

УДК 004.6, 004.8, 004.942, 004.021: 519.6

DOI [https://doi.org/10.24144/2616-7700.2024.45\(2\).216-222](https://doi.org/10.24144/2616-7700.2024.45(2).216-222)

Н. О. Михайлов

Київський національний університет ім. Т. Шевченка,
аспірант кафедри теорії та технології програмування
nikmikhailov13@gmail.com

ORCID: <https://orcid.org/0009-0002-9374-3403>

АДАПТИВНА МОДЕЛЬ ПЛАНУВАННЯ ПРОЄКТІВ ТА ОЦІНКИ РИЗИКІВ ІЗ ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ

Ефективне управління проектами в умовах постійно змінюваних ринкових вимог та технологічного прогресу є необхідною складовою успішної діяльності компаній. Оптимізація процесу планування та управління ризиками відіграє ключову роль у підвищенні якості результатів і скороченні термінів реалізації. Традиційні методи планування, такі як метод критичного шляху і метод програмування в мережі, ефективно допомагають структурувати задачі проекту. Однак, через зростаючу складність проєктів, ці методи не завжди можуть врахувати динаміку змін та непередбачені ризики.

Застосування сучасних технологій, зокрема машинного навчання, дозволяє розробляти гнучкі та адаптивні системи управління, здатні враховувати як історичні дані, так і нові зміни в проєкті в режимі реального часу. Алгоритми машинного навчання можуть ефективно прогнозувати строки виконання завдань, оцінювати ризики та допомагати в оптимальному розподілі ресурсів. Дана стаття зосереджена на розробці адаптивної моделі, що дозволяє покращити точність планування та мінімізувати ризики під час виконання проєктів.

Ключові слова: машинне навчання, оцінка ризиків, нейронні мережі, управління проєктами, обробка даних, методи оптимізації, обчислювальні методи, інтелектуальні системи.

1. Вступ. Ефективне управління проектами в умовах швидких змін є критично важливим для досягнення успіху в будь-якій організації. Сучасний бізнес вимагає високого рівня точності у плануванні та прогнозуванні ризиків, щоб забезпечити стабільність виконання завдань і мінімізувати можливі збої. Традиційні методи планування, досі залишаються важливими, проте вони мають певні обмеження в умовах постійних змін і непередбачуваних факторів.

Впровадження алгоритмів машинного навчання дозволяє підвищити адаптивність систем управління проектами, використовуючи великі обсяги історичних даних для прийняття рішень. Такі технології допомагають системам автоматично коригувати плани відповідно до нових викликів і зменшують ймовірність виникнення помилок. Використовуючи ці підходи, організації можуть краще керувати ресурсами, прогнозувати терміни виконання та оцінювати ризики у режимі реального часу, що особливо актуально в умовах сучасних ринкових умов.

У статті запропоновано адаптивну модель управління проектами, яка заснована на використанні машинного навчання. Модель спрямована на покращення процесу планування, а також на більш ефективне управління ризиками, що виникають під час реалізації складних проєктів.

2. Аналіз показників планування. Показники планування проєктів є важливим інструментом для оцінки та оптимізації виконання завдань. Одним

із ключових показників є тривалість завдань, що визначає загальну тривалість проєкту. Для ефективного управління досить важливо оцінити наявні ресурси, включаючи їх доступність та ефективність використання. Пріоритетність завдань дозволяє концентрувати зусилля на особливо важливих етапах, забезпечуючи оптимальний порядок їх виконання. Використання буфера часу допомагає запобігти затримкам через непередбачувані обставини. Оцінка ефективності виконання завдань у порівнянні з початковими планами дозволяє вчасно визначати відхилення та коригувати процес. Також варто звернути увагу на рівномірний розподіл навантаження між членами команди, задля уникнення перевантаження і забезпечення продуктивності на кожному етапі проєкту [1].

Такий підхід дозволяє забезпечити комплексний аналіз планування і допомагає ефективно управляти як ресурсами, так і часом, що є вирішальним для успішного завершення проєкту.

Аналіз наведених показників планування дозволяє точніше визначити оптимальні стратегії управління проєктом, але традиційні методи не завжди забезпечують необхідну гнучкість і адаптивність. Для подолання цих обмежень пропонується розробка алгоритму на основі машинного навчання, який буде враховувати зазначені показники — тривалість завдань, ресурси, пріоритетність і взаємозалежність — і автоматично коригувати плани у відповідь на нові дані. Цей алгоритм дозволить підвищити точність прогнозування та мінімізувати ризики, пов'язані з динамікою проєкту [2].

3. Аналіз показників оцінки ризиків. Оцінка ризиків є важливою складовою управління проєктами, оскільки вона дозволяє ідентифікувати потенційні загрози та вчасно вжити заходів для їх мінімізації. Одним із основних показників оцінки ризиків є ймовірність виникнення події, яка може вплинути на успішність проєкту. Визначення рівня впливу таких подій допомагає оцінити, наскільки серйозно вони можуть зашкодити виконанню завдань. Інший важливий показник — це час реакції на ризик: чим швидше команда може реагувати на ризики, тим менше вони впливатимуть на проєкт. Крім того, важливо враховувати ступінь контролю над ризиком, оскільки деякі ризики є зовнішніми і не можуть бути повністю усунені.

Для проведення глибокого аналізу ризиків необхідно використовувати як якісні, так і кількісні методи оцінки. Це включає збір історичних даних про подібні проєкти та використання прогнозних моделей для оцінки потенційних загроз. Тут на допомогу приходять алгоритми машинного навчання, які здатні автоматично визначати взаємозв'язки між різними ризиковими факторами та оцінювати їхній вплив на проєкт. Застосування таких моделей дозволяє створювати системи, здатні оперативної і точно оцінювати ризики та вносити корективи в плани [3].

Результати оцінки ризиків допомагають приймати рішення щодо розподілу ресурсів для запобігання найбільш ймовірним і значущим загрозам. Це, в свою чергу, сприяє підвищенню надійності виконання проєкту і зниженню ймовірності серйозних відхилень від плану.

В рамках цієї роботи пропонується розробка системи, яка використовує машинне навчання для адаптивної оцінки ризиків. Завдяки алгоритмам машинного навчання, система зможе автоматично прогнозувати ризики на основі великих обсягів даних і миттєво оновлювати оцінки в реальному часі, забезпечуючи

ефективне управління ризиками на всіх етапах проекту [4].

4. Запропонований алгоритм. Запропонований алгоритм ґрунтується на глибокій нейронній мережі, яка здатна виконувати аналіз і обробку великих масивів даних проєктного планування та ризиків. Модель побудована так, що на вхідний шар подаються дані про тривалість та тип завдань, їх виконавців та затрачений час на виконання. Дані нормалізуються для подальшої обробки прихованими шарами.

Нейронна мережа включає кілька прихованих шарів, серед яких використовуються рекурентні нейронні мережі, оскільки вони ефективно працюють з часовими послідовностями даних, наприклад, змінами у стані проєкту протягом його виконання. Вхідні дані, що складаються з історичної інформації та нових умов, обробляються для виявлення закономірностей, які можуть вплинути на планування або оцінку ризиків [5].

Спеціалізований шар нейронної мережі відповідає за прогнозування можливих ризиків за допомогою алгоритму зворотного поширення помилок. На основі історичних даних модель генерує оцінку ймовірності виникнення ризику.

Ще одна ключова функція алгоритму — прогнозування тривалості. Шар, що відповідає за прогнозування термінів, використовує попередні дані та поточну доступність ресурсів для коригування часових рамок між завданнями.

На виході модель генерує два основні показники: прогнозовану тривалість виконання та рівень ризику. Ці результати інтегруються в систему планування, дозволяючи проєктним менеджерам приймати обґрунтовані рішення та швидко реагувати на зміни в умовах виконання проєкту [6].

5. Детальний опис алгоритму. Початковий етап передбачає підключення до JIRA за допомогою API для отримання даних про завдання зі спринтів. JIRA є популярним інструментом для управління проєктами, тому отримані дані відображають реальні робочі процеси команди. Після підключення через API, з JIRA можна отримати інформацію про попередні спринти. Ці дані включають імена виконавців завдань, тривалість виконання (скільки часу було витрачено на виконання кожного завдання), початкові оцінки часу та пріоритети завдань тощо. Підключення до JIRA дозволяє автоматизовано отримувати дані, що спрощує процес збору інформації для прогнозів, оскільки це робиться без ручного втручання [7].

Так як нейронні мережі працюють із числовими даними, категорійні дані, такі як імена виконавців і пріоритети завдань, перетворюються у числовий формат. Для цього використовуються спеціальні алгоритми перетворення, які змінюють текстові значення на числові коди. Наприклад, кожен виконавець отримує свій унікальний числовий код, що дозволяє моделі аналізувати їх вплив на тривалість виконання завдань.

Також, оскільки дані про час виконання завдань можуть мати різні діапазони (одні завдання можуть тривати години, а інші — дні), необхідно нормалізувати ці значення, щоб привести їх у стандартний формат, що є зручним для навчання моделі. Цього можна досягти за допомогою масштабування, яке перетворює всі значення в діапазон $[0, 1]$. Такий підхід дозволяє підвищити ефективність навчання моделі, оскільки вона працює з даними в одному масштабі [8].

Нейронна мережа складається з вхідного шару, кількох прихованих шарів та

двох вихідних шарів. Кожен шар мережі реалізує функцію, яка обчислюється на основі вагових коефіцієнтів та функції активації. Для прогнозування тривалості завдань використовується регресія, а для оцінки ризиків — класифікація [9].

Вхідний шар приймає набір характеристик завдання. Ці характеристики подаються у вигляді вектора, де кожен елемент відповідає певній характеристиці, такий як:

- виконавець (закодований як числове значення);
- пріоритет завдання;
- тип завдання (наприклад, «feature», «bug» тощо);
- початкова оцінка часу;
- витрачений час.

Припустимо, що вхідний вектор можна описати як:

$$X = [x_1, x_2, x_3, \dots, x_n],$$

де:

- $x_1, x_2, x_3, \dots, x_n$ — це числові значення кожної характеристики.

Модель містить два приховані шари, які використовують функцію активації Rectified Linear Unit (ReLU). Формально функція ReLU визначається як:

$$\text{ReLu}(z) = \max(0, z),$$

де ця функція вибрана тому, що вона дозволяє уникнути проблеми «зникаючого градієнта» та сприяє ефективному навчанню глибоких нейронних мереж.

Кожен нейрон у прихованому шарі обчислює лінійну комбінацію вхідних значень:

$$z = \sum_{i=1}^n w_i x_i + b,$$

де:

- w_i — вагові коефіцієнти;
- b — зміщення (bias).

Значення z передається через функцію активації ReLU, що дозволяє моделі враховувати лише позитивні значення для подальших обчислень.

Мережа має два виходи: один для прогнозування тривалості завдання, інший для оцінки ризиків. Оскільки задача прогнозування є регресійною (тривалість — це безперервне число), використовуються лінійні нейрони без активації:

$$\hat{y}_{duration} = w \times z + b,$$

де:

- $\hat{y}_{duration}$ — прогнозоване значення тривалості.

Для задачі оцінки ризиків використовується бінарна класифікація, тому вхідний шар використовує сигмоїдну функцію активації:

$$\sigma(z) = \frac{1}{1 + e^{-z}}.$$

Значення на виході в інтервалі $[0, 1]$ визначає ймовірність ризику виконання завдання [10].

Для навчання нейронної мережі використовуються дві різні функції втрат:

1. Середньоквадратична похибка (MSE) для прогнозування тривалості:

$$L_{duration} = \frac{1}{N} \sum_{i=1}^N (y_{duration,i} - \hat{y}_{duration,i})^2,$$

де:

- $y_{duration,i}$ — це фактична тривалість;
- $\hat{y}_{duration,i}$ — прогнозоване значення для завдання i .

2. Бінарна крос-ентропія для оцінки ризиків:

$$L_{risk} = -\frac{1}{N} \sum_{i=1}^N (y_{risk,i} \log(\hat{y}_{risk,i}) + (1 - y_{risk,i}) \log(1 - \hat{y}_{risk,i})),$$

де:

- $y_{risk,i}$ — фактичне значення ризику для завдання i ;
- $\hat{y}_{risk,i}$ — прогнозована ймовірність ризику.

Для оптимізації використовується алгоритм Adam (Adaptive Moment Estimation), який є варіантом градієнтного спуску. Алгоритм адаптивно оновлює ваги моделі на основі моментів першого та другого порядку градієнтів:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t,$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2,$$

де:

- g_t — градієнт функції втрат;
- m_t та v_t — моменти;
- β_1, β_2 — параметри згладжування.

Оптимізатор Adam використовується через його стійкість до зміни масштабів градієнтів і стабільність у процесі навчання на різних наборах даних. Це особливо важливо для задач прогнозування, де дані можуть бути неоднорідними за своїм масштабом (наприклад, різні естимейти часу) [11].

Після навчання модель перевіряється на тестовій вибірці даних, щоб визначити, наскільки точно вона може передбачати результати. Потім нові завдання подаються на вхід моделі для прогнозування тривалості їх виконання і рівня ризику. Це дозволяє менеджерам проєктів отримувати прогнози ще до початку роботи над завданнями, допомагаючи краще планувати час і ресурси [12].

6. Висновки. Запропонований алгоритм для планування проєктів на основі машинного навчання має низку переваг. Він не лише прогнозує тривалість завдань, але й оцінює ризики, що дає змогу проєктним командам реагувати на можливі проблеми до їх виникнення. Алгоритм використовує рекурентні нейронні мережі для обробки даних і враховує такі ключові показники, як виконавець, початковий естимейт, витрачений час і пріоритет та тип завдання.

Завдяки своїй здатності працювати навіть із невеликими наборами даних, алгоритм є гнучким і придатним для багатьох типів проєктів, незалежно від їх масштабу. Навіть якщо дані обмежені, модель може ефективно навчатися

та генерувати корисні прогнози, що є значною перевагою в умовах, коли доступність історичних даних мінімальна. Однак, із розширенням обсягів даних точність прогнозів і оцінка ризиків значно покращуються.

Ключова перевага в тому, що алгоритм дозволяє автоматично оновлювати плани на основі змін у реальному часі. Це значно полегшує управління великими, складними проєктами, де фактори ризику і непередбачувані події можуть суттєво впливати на результати.

Список використаної літератури

1. Kerzner H. Project Management: A Systems Approach to Planning, Scheduling, and Controlling. John Wiley & Sons, 2017. P. 129–133.
2. Shtub A., Bard J. F., Globerson S. Project Management: Processes, Methodologies, and Economics. Upper Saddle River, NJ : Pearson Prentice Hall, 2005. P. 215–223.
3. Hillson D. Managing Risk in Projects. New York : Routledge, 2017. P. 145–167.
4. Raz T., Shenhar A. J., Dvir D. Risk management, project success, and technological uncertainty. *R&D Management*. 2002. Vol. 32, No. 2. P. 101–109. DOI: <https://doi.org/10.1111/1467-9310.00243>
5. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016. P. 315–350.
6. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd Edition. Springer Series in Statistics : Springer, 2009. P. 389–400.
7. Rapp K. Mastering JIRA 7: An expert guide to building and managing JIRA projects. Packt Publishing, 2018. P. 89–105.
8. Nielsen M. A. Neural Networks and Deep Learning. Determination Press, 2015. P. 120–145.
9. Bishop C. M. Pattern Recognition and Machine Learning. Springer-Verlag New York : Springer, 2006. P. 205–250.
10. Glorot X., Bordes A., Bengio Y. Deep sparse rectifier neural networks. *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Fort Lauderdale, FL, USA, 2011. Vol. 15. P. 315–323.
11. Kingma D. P., Ba J. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR) : Main Conference — Poster Presentations*. Ithaca, NY: ArXiv, 2015. P. 1–13.
12. Schmidhuber J. Deep learning in neural networks: An overview. *Neural Networks*. 2015. Vol. 61. P. 85–117.

Mykhailov N. O. Adaptive model for project planning and risk assessment using machine learning.

Effective project management in the context of constantly changing market demands and technological progress is essential for the success of companies. Optimizing the planning process and risk management is key to improving result quality and reducing project timelines. Traditional methods like Critical Path Method and Program Evaluation and Review Technique help structure project tasks but often fail to account for dynamic changes and unforeseen risks.

The application of modern technologies, such as machine learning, enables the development of flexible and adaptive management systems that incorporate historical data and real-time project changes. Machine learning algorithms can accurately forecast task durations, assess risks, and optimize resource allocation. This article focuses on developing an adaptive model that improves planning accuracy and minimizes risks during project execution.

Keywords: machine learning, risk assessment, neural networks, project management, artificial intelligence, data processing, optimization methods, computational methods, intelligent systems.

References

1. Kerzner, H. (2017). *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*. John Wiley & Sons.
2. Shtub, A., Bard, J. F., & Globerson, S. (2005). *Project Management: Processes, Methodologies, and Economics*. Upper Saddle River, NJ: Pearson Prentice Hall.
3. Hillson, D. (2017). *Managing Risk in Projects*. New York: Routledge.
4. Raz, T., Shenhar, A. J., & Dvir, D. (2002). Risk management, project success, and technological uncertainty. *R&D Management*, 32(2), 101–109. <https://doi.org/10.1111/1467-9310.00243>
5. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
6. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd Edition*. Springer Series in Statistics: Springer.
7. Rapp, K. (2018). *Mastering JIRA 7: An expert guide to building and managing JIRA projects*. Packt Publishing.
8. Nielsen, M. A. (2015). *Neural Networks and Deep Learning*. Determination Press.
9. Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag New York: Springer.
10. Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks. *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Fort Lauderdale, FL, USA, 15.
11. Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*. Ithaca, NY: ArXiv.
12. Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61, 85–117.

Одержано 09.09.2024