**ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ**
**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕКОНОМІКИ, УПРАВЛІННЯ, ПРАВА ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**
**КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ**

# Пояснювальна записка

до кваліфікаційної роботи на здобуття ступеня вищої освіти магістр

на тему: «**Прогнозування ринку ІТ-вакансій в Україні за допомогою нейронної мережі**»

Виконав: здобувач вищої освіти
за освітньо-професійною
програмою Інформаційні
управляючі системи та технології
спеціальності 126 Інформаційні
системи та технології ступеня
вищої освіти магістр
групи 126ІСТ_мз_2022[1](л.н.)
Коваль Д. М.
Керівник: Калініченко А. В.
Рецензент: Біловод О. І

**Полтава – 2023 року**

# ВСТУП

Дана робота присвячена прогнозуванню ринку IT-вакансій в Україні за допомогою інструментарію нейронних мереж (НМ). Ринок IT-праці є одним з найбільш динамічних та важливих сегментів сучасної економіки України, і точне прогнозування його розвитку має велике значення для бізнесу та управління ресурсами. Робота розглядає аналіз ринку, використання сучасних методів машинного навчання (МН), та розробку НМ для прогнозування кількості IT-вакансій.

*Актуальність теми* зумовлена наявністю кількох потужних факторів, що діють на ринку IT-вакансій в Україні. З одного боку, це позитивні тенденції до динамічного зростання IT-галузі в Україні, і, як наслідок, зростання попиту на кваліфікованих IT-фахівців, пов'язаний з необхідністю адаптації бізнесу до швидких технологічних змін. З іншого боку, це загрози, такі як пандемія, що спричинила збільшення попиту на цифрові технології та віддалену роботу, і військовий стан, який ставить нові завдання для IT-галузі, зокрема, у сферах кібербезпеки та розвитку технічних рішень. В цих умовах ефективне прогнозування ринку IT-вакансій є надзвичайно важливим для планування кадрових ресурсів та оптимізації розвитку компаній. Використання нейронних мереж, як найбільш ефективного з методів аналізу даних дозволить отримувати точні та актуальні прогнози щодо ринку IT-вакансій, отже є актуальним в сучасних умовах.

*Зв'язок роботи з науковими програмами, планами, темами.* Робота виконана у відповідності до науково-дослідної ініціативної теми «Організаційно-методологічні аспекти впровадження інформаційно-комунікаційних систем і технологій в управлінні діяльністю сучасних організацій та підприємств за умов переходу до цифрової економіки» ДРН 0117U003099.

*Мета роботи* полягає у розробці та застосуванні моделі прогнозування ринку IT-вакансій в Україні, що базується на використанні нейронних мереж.

*Завдання роботи*:

- збір та обробка історичних даних з різних джерел про ринок IT-вакансій в Україні;

- розробка та налаштування архітектури нейронної мережі для прогнозування попиту на IT-професії та навички;

- навчання, тестування та валідація моделі з метою оцінки її точності та ефективності;

- аналіз та порівняння результатів моделі з іншими методами прогнозування;

- формування висновків та рекомендацій щодо застосування моделі у практиці управління кадровими ресурсами та для подальших досліджень.

*Об'єкт дослідження:* застосування нейронних мереж в аналізі ринку праці в IT-галузі.

*Предмет дослідження:* розробка та використання моделі нейронної мережі для прогнозування ринку IT-вакансій в Україні.

*Методи дослідження*:

- аналіз даних, що включає використання статистичних та аналітичних інструментів для збору, обробки та аналізу історичних даних про IT-вакансії в Україні з метою визначення трендів та шаблонів, які можуть бути використані для подальшого прогнозування;

- моделювання, розробка та тренування моделі нейронної мережі для прогнозування майбутнього попиту на IT-професії, що дозволяє моделювати складні взаємозв'язки між різними факторами на ринку праці;

- проведення експериментів для тестування та валідації моделі нейронної мережі, що включає вибір оптимальних гіперпараметрів і оцінку точності прогнозування моделі;

- порівняння результатів, отриманих за допомогою розробленої нейронної мережі, з результатами існуючих методів прогнозування ринку IT-вакансій, що дозволить оцінити переваги та недоліки розробленої моделі.

*Інформаційна база*: дані з онлайн-платформ пошуку роботи – інформація про вакансії, заробітні плати, вимоги до кандидатів та інші релевантні дані, що публікуються на сайтах пошуку роботи; офіційні дані від державних служб, що стосуються зайнятості в ІТ-галузі, міграції фахівців, ринкових тенденцій тощо; наукові публікації та дослідження – академічні статті, матеріали конференцій та інші наукові роботи, що стосуються використання нейронних мереж у прогнозуванні ринку праці та аналізу ІТ-галузі; звіти аналітичних агенцій та консалтингових компаній, що містять аналіз ринку ІТ-вакансій, тенденції розвитку галузі, огляди ринку праці тощо; інформація від професійних асоціацій та організацій ІТ-галузі – публікації, звіти та аналітичні матеріали від організацій, що представляють інтереси ІТ-галузі; соціальні медіа та форуми професіоналів – аналіз обговорень та тенденцій на професійних форумах, соціальних мережах, блогах, що стосуються ринку ІТ-вакансій; наукові публікації, книги та статті, що описують принципи роботи, архітектуру та застосування нейронних мереж, особливо в контексті прогнозування в економічних та соціальних системах; дослідження та звіти про застосування нейронних мереж для прогнозування, зокрема в галузі управління кадровими ресурсами та аналізу ринку праці; огляди та технічна документація інструментів та платформ для розробки нейронних мереж, таких як TensorFlow, PyTorch, Keras тощо.

*Елементи наукової новизни*: розробка моделі нейронної мережі, яка адаптована для прогнозування тенденцій на ринку ІТ-вакансій в Україні, враховуючи особливості місцевого ринку та його динаміку; порівняння розробленої моделі з традиційними методами прогнозування, що показує її переваги та потенціал для управління кадровими ресурсами в ІТ-галузі; рекомендації щодо використання моделі для планування кадрової політики в компаніях ІТ-сектору.

*Практична значущість* роботи. Розроблена модель нейронної мережі може бути використана ІТ-компаніями в Україні для прогнозування потреб у спеціалістах, що дозволить планувати набір персоналу та розвиток навичок

співробітників відповідно до перспективних тенденцій на ринку. Модель також може бути використана для довгострокового аналізу та планування в умовах швидкозмінного ринку, що допоможе компаніям адаптуватися до ринкових змін та мінімізувати ризики, пов'язані з недостатнім або надмірним найманням спеціалістів. Результати дослідження можуть бути використані й освітніми установами для оновлення та адаптації навчальних програм з урахуванням потреб ринку праці в IT-галузі. Розроблена модель може бути корисною для органів державного управління при розробці стратегій розвитку IT-галузі, плануванні програм підтримки та стимулювання розвитку цифрової економіки.

*Апробація результатів дослідження*. За результатами проведеного дослідження опубліковано тези доповідей: «Співвідношення понять IT-ринку та IT-вакансії у контексті прогнозування IT-ринку та ринку IT-праці», Матер. VIII Всеукраїнської науково-практичної інтернет-конференції «Управління ресурсним забезпеченням господарської діяльності підприємств реального сектору економіки», 23 листопада 2023 року, м. Полтава; «Моделі прогнозування ринку праці та їх застосування в IT-галузі», Матер. VIII Всеукраїнської науково-практичної інтернет-конференції «Управління ресурсним забезпеченням господарської діяльності підприємств реального сектору економіки», 23 листопада 2023 року, м. Полтава.

*Структура та обсяг кваліфікаційної роботи.* Робота складається зі вступу, трьох розділів та висновків. Основний текст роботи викладений на 78 сторінках, містить 2 таблиці, 27 рисунків. Список використаних джерел налічує 69 найменувань.

# РОЗДІЛ 1
# ЗАВДАННЯ ТА МОДЕЛІ ПРОГНОЗУВАННЯ РИНКУ ПРАЦІ В СФЕРІ IT

## 1.1 Співвідношення понять IT-ринку та ринку IT-вакансій

Сфера інформаційних технологій (IT) має надзвичайно важливе значення в України та усьому світі. Саме тому сьогодні значної актуальності набувають нові дослідження за різними аспектами її розвитку.

Огляд інформаційних джерел свідчить, що увага до IT постійно зростала по мірі розвитку автоматизації виробництва на основі обчислювальної техніки (ОТ), та помітно активізувалась із початком надання IT-послуг. Дослідження у цьому напрямку умовно поділяються на такі групи: дослідження шляхів підвищення ефективності виробництва за рахунок впровадження IT, аналіз IT-витрат на підприємстві тощо; IT-ринок, розвиток IT-компаній, конкуренція в IT-бізнесі, ціноутворення, IT-роботодавці, підготовка IT-фахівців, аналіз користувачів IT-послуг, IT-кластери, розвиток інформаційних та комунікаційних технологій та ін.; роль та значення IT у розвитку економіки окремих країн та світу, IT у системі управління конкурентоспроможністю, інвестування сфери IT, подолання «цифрового розриву» та інші проблєми. У той же час, поняття ринку IT-вакансій досі не отримало значної уваги дослідників [1].

Ключовими поняттями даного дослідження є: ринок IT-вакансій, прогнозування ринку праці, НМ, рекурентні НМ (RNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), функція активації, функція втрат, оптимізаційний алгоритм, гіперпараметри, точність моделі, метрики прогнозування, часовий ряд, підготовка даних для навчання, тестовий набір даних, валідація моделі, ресурси IT-компаній, планування персоналу, аналіз результатів та ін. Ці поняття є важливими для розуміння та вивчення теми

роботи. Більшість із них докладно описано в спеціальній літературі й використовується далі у цій роботі.

Важливість правильного розуміння «ринку IT-вакансій» відмічається багатьма дослідниками. Зокрема, зазначається, що «через плутанину у поняттях «IT-сфера», «IT-галузь», «IT-ринок» та ін. часто існує невірне уявлення про стан та тенденції IT-ринку України». Однією з головних причин таких суперечностей вважається «відсутність єдиної думки щодо того, що представляє собою IT-ринок» [1].

Згідно міжнародної експертизи International Data Corporation (IDC), IT-ринок є складовою інформаційно-телекомунікаційного ринку. Він охоплює такі сегменти, як апаратне забезпечення (hardware), програмне забезпечення (software) та IT-послуги (IT-services). Останнім часом до цієї структури долучають «публічні хмари» [2].

Поняття IT-вакансії у даній роботі розглядається, як окреме самостійне поняття. Воно позначає посаду або посадовий обов'язок в галузі IT, яка передбачає виконання робіт із розробки, налагодження, підтримки або управління інформаційними системами (ІС) та програмними продуктами (ПП). IT-вакансія може включати в себе різноманітні спеціалізації, як розробник програмного забезпечення (ПЗ), системний адміністратор, аналітик даних, інженер з безпеки інформації та інші. Такі вакансії вимагають специфічних навичок та знань у сфері інформатики та технологій та є популярними в сучасному ринку праці, особливо в IT-галузі.

У свою чергу, під ринком IT-вакансій розуміється специфічний сегмент ринку праці, що характеризується постійним попитом та пропозицією на робочі місця в галузі IT. Він включає в себе набір IT-посад, які вимагають різних технічних та професійних навичок, таких як розробка програмного забезпечення, адміністрування систем, аналіз даних, тестування програм тощо. Ринок IT-вакансій є динамічним і піддається впливу ряду факторів, таких як технологічний прогрес, попит на конкретні компетенції та економічні умови. Аналіз ринку IT-вакансій включає в себе оцінку потреби в

ІТ-фахівцях, їхню кількість та спеціалізації, що допомагає планувати рекрутингові та управлінські рішення в цій галузі.

Таким чином, поняття «ІТ-вакансії» і «ринок ІТ-вакансій» є взаємопов'язаними і взаємодоповнюються. ІТ-вакансія визначає окрему посаду або конкретне робоче місце в галузі ІТ. Це може бути, наприклад, посада програміста, системного адміністратора, інженера з безпеки, аналітика даних тощо. Тобто ІТ-вакансія описує специфічну посаду зі своїми вимогами та обов'язками. Ринок ІТ-вакансій – більш широке поняття, яке відображає загальну динаміку попиту та пропозиції на ІТ-посади в цілому. Ринок ІТ-вакансій включає в себе всі доступні ІТ-посади у певному географічному регіоні або в певній галузі. Він описує, які технологічні компетенції та професійні навички вимагаються на ринку, як змінюється кількість вакансій з часом та як впливають різні фактори (економічні, технологічні тощо) на ринок ІТ-вакансій. Отже, ІТ-вакансії – це окремі позиції на ринку ІТ-вакансій, і вони разом формують загальну картину ринку праці в галузі інформаційних технологій [3]. У даній роботі розглядається ринок ІТ-вакансій в Україні.

## 1.2 Огляд факторів ринку ІТ-вакансій в Україні

Огляд ринку ІТ-вакансій в Україні включає дослідження стану та динаміки цього ринку, зокрема:

- розвиток галузі – аналіз зростання та розвитку ІТ-галузі в Україні, зокрема, кількості ІТ-компаній, розміщення їх центрів розробки та офісів;

- попит на ІТ-працівників – вивчення попиту на ІТ-спеціалістів, які проявляють компанії на ринку праці, включаючи різні професії та технічні навички;

- пропозиції ІТ-спеціалістів – аналіз кількості та складу робочої сили в галузі ІТ, включаючи рівень освіти та досвіду роботи;

- зарплати та заохочення (бонуси) – вивчення зарплатних ставок, бонусів та інших факторів, які впливають на привабливість IT-вакансій для працівників;

- регіональні особливості – аналіз розподілу IT-компаній та вакансій по різних регіонах України.

Огляд сучасного стану ринку IT-вакансій в Україні проведений на основі даних, представлених у вільному доступі спеціалізованими аналітичними інтернет-ресурсами [4-6]. Діаграма на рис. 1.1 узагальнює дані щодо працевлаштування спеціалістів у сфері IT в Україні станом на жовтень 2023 року [7].

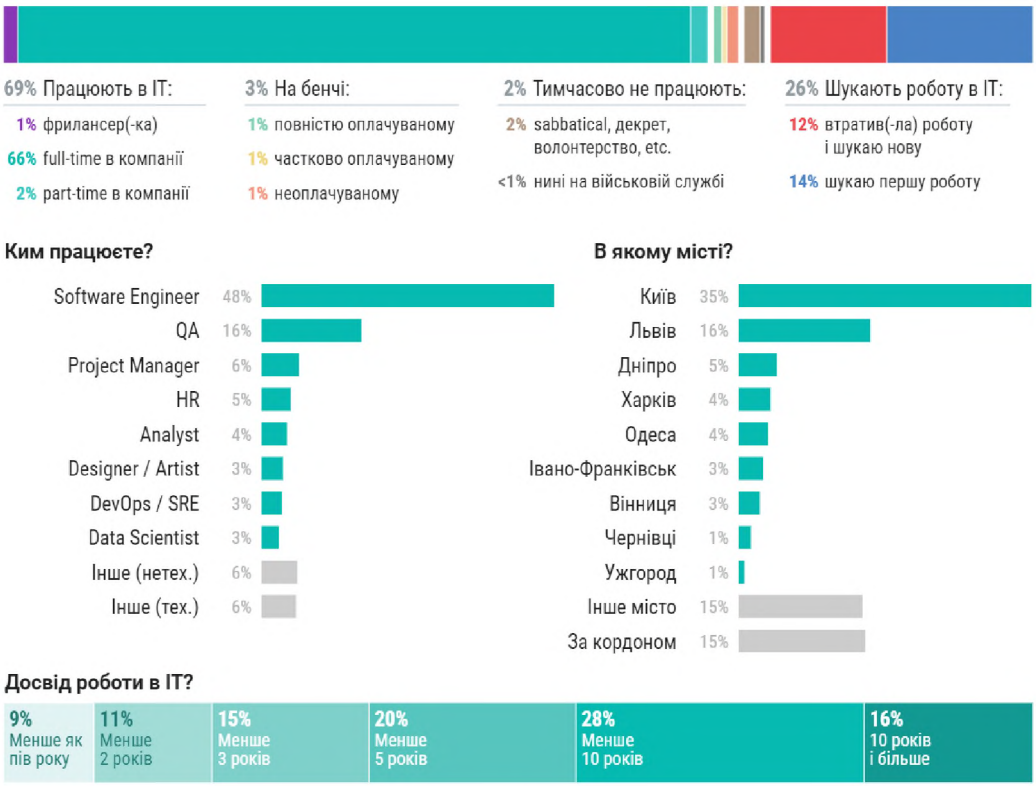



Рисунок 1.1 – Результати опитувань щодо працевлаштування у сфері IT в Україні

За даними аналітичних джерел, близько 66% українських IT-спеціалістів вважають, що знайти роботу зараз важко [4, 8-12]. Найбільше претендентів шукали й знаходили нові вакансії через вебсайт Djinni.co та за рекомендаціями. Водночас, відзначаються проблєми, головні з яких: мала

кількість вакансій, компанії часто не відповідають на звернення, процес співбесіди може бути занадто тривалим. Відомо, що близько половини IT-фахівців отримують пропозиції від рекрутерів через соціальну мережу LinkedIn. Українські IT-ці, які працюють за кордоном, є більш оптимістичними щодо пошуку роботи, ніж їхні колеги в Україні [13].

Особливості та результати пошуку роботи українськими IT-спеціалістами у 2023 році узагальнені на діаграмах (рис. 1.2 – 1.7) [7].

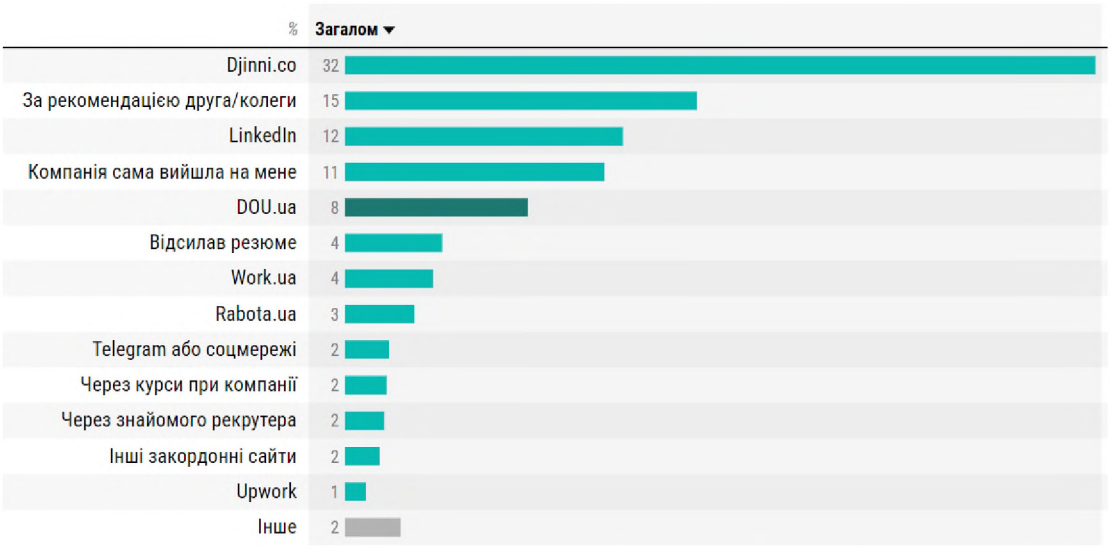
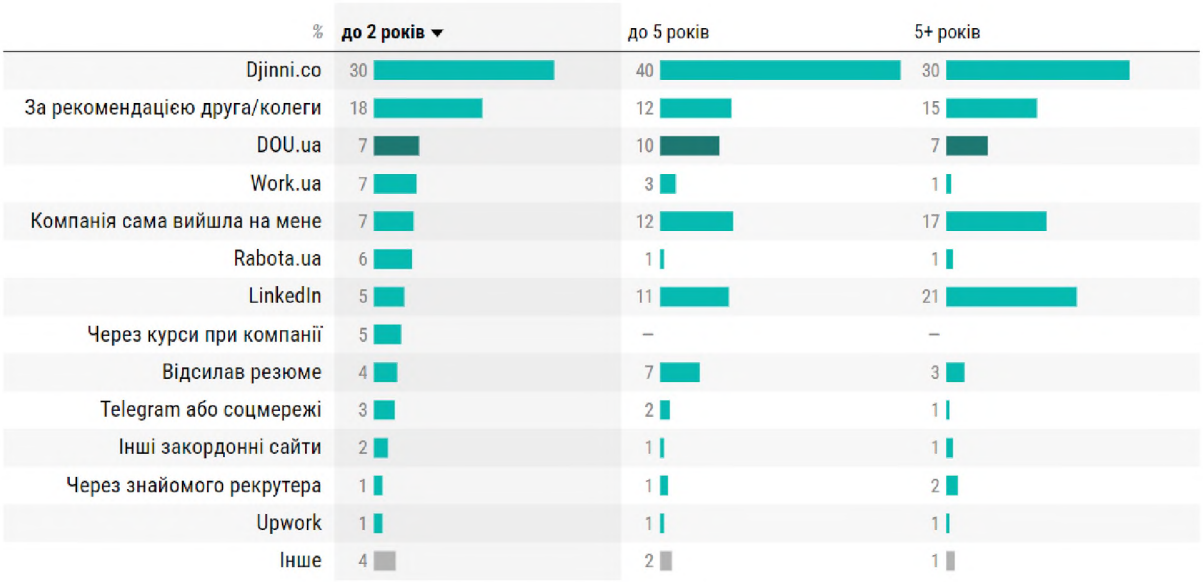



Рисунок 1.2 – Дані щодо способів та результатів пошуку роботи в IT у 2023 році

| % | SE | QA | Інше (нетех.) | PM | Інше (тех.) | HR | Analyst | Designer | DS | DevOps |
|---|---|---|---|---|---|---|---|---|---|---|
| Djinni.co | 34 | 36 | 25 | 42 | 23 | 26 | 36 | 31 | 26 | 22 |
| За рекомендацією друга/колеги | 14 | 17 | 16 | 7 | 26 | 12 | 16 | 22 | 18 | 8 |
| LinkedIn | 13 | 13 | 15 | 10 | 13 | 11 | 9 | 2 | 13 | 16 |
| Компанія сама вийшла на мене | 10 | 7 | 15 | 11 | 8 | 23 | 13 | 7 | 16 | 22 |
| DOU.ua | 8 | 7 | 8 | 10 | 7 | 11 | 9 | 7 | 5 | 8 |
| Відсилав резюме | 4 | 4 | 5 | 1 | 3 | 7 | 4 | 7 | 5 | 3 |
| Work.ua | 4 | 3 | 2 | 4 | 7 | – | 2 | 11 | – | 5 |
| Через курси при компанії | 2 | 1 | 2 | – | 2 | – | 4 | – | 3 | 3 |
| Через знайомого рекрутера | 2 | 1 | 1 | 6 | 2 | 2 | – | – | – | 3 |
| Telegram або соцмережі | 2 | 2 | 1 | 1 | 3 | 2 | 2 | 7 | – | – |
| Rabota.ua | 2 | 4 | 5 | 7 | 5 | 4 | 4 | 2 | 5 | 3 |
| Інші закордонні сайти | 1 | 4 | – | – | 2 | 2 | 2 | – | 3 | 5 |
| Upwork | 1 | 1 | 1 | – | – | – | – | – | 3 | 3 |
| Інше | 3 | 1 | 5 | 1 | 2 | 2 | – | 4 | 3 | – |

Рисунок 1.3 – Способи та результати пошуку роботи в IT у 2023 році за напрямками



| % | до 2 років ▾ | до 5 років | 5+ років |
|---|---|---|---|
| Djinni.co | 30 | 40 | 30 |
| За рекомендацією друга/колеги | 18 | 12 | 15 |
| DOU.ua | 7 | 10 | 7 |
| Work.ua | 7 | 3 | 1 |
| Компанія сама вийшла на мене | 7 | 12 | 17 |
| Rabota.ua | 6 | 1 | 1 |
| LinkedIn | 5 | 11 | 21 |
| Через курси при компанії | 5 | – | – |
| Відсилав резюме | 4 | 7 | 3 |
| Telegram або соцмережі | 3 | 2 | 1 |
| Інші закордонні сайти | 2 | 1 | 1 |
| Через знайомого рекрутера | 1 | 1 | 2 |
| Upwork | 1 | 1 | 1 |
| Інше | 4 | 2 | 1 |

Рисунок 1.4 – Способи пошуку роботи у 2023 році за досвідом

Дані щодо способів пошуку роботи в сфері IT (рис. 1.4) показують, що найбільш популярними способами пошуку роботи IT-спеціалістами з різним досвідом роботи є вебсайти, зокрема, такі, як Djinni.co, DOU.ua, Work.ua, а також рекомендації друзів та колег.

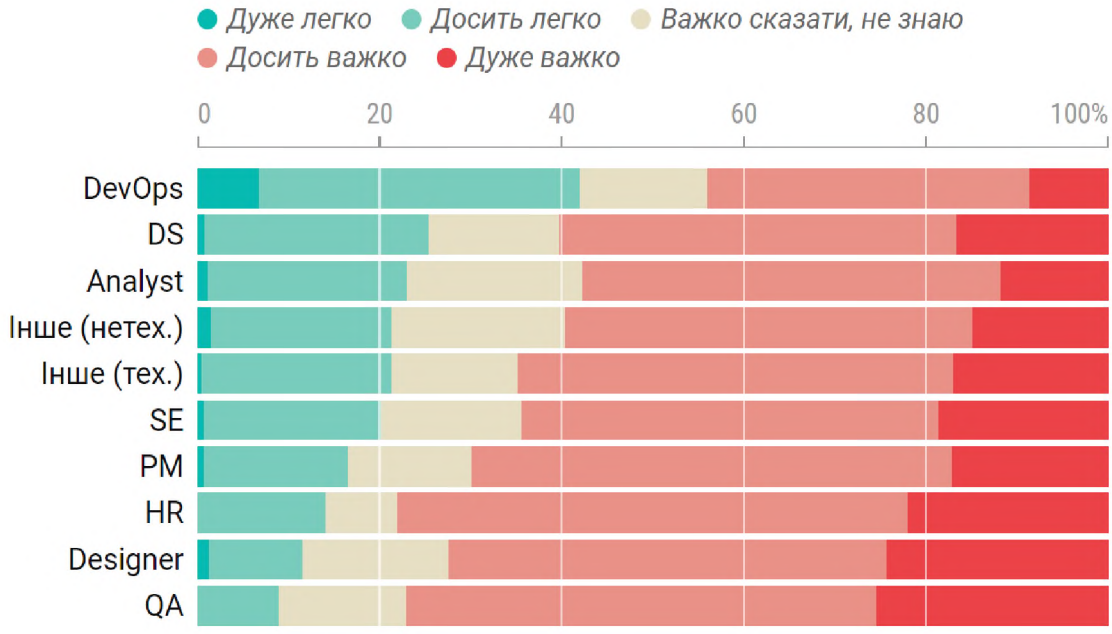



Рисунок 1.5 – Дані щодо труднощів у пошуку роботи за спеціальностями

Аналіз інформації стосовно труднощів у пошуку роботи (рис. 1.5) показує, що найлегше було знайти роботу IT-спеціалістам напрямків DevOps, DS (Data Science), Analyst (аналітика даних), що потребує кваліфікації, яка виходить за межі суто технічних IT-компетенцій, а також, іншим (нетехнічним) працівникам IT-сфери. Навпаки, найбільші складності у пошуку роботи відчувають фахівці напрямків QA (Quality Assurance, забезпечення якості, тестувальники ПЗ), Designer, HR (фахівці з добору персоналу) та PM (Project Management).

Досвід практичної роботи кандидата помітно впливає на результати пошуку роботи у сфері IT (рис. 1.6): із збільшенням досвіду роботи труднощі з пошуком роботи зменшуються. Більше 80% початківців відмічають помітні складності у пошуку роботи, і майже 40% з них оцінюють це, як значні складності. У той же час, серед IT-фахівців з досвідом роботи 3 роки й більше значні труднощі відчувають лише приблизно 15-18% відсотків респондентів.

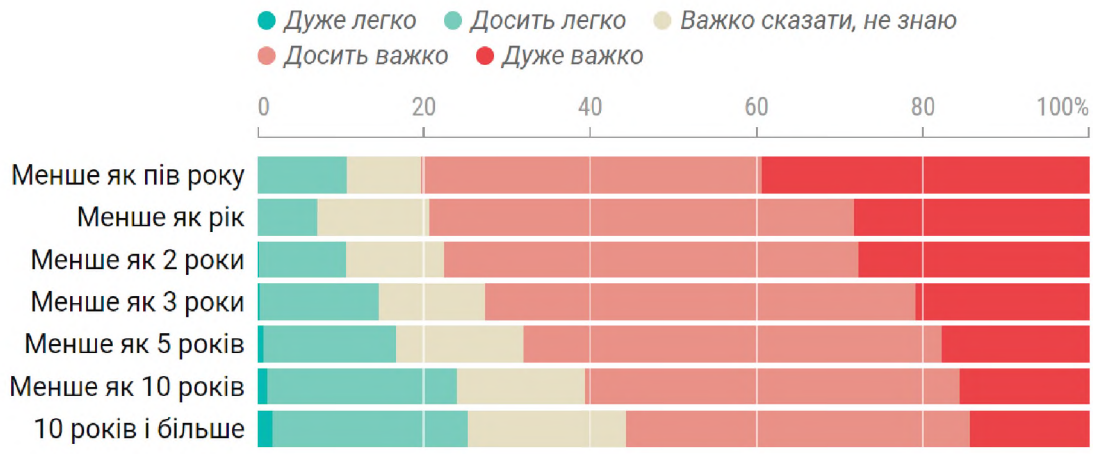



Рисунок 1.6 – Дані щодо оцінювання труднощів у пошуку роботи кандидатами з різним досвідом практичної роботи в сфері IT

Серед основних причин труднощів з пошуком роботи IT-працівники з різним досвідом роботи відмічають: низькі зарплати, тривалість процедури відбору, високі вимоги до кандидатів, малу зацікавленість компаній

(компанії не відповідають). Але на першому місці – мала кількість вакансій та висока конкуренція. У той же час, такі вимоги, як співбесіда англійською мовою, складні тестові завдання, знання теорії, не становлять труднощів для більшості кандидатів з будь-яким досвідом (рис. 1.7).

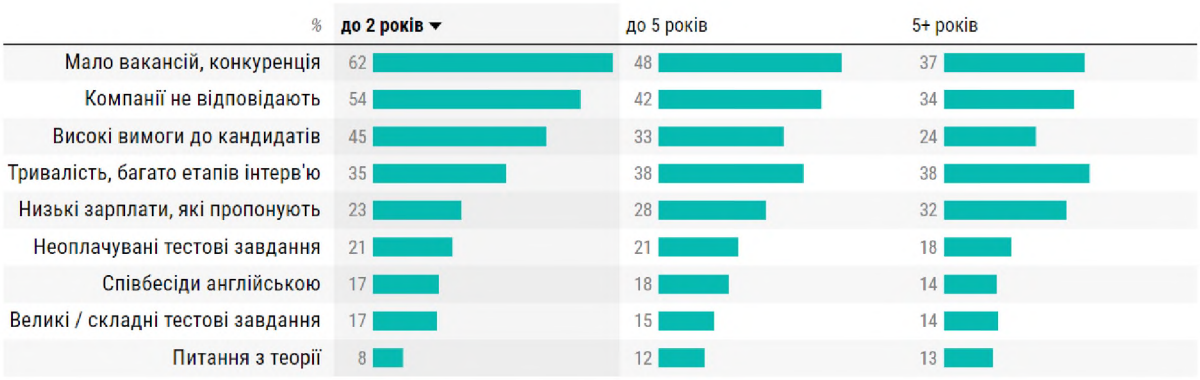



Рисунок 1.7 – Основні причини труднощів у пошуку роботи

За даними DOU.ua, у квітні 2023 року загальна кількість вакансій була мінімальною за останні три роки, що є нижчим показником навіть у порівнянні з першим місяцем повномасштабного вторгнення у березні 2022 року. Конкуренція серед кандидатів досягла рекордного рівня. У той же час, помітно зросла кількість нетехнічних вакансій, особливо для початківців, і понад половина найактивніших компаній у квітні були продуктовими. Найбільше вакансій було у категоріях «від 3 до 5 років досвіду» та «від 1 до 3 років» [5].

Узагальнені дані, що відображають зміни кількості IT-вакансій з квітня 2022 року по квітень 2023 року за категоріями, представлені на рис. 1.8 [13]. Ці дані свідчать, що за цей період найбільше (майже у 2,3 рази) зросла кількість вакансій спеціалістів EPR/CRM. Позитивну динаміку також мали нетехнічні вакансії в IT-сфері. У той же час, негативна динаміка властива практично для всіх вакансій програмістів (C++, Golang, PHP, .NET, Python, Java, Flutter, Front-End, Nide.js, React, iOS, Ruby, Android, Scala), а також спеціалістів напрямку Design, розробників ігор Unreal Engine та Unity, фахівців Big Data, DevOps, Scrum Master, QA тощо.

**Кількість вакансій у квітні 2022 і квітні 2023 за категоріями**

| Посада/Технологія | 2022, квітень | 2023, квітень | Зміна, % |
|---|---|---|---|
| ERP/CRM | 7 | 23 | 229 |
| Office Manager | 7 | 16 | 129 |
| Legal | 11 | 20 | 82 |
| Finances | 39 | 62 | 59 |
| Sales | 165 | 258 | 56 |
| Marketing | 231 | 358 | 55 |
| SEO | 23 | 35 | 52 |
| Copywriter | 22 | 29 | 32 |
| SysAdmin | 50 | 62 | 24 |
| DBA | 13 | 16 | 23 |
| C-level | 14 | 16 | 14 |
| Artist | 45 | 51 | 13 |
| Analyst | 170 | 170 | 0 |
| Other | 152 | 151 | -1 |
| Product Manager | 92 | 90 | -2 |
| Support | 82 | 78 | -5 |
| Security | 37 | 34 | -8 |
| Data Science | 57 | 51 | -11 |
| Embedded | 40 | 35 | -12 |
| Project Manager | 156 | 126 | -19 |
| C++ | 92 | 68 | -26 |
| Design | 193 | 141 | -27 |
| Golang | 52 | 35 | -33 |
| HR | 135 | 91 | -33 |
| PHP | 218 | 128 | -41 |
| Unreal Engine | 22 | 13 | -41 |
| Big Data | 38 | 20 | -47 |
| DevOps | 252 | 131 | -48 |
| Scrum Master | 20 | 10 | -50 |
| .NET | 196 | 97 | -51 |
| Python | 185 | 87 | -53 |
| Java | 263 | 120 | -54 |
| Technical Writer | 16 | 7 | -56 |
| Flutter | 24 | 10 | -58 |
| QA | 365 | 152 | -58 |
| Unity | 59 | 24 | -59 |
| Front End | 453 | 166 | -63 |
| Node.js | 251 | 92 | -63 |
| React Native | 56 | 20 | -64 |
| iOS/macOS | 76 | 26 | -66 |
| Ruby | 69 | 22 | -68 |
| Salesforce | 28 | 9 | -68 |
| Android | 83 | 25 | -70 |
| Scala | 21 | 6 | -71 |

Рисунок 1.8 – Зміни у кількості IT-вакансій в Україні за категоріями за рік

Загальне уявлення про динаміку ринку IT-вакансій в Україні протягом року надає графік зміни загальної кількості вакансій з квітня 2022 року по квітень 2023 року, що демонструє незначні коливання з тенденцією до спадання (рис. 1.9) [12].


Рисунок 1.9 – Динаміка кількості IT-вакансій в Україні у 2023 році

Графік на рис. 1.10 відображає загальну динаміку кількості технічних посад для IT-фахівців з ERP/CRM протягом у 2023 році [12].

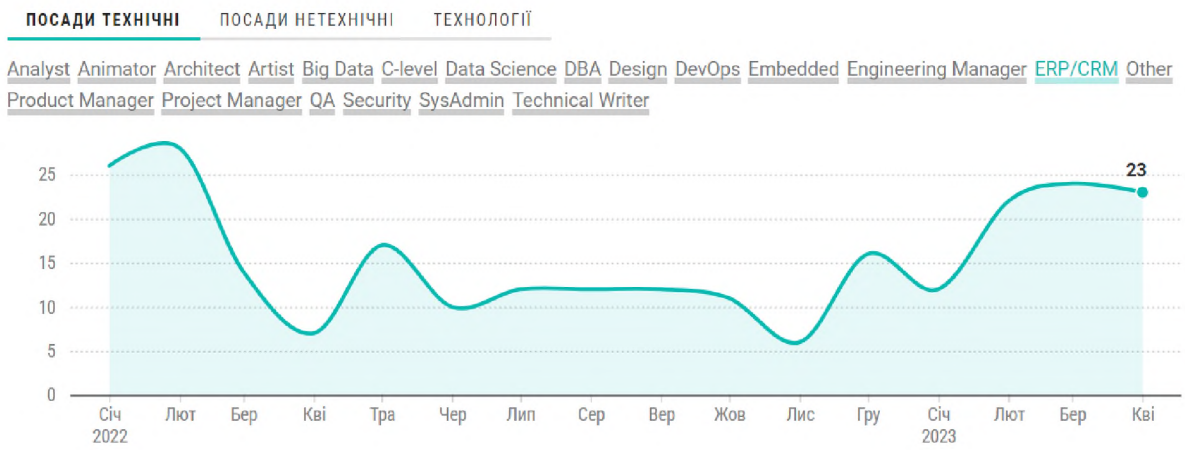



Рисунок 1.10 – Динаміка кількості технічних посад для IT-фахівців з ERP/CRM

Наведені дані дозволяють зробити висновки, що ринок IT-вакансій в Україні є високо конкурентним та характеризується певними труднощами з пошуком роботи, особливо для фахівців технічних напрямків. При цьому,

зростає кількість вакансій у нетехнічних напрямках та для фахівців з досвідом від одного до п'яти років.

Загалом, ринок ІТ-праці в Україні зазнав суттєвих змін впродовж останніх років. Незважаючи на те, що ІТ-індустрія продемонструвала рекордні показники експорту у 2022 році, кількість нових вакансій при цьому скоротилася на 13%. Станом на квітень 2023 року, кількість кандидатів, що шукають роботу, становила 82 тисячі, у той час як вакансій було лише 9,5 тисяч, тобто кандидатів у 8,5 рази більше, ніж пропозицій. Зарплати для більшості ІТ-фахівців залишились на рівні кінця 2021 року, зарплатні очікування деяких категорій знизились на 10% у 2023 році [7].

Таким чином, дані аналітики свідчать про високу конкуренцію на ринку ІТ-праці в Україні, з великою кількістю шукачів і малою кількістю вакансій. Зарплати, в основному, залишаються на рівні попередніх років. При цьому, спостерігається зменшення зарплатних пропозицій для окремих категорій ІТ-спеціалістів.

Основними факторами, які впливають на результати кандидатів на ринку ІТ-вакансій України є: посада, місто (регіон), статус та досвід роботи кандидата (має досвід, без досвіду в ІТ), спосіб пошуку роботи, час (тривалість) пошуку, активність рекрутерів тощо.

## 1.3 Моделі прогнозування ринку праці та їх застосування в ІТ-галузі

Традиційні (класичні) підходи до прогнозування ринку праці дозволяють отримати узагальнене уявлення про тенденції на ринку, виявити потенційні виклики та можливості для розвитку ІТ-галузі. Вони важливі для роботодавців, політиків, освітніх інституцій та інших зацікавлених сторін для планування і реагування на майбутні зміни на ринку праці. Зокрема, вони включають економетричні моделі та моделі часових рядів. Економетричні

моделі описують історичні дані для виявлення кореляції та залежності між різними економічними змінними. В IT-галузі економетричні моделі можуть використовуватися для аналізу трендів зайнятості, зарплат, інвестицій у технології, тощо. Часові ряди: цей підхід передбачає аналіз даних, що змінюються з часом. В IT-секторі часові ряди можуть допомогти передбачити майбутні тенденції на ринку праці, наприклад, зростання попиту на певні технологічні навички або зміну зарплатних очікувань.

Економетричні моделі є одним з основних інструментів для аналізу економічних даних. Ці моделі використовують статистичні методи для оцінювання і перевірки економічних теорій, а також для прогнозування майбутніх тенденцій. Вони включають [13, 14, 15]:

- лінійні регресійні моделі; використовуються для оцінки взаємозв'язків між змінними. Наприклад, можна використовувати лінійну регресію для аналізу, як зміни в технологічних інвестиціях впливають на кількість IT-вакансій;

- моделі часових рядів, наприклад, ARIMA (авторегресійні інтегровані моделі ковзних середніх), які можуть використовуватися для прогнозування майбутніх тенденцій, наприклад, змін у заробітних платах IT-фахівців на основі історичних даних;

- логістична регресія; цей метод може бути застосований для прогнозування ймовірності певних подій, таких як можливість отримання роботи в IT-секторі залежно від набору кваліфікацій або досвіду.

Для створення та аналізу таких моделей часто використовують мови програмування R [16] або Python [17]. У Python для роботи з економетричними моделями можна використати бібліотеку statsmodels [18]. Фрагмент коду, який демонструє побудову лінійної регресії за допомогою Python-бібліотеки statsmodels представлений у додатку А (лістингА.1). Наведений код ілюструє базове використання лінійної регресії для оцінювання впливу технологічних інвестицій та інших факторів на кількість IT-вакансій (рис. 1.11).

```python
import statsmodels.api as sm
import pandas as pd

# Завантаження даних
data = pd.read_csv('data.csv')  # data.csv містить ваші дані

# Визначення залежної та незалежних змінних
X = data[['технологічні_інвестиції', 'інші_фактори']]  # незалежні змінні
y = data['кількість_IT_вакансій']  # залежна змінна

# Додавання константи до незалежних змінних
X = sm.add_constant(X)

# Створення та навчання моделі
model = sm.OLS(y, X).fit()

# Виведення результатів
print(model.summary())
```

Рисунок 1.11 – Використання Python-бібліотеки statsmodels для оцінювання впливу технологічних інвестицій та інших факторів на кількість IT-вакансій методом лінійної регресії

Часові ряди – це послідовність точок даних, виміряних через рівні проміжки часу. Моделі часових рядів є потужним інструментом для аналізу даних, які мають таку структуру [19]. В IT-галузі їх використовують для аналізу тенденцій зайнятості, заробітних плат, попиту на певні технології тощо. Типові приклади застосування моделей часових рядів в IT-сфері:

- прогнозування зростання кількості IT-фахівців у певній області;
- аналіз сезонних коливань у попиті на IT-послуги;
- прогнозування майбутніх заробітних плат у IT-галузі на основі історичних даних.

Для аналізу часових рядів засобами Python також можна використовують бібліотеку statsmodels. Наприклад, для аналізу часових рядів може бути використана модель ARIMA (авторегресійна інтегрована модель

ковзного середнього), що дозволяє аналізувати та робити прогнози на основі історичних даних, враховуючи тренди, сезонність та інші фактори [20]. Код Python, який демонструє використання ARIMA для аналізу та прогнозування часових рядів у контексті заробітних плат в IT-галузі, представлений у додатку А (лістингА.2) (рис. 1.12).

```
import numpy as np
import pandas as pd
import statsmodels.api as sm
from statsmodels.tsa.arima.model import ARIMA

# Завантаження та підготовка даних
data = pd.read_csv('data.csv')  # 'data.csv' містить ваші часові дані
ts = data['зарплата']  # припустимо, ми аналізуємо часовий ряд заробітних плат

# Побудова моделі ARIMA
model = ARIMA(ts, order=(5,1,0))
model_fit = model.fit()

# Прогнозування
forecast = model_fit.forecast(steps=5)
print(forecast)
```

Рисунок 1.12 – Використання Python-бібліотеки statsmodels для аналізу часових рядів

Методи МН також можуть бути ефективно використані для прогнозування попиту на IT-працівників [21, 22]:

- лінійна регресія – один з найпростіших методів МН, який використовується для прогнозування кількісних значень. Наприклад, можна використовувати лінійну регресію для прогнозування заробітної плати IT-спеціалістів на основі їх досвіду, освіти, та інших факторів;

- дерева рішень (decision trees) – цей метод використовується для класифікації або регресії. Дерева рішень дозволяють визначити, які фактори найбільше впливають на попит на певні IT-спеціальності;

- методи класифікації, такі як логістична регресія, Random Forest [23], SVM (Support Vector Machine) [24, 25] застосовуються для класифікації IT-фахівців за різними категоріями (наприклад, за рівнем досвіду, спеціалізацією, очікуваною зарплатою) та прогнозування їх працевлаштування в залежності від ринкових умов.

Вказані методи МН можуть аналізувати великі обсяги даних, ідентифікувати закономірності та тенденції, які не завжди очевидні при традиційному аналізі, і надавати точніші прогнози щодо майбутнього ринку IT-праці.

Лінійна регресія є основним інструментом у МН для прогнозування кількісних змінних. Її використовують для оцінки, наприклад, як різні фактори впливають на заробітну плату IT-працівників. Код Python, який демонструє базовий процес створення, навчання та оцінки моделі лінійної регресії для прогнозування заробітної плати на основі факторів, таких як досвід, освіта та місто проживання представлений у додатку А (лістингА.3).

Дерева рішень – метод МН, який використовується для класифікації або регресії. Вони мають інтуїтивно зрозумілу структуру, що дозволяє легко візуалізувати, як досягається рішення. Дерева рішень можуть бути застосовані для прогнозування попиту на IT-працівників, аналізуючи такі фактори, як спеціалізація, досвід, освіта тощо [26].

Приклад використання дерева рішень в Python за допомогою бібліотеки scikit-learn представлений у додатку А (лістингА.4). У представленому коді дерево рішень навчається на основі даних про IT-спеціалістів (спеціалізація, досвід, освіта) та їхнє працевлаштування, щоб спрогнозувати, чи буде конкретний спеціаліст працевлаштований.

Методи класифікації в МН використовуються для розпізнавання та категоризації даних. В IT-сфері вони можуть застосовуватися для класифікації IT-спеціалістів, прогнозування успішності працевлаштування тощо. До популярних методів класифікації належать [27]:

- логістична регресія – використовується для прогнозування ймовірності події (наприклад, чи працевлаштується спеціаліст у певній компанії);

- Random Forest – метод, який використовує ансамбль дерев рішень для покращення точності та зменшення перенавчання;

- Support Vector Machine (SVM) – використовується для визначення межі між різними класами.

Приклад коду на Python з використанням бібліотеки scikit-learn для методу Random Forest представлений у додатку А (лістингА.5). У цьому коді модель Random Forest навчається на тренувальному наборі даних, її точність перевіряється на тестовому наборі, що дозволяє оцінити ефективність класифікації.

Прогностичні моделі в ІТ-галузі включають [28]:

- моделі часових рядів для прогнозування тенденцій, таких як попит на певні технології або заробітні плати;

- економетричні моделі – використовують для аналізу історичних даних для прогнозування економічних трендів у ІТ-секторі;

- методи МН (лінійна регресія, дерева рішень, та SVM) – використовуються для аналізу великих даних, прогнозування зайнятості, заробітних плат та інших тенденцій.

Використання таких моделей допомагає ІТ-компаніям та ІТ-фахівцям адаптуватися до змін на ринку, планувати бізнес-стратегії та оптимізувати процеси найму та пошуку роботи.

## 1.4 Нейронні мережі та їх застосування у прогнозуванні ринку праці

Нейронні мережі (НМ) – потужні інструменти МН, які набувають все більшої популярності у різних сферах, у тому числі й для прогнозування

ринку праці. НМ здатні аналізувати великі обсяги даних, що дозволