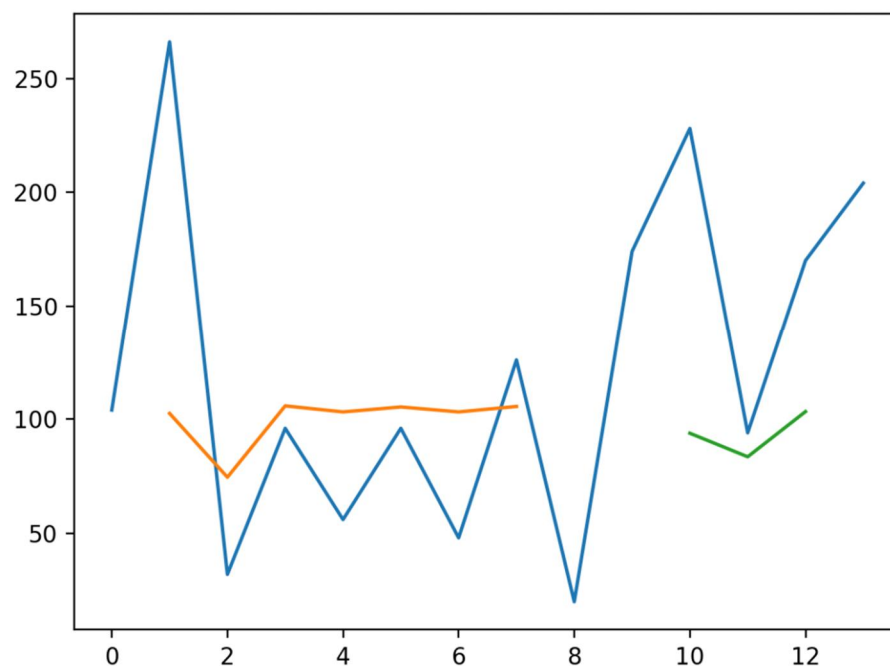


Рис. 3. Результат спроби № 3 першої ітерації експерименту

Таблиця 3

**Зразок оновлених даних продуктивності**

День проєкту	Кількість витрачених зусиль (Story Points)
1	2
6	9
11	49
15	13
27	2
63	4
93	13

У табл. 3 дані наведено вибірково для демонстрації стилю та форматування даних. Повна вибірка даних налічує 140 записів.

Тренування моделі ДКЧП відбувалося з такими самими конфігураціями, як і в разі неопрацьованих даних. Модель будується з одним шаром LSTM, що має 4 нейрони, і одним щільним шаром на виході. Навчання відбувається за 100 епох із використанням оптимізатора Adam і функції втрат RMSE. Для такої вибірки даних середнє значення показників дорівнює 12,2. Це означає, що модель зможе вважатись успішною, якщо її RMSE тестового набору буде меншим, ніж 2,44. Результати другої ітерації дослідження представлено в табл. 4.

Таблиця 4

**Результати експерименту з поденними даними продуктивності**

Спроба	Епохи	Кількість часу (с)	RMSE тренувального набору	RMSE тестового набору
1	10	2,46	12,16	12,1
2	50	5,98	12,09	11,56
3	100	10,37	12,06	11,17

З результатів, наведених у табл. 4, можна зробити висновок, що модель витрачає більше ресурсного часу, щоб навчитися на більшому обсязі даних, але найменший RMSE рівний 11,17, що становить 91,6 % від середнього значення використаних даних. Це свідчить про те, що модель, в основі якої знаходяться денні дані, навіть менш придатна для використання, ніж модель, що використовує неопрацьовані дані.

Проте, незважаючи на це, візуальне представлення результатів демонструє позитивну динаміку в сприйнятті даних моделлю, водночас підтверджуючи її непридатність для подальших прогнозувань. Візуальне представлення найуспішнішої моделі вказаної ітерації дослідження наведено на рис. 4.

З рис. 4 непридатність моделі для подальшого використання в процесі прогнозування конкретних значень є очевидною. Проте, порівнюючи з моделлю, що використовує неопрацьовані дані, можна зробити висновок про позитивну динаміку сприйняття моделлю геометрії графіка, що особливо помітно на відрізках по осі абсцис тренувального етапу (помаранчева крива) “20”–“30” та “80”–“95”, а також тестового етапу (зелена крива) на відрізку “115”–“125”. У цих проміжках геометрія кривої згенерованих даних доволі точно відображає лінію спостережуваних даних, однак не відтворює фактичних значень.

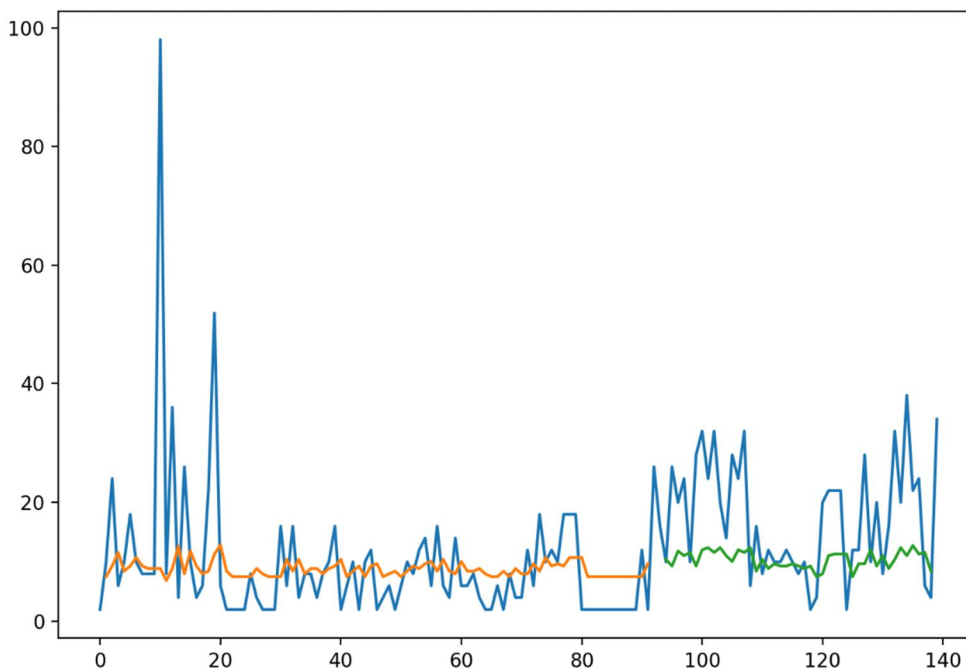


Рис. 4. Результат спроби № 3 другої ітерації експерименту

Провівши детальний огляд результату, зроблено висновок, що модель гірше навчається на даних, в яких присутня занадто різка динаміка, де попереднє значення може відрізнятись від наступного майже удев'ятеро. Тому було проведено нормалізацію даних задля “згладжування” кривої спостережуваних показників. Для того, щоб при нормалізації даних не втратити практичної значущості моделі, було прийнято рішення використовувати дані продуктивності за спринт, але розділити їх по кожному дню проекту у такому вигляді:

$$y_i = \sum_i^{i+9} x_i, \tag{7}$$

де  $i$  – день проекту,  $y_i$  – показник продуктивності за термін спринту в  $i$ -ий день,  $x$  – показник продуктивності за день. Отже, кожен показник відображає суму витрачених зусиль за наступні

десять днів (2 робочі тижні або один спринт). Масив даних продовжується, доки такий розрахунок можливий, тобто до останнього спостережуваного значення, а саме значення витрачених зусиль у спринті № 14. Приклад нормалізованих даних наведено у табл. 5.

Таблиця 5

## Зразок нормалізованих даних продуктивності

День проєкту	Кількість витрачених зусиль (Story Points)
1	104
6	224
11	266
15	132
27	68
63	52
93	216

У табл. 5 дані також наведено вибірково для демонстрації стилю та форматування даних. Повна вибірка даних налічує 131 запис. Така кількість зумовлена нехтуванням останніми 9-ма записами, які не можуть бути розрахованими формулою (7) через брак даних.

Умови тренування моделі ДКЧП не змінювалися та застосовувалися ті самі параметри, що і під час двох попередніх ітерацій. Для вказаної вибірки даних середнє значення показників дорівнює 119,5. Це означає що модель зможе вважатись успішною, якщо її RMSE тестового набору буде меншим, ніж 23,9. Результати третьої ітерації дослідження представлено в табл. 6.

Таблиця 6.

## Результати експерименту з нормалізованими даними продуктивності

Спроба	Епохи	Кількість часу (с)	RMSE тренувального набору	RMSE тестового набору
1	10	2,08	43,26	67,83
2	50	7,22	18,46	14,79
3	100	9,93	18,27	14,83

З результатів, наведених в табл. 6, можна констатувати успішність застосування моделі при спробах № 2 та № 3, адже RMSE тестового набору в цих спробах менш ніж 20 % і становить 12,38 % та 12,41 % відповідно.

Продовжуючи аналіз результатів експерименту, зазначеного в табл. 6, можна стверджувати, що оптимізація моделі ДКЧП досягається при кількості епох, близькій до 50. Це підтверджено тим фактом, що значення RMSE тренувального та тестового наборів при 50-ти епохах значно зменшується порівняно зі спробою № 1, в якій використовувалося 10 епох. Водночас подальше збільшення кількості епох до 100 (спроба № 3) не призвело до істотного покращення точності прогнозів, оскільки RMSE тестового набору майже не змінилося порівняно зі спробою № 2.

Отже, можна зробити висновок, що збільшення кількості епох понад 50 не призводить до суттєвого покращення якості моделі в контексті поточної вибірки даних. Це може свідчити про досягнення точки перенавчання, де модель починає запам'ятовувати тренувальні дані, але втрачає здатність до генерації нових тестових даних.

Успішність моделі чітко видно на рис. 5, де зображено візуальне представлення роботи моделі при 50-ти повних переглядах набору даних (спроба № 2), адже вона показала найкращий результат згідно з показником RMSE.

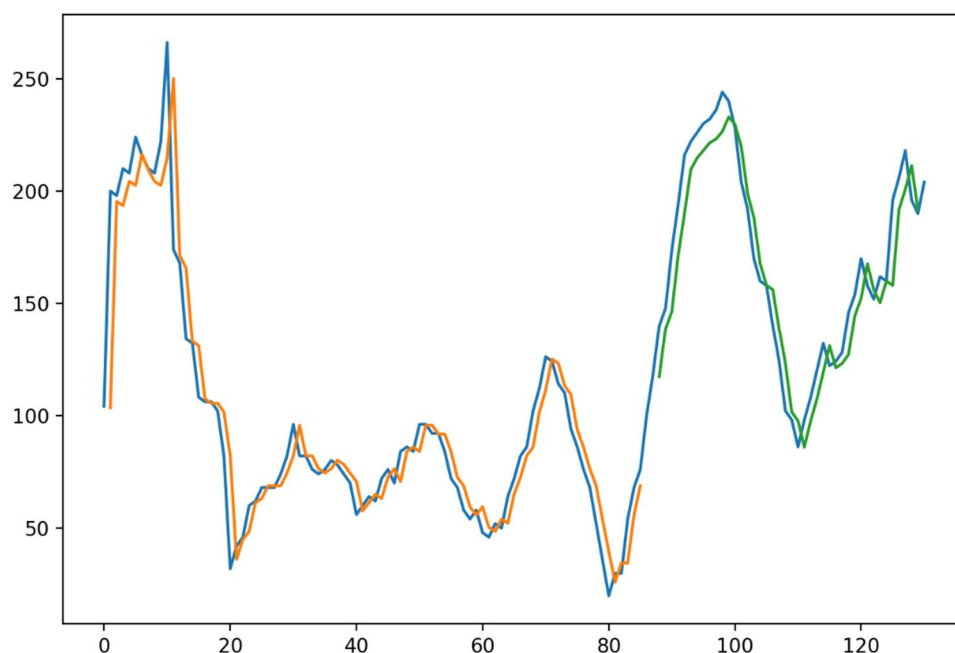


Рис 5. Результат спроби № 2 третьої ітерації експерименту

Згідно з графіком, крива прогнозованих значень (помаранчева лінія) досить точно слідує за кривою реальних даних (синя лінія). Це свідчить про те, що модель демонструє високий рівень точності за такої кількості епох. Здебільшого прогнозована крива збігається з реальною, що свідчить про мінімальні відхилення та відповідно нижчий RMSE для цієї спроби. Незначні відхилення можна спостерігати у певних ділянках, особливо в пікоподібних коливаннях. Однак такі розбіжності є типовими для таких моделей і можуть бути покращені за допомогою подальшої оптимізації параметрів.

Незначний розрив між лініями тренувальних та тестових прогнозованих значний на відрізку “82”–“90” може бути зумовлений недостатнім обсягом тренувальних даних, проте її наявність не впливає на оцінку придатності моделі для подальшого застосування.

Загалом графічна інтерпретація підкріплює висновок, що модель після 50-ти епох навчання досягла оптимальної точності прогнозування, оскільки її помилка мінімальна й вона добре узгоджується з реальними даними.

### Обговорення результатів дослідження

Проблематика процесу продуктивності ітерацій в проектному менеджменті давно досліджується і запропоновані рішення для цього варіюються залежно від об'єктів дослідження. Зазвичай у цій галузі науковці ретельніше досліджують соціологічну та менеджмент складові цього процесу, як, наприклад, у роботі [2] автори С. Шарма, Д. Кумар та М. Файяд зосереджуються на впливі Agile церемоній на загальну продуктивність команди створення програмних продуктів. Очевидно, що показник продуктивності команди залежить водночас від суттєвої кількості факторів, які зокрема були наведені у роботі [1], і ці фактори є дуже варіативними, та важко піддаються виявленню закономірностей чи залежностей. До таких відносяться і фактори м'яких навичок описані в

дослідженнях [3] і [4], які безумовно впливають на продуктивність команди, проте не піддаються прогнозуванню. Тож навіть при ретельному аналізі важливо враховувати людський фактор, який залишається складним для алгоритмічного прогнозування. Саме тому в сучасних дослідженнях усе більше уваги приділяється застосуванню моделей штучного інтелекту для автоматизації аналізу й прогнозування продуктивності команд, що допомагає заповнити прогалини у вивченні людського фактора.

Це дослідження не можна безпосередньо порівняти з попередніми роботами через унікальні особливості середовища, в якому воно проводилося. Однак використання машинного навчання як інструменту для підтримки процесу управління проектами вже було розглянуте в інших дослідженнях, наприклад, у роботі автора А. Йордан [12] чи роботі М. Магді, М. Забіл, А. Ахмада, Р. Ізмаїл, Ю. Юсоф, Л. Ченга, М. Азмі, Х. Натік та Х. Наїду [13], де підтверджено позитивний вплив використання технологій машинного навчання у сфері проектного менеджменту, особливо в контексті оцінки необхідних зусиль для виконання завдань.

Беручи до уваги успішне застосування технології саме рекурентних нейронних мереж у роботі [12], а також успішність її застосування в інших сферах [8], [10], [11], автори застосували метод довгої короткочасної пам'яті для прогнозування продуктивності команди проекту підтримки віртуального середовища.

Після проведення трьох ітерацій моделі на різних вибірках даних видно, що оптимізація моделі ДКЧП (довга короткочасна пам'ять) досягається при кількості епох, що дорівнює 50-м. Це підтверджується результатами другої спроби, де RMSE тренувального набору становило 18,46, а тестового – 14,79. Збільшення кількості епох до 100-а (третя спроба) не показало значного покращення результатів, оскільки RMSE для тестового набору лише незначно змінилося до 14,83, що підтверджує досягнення точки перенавчання моделі.

На основі аналізу графіку результатів для другої спроби можемо стверджувати, що прогнозована крива доволі точно повторює реальні дані, за винятком незначних розбіжностей у деяких ділянках. Ці розбіжності можуть бути зумовлені недостатнім обсягом тренувальних даних або незначними відхиленнями, які можуть виникати через нерегулярні зміни в історичних даних. Важливим моментом є те, що модель ДКЧП ефективно захоплює як короткострокові, так і довгострокові залежності, що робить її придатною для завдань прогнозування у контексті планування спринтів.

Результати дослідження підтверджують можливість використання моделі для прогнозування у контексті планування спринтів. Це підкреслює практичне значення для покращення процесу управління проектами, які реалізуються за методологією Scrum. Модель на основі рекурентних нейронних мереж, зокрема ДКЧП, дає змогу точніше прогнозувати продуктивність команди, що значно полегшує планування спринтів. Маючи прогноз наступної ітерації методології Scrum, відповідальні учасники цього процесу можуть орієнтуватися на тренд та коригувати планування відповідно до нього. Це дає можливість керівникам проектів краще розподіляти ресурси та завчасно виявляти потенційні проблеми в роботі команди, як-от недооцінка складності завдань або нестача ресурсів, у такий спосіб покращуючи процес планування і завдяки цьому збільшувати ймовірність успішного завершення проекту.

У практичному застосуванні ця модель може бути інтегрована у вже наявні системи управління проектами (наприклад, Jira або Trello) для автоматизованого прогнозування тривалості завдань і загальної продуктивності спринтів. Такий підхід допоможе зменшити кількість помилок, пов'язаних із людським фактором, та підвищити точність планування ітерації. Окрім того, можливе використання цієї моделі для адаптації процесів до змінних умов проектів, що особливо важливо у великих або довготривалих проектах, де врахування факторів сезонності та зміни складу команди має суттєве значення.

Отже, впровадження рішення на основі результатів цього дослідження дозволить підвищити ефективність та передбачуваність Scrum-процесів, що в кінцевому результаті сприятиме своєчасному та успішному завершенню проєктів.

Надалі метод довгої короткочасної пам'яті можна вдосконалювати за рахунок застосування більш складних моделей або використання додаткових даних для покращення точності прогнозування. Зокрема, можна інтегрувати інші моделі машинного навчання, як-от XGBoost або глибокі нейронні мережі, для створення гібридних систем, що дадуть змогу більш точно прогнозувати продуктивність команд на різних етапах проєкту. Також можливо покращити якість даних, що використовуються для навчання шляхом залучення додаткових історичних даних або реальних кейсів із виконання проєктів в аналогічних умовах.

Володіючи якісними даними, зазначений підхід може бути застосований у будь-яких проєктах, що працюють у межах методології Scrum. Проте варто зважати на обмеження, які притаманні моделям ДКЧП, як-от: необхідність достатньої кількості тренувальних даних, чутливість до гіперпараметрів, обмеженість в обробці великих обсягів даних, час навчання та можливість перенавчання. Якщо проєкт планує використовувати описаний підхід, проте не здатен забезпечити умов роботи алгоритму, рекомендовано розглянути альтернативні методи прогнозування на основі машинного навчання, зокрема XGBoost, який є більш придатним для менших наборів даних або ситуацій, де обчислювальні ресурси обмежені, оскільки цей алгоритм є менш вимогливим і зазвичай показує високу продуктивність на структурованих наборах даних.

### Висновки

У статті досліджено доцільність застосування рекурентних нейронних мереж, зокрема моделі ДКЧП, для покращення процесу планування Scrum-спринтів. Здійснено аналіз продуктивності команд IT-проєктів на основі історичних даних та проведено оцінку ефективності розробленої моделі.

Результати дослідження показали, що ДКЧП може точно прогнозувати майбутні показники продуктивності команд завдяки здатності моделі зберігати інформацію про попередні стани ітерацій. Отримуючи на вхід історичні дані, модель прогнозування також враховує низку факторів, притаманних тривалим проєктам, наприклад сезонність. Однак аналіз виявив деякі обмеження, пов'язані із перенавчанням моделі та недостатнім обсягом тренувальних даних.

Дослідження підтверджує, що за наявності якісних даних використання рекурентних нейронних мереж у плануванні Scrum-спринтів може покращити процес прогнозування зусиль та ресурсів команди. Цей підхід має потенціал для подальшої інтеграції в наявні системи управління проєктами для вдосконалення процесів планування, особливо в умовах динамічних змін вимог і середовища, що зрештою підвищить імовірність вчасного та успішного завершення проєктів.

Надалі рекомендовано збільшити обсяг даних для тренування моделей та розглянути можливість інтеграції інших моделей машинного навчання, як-от XGBoost, для створення гібридних систем прогнозування, що сприятиме підвищенню точності прогнозів. Також варто провести порівняльне дослідження впливу використання цього підходу до планування на продуктивність ітерацій у Scrum-проєктах, різних за цілями та конфігураціями.

### REFERENCES

1. Dong, W. (2015). *Exploring and Theorizing Velocity Flux in Agile Development*. <http://www.diva-portal.se/smash/get/diva2:819558/FULLTEXT01.pdf>
2. Sharma, S., Kumar, D., Fayad, M. (2021). *An Impact Assessment of Agile Ceremonies on Sprint Velocity Under Agile Software Development*. <https://doi.org/10.1109/icrito51393.2021.9596508>

3. Almeida, F., Carneiro, P. (2023b). Perceived Importance of Metrics for Agile Scrum Environments. *Information*, 14(6), 327. <https://doi.org/10.3390/info14060327>
4. Haputhanthrige, V., Asghar, I., Saleem, S., Shamim, S. (2024). The Impact of a Skill-Driven Model on Scrum Teams in Software Projects: A Catalyst for Digital Transformation. *Systems*, 12(5), 149. <https://doi.org/10.3390/systems12050149>
5. Almeida, F., Carneiro, P. (2023). Perceived Importance of Metrics for Agile Scrum Environments. *Information*, 14(6), 327. <https://doi.org/10.3390/info14060327>
6. Turner, A., Shieff, D., Dwivedi, A., Liarokapis, M. (2021). Comparing Machine Learning Methods and Feature Extraction Techniques for the EMG Based Decoding of Human Intention. *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. <https://doi.org/10.1109/embc46164.2021.9630998>
7. Oukhouya, H., Himdi, K. E. (2023). *Comparing Machine Learning Methods – SVR, XGBoost, LSTM, and MLP – For Forecasting the Moroccan Stock Market*. <https://doi.org/10.3390/iocma2023-14409>
8. Ali, P. J. M. (2022). Investigating the Impact of Min-Max Data Normalization on the Regression Performance of K-Nearest Neighbor with Different Similarity Measurements. *ARO-The Scientific Journal of Koya University*, 10(1), 85–91. <https://doi.org/10.14500/aro.10955>
9. Akande, Y. F., Idowu, J., Misra, A., Misra, S., Akande, O. N., Ahuja, R. (2022). Application of XGBoost Algorithm for Sales Forecasting Using Walmart Dataset. *Lecture notes in electrical engineering*, 147–159. [https://doi.org/10.1007/978-981-19-1111-8\\_13](https://doi.org/10.1007/978-981-19-1111-8_13)
10. Hewamalage, H., Bergmeir, C., Bandara, K. (2021). Recurrent Neural Networks for Time Series Forecasting: Current status and future directions. *International Journal of Forecasting*, 37(1), 388–427. <https://doi.org/10.1016/j.ijforecast.2020.06.008>
11. Moghar, A., Hamiche, M. (2020). Stock Market Prediction Using LSTM Recurrent Neural Network. *Procedia Computer Science*, 170, 1168–1173. <https://doi.org/10.1016/j.procs.2020.03.049>
12. Iordan, A. E. (2024). An Optimized LSTM Neural Network for Accurate Estimation of Software Development Effort. *Mathematics*, 12(2), 200. <https://doi.org/10.3390/math12020200>
13. Mahdi, M. N., Zabil, M. H. M., Ahmad, A. R., Ismail, R., Yusoff, Y., Cheng, L. K., Azmi, M. S. B. M., Natiq, H., Naidu, H. H. (2021). Software Project Management Using Machine Learning Technique – A Review. *Applied Sciences*, 11(11), 5183. <https://doi.org/10.3390/app11115183>

## THE FEASIBILITY OF USING RECCURENT NEURAL NETWORKS AS A TOOL FOR IMPROVING THE SCRUM SPRINT PLANNING PROCESS

Oleksa Hud<sup>1</sup>, Nataliia Kunanets<sup>2</sup>

<sup>1,2</sup>Lviv Polytechnic National University,

Information Systems and Networks Department, Lviv, Ukraine,

<sup>1</sup>E-mail: oleksa.o.hud@lpnu.ua, ORCID: 0009-0007-4298-7332

<sup>2</sup>E-mail: nataliia.e.kunanets@lpnu.ua, ORCID: 0000-0003-3007-2462

© Hud O., Kunanets N., 2024

The study substantiates the feasibility of using machine learning technology to improve the iteration planning process in IT projects implemented using the Scrum methodology. The problem of productivity planning in teams is set. The subject and object of the research are formulated. The expected scientific novelty and practical significance of the research results are described. A range of potential issues related to task planning in IT projects, particularly the accuracy of team productivity forecasting, is considered. Key factors influencing the planning process are identified, and possible solutions are analyzed. The success of applying machine learning technologies in project management is analyzed. An evaluation of machine learning technologies for forecasting the implementation of tasks in



Scrum project iterations is conducted. The focus is on the use of recurrent neural networks in these processes. The Long Short-Term Memory (LSTM) model is selected for predicting the productivity of IT project teams. The goals, objectives, and tasks of the research are formulated. Historical project performance data is collected and analyzed. The performance of the developed model is analyzed depending on the specified parameters and input data format. Three model variants with different numbers of complete algorithm training cycles are proposed. The data is normalized to optimize the model. It is established that Long Short-Term Memory models can accurately predict future performance based on normalized historical data from previous sprints. The prediction results are analyzed. Ways to further improve the model are identified. The feasibility of using the recurrent neural network approach in sprint planning is proven. Methods of using recurrent neural networks for IT project task planning are proposed. The limitations of this approach are identified. An alternative option for applying recurrent neural networks in case of non-compliance with the limitations is proposed. Prospects for further research are outlined. Conclusions were drawn regarding the course and results of the conducted research.

**Keywords:** forecasting, machine learning, LSTM, Scrum, recurrent neural network.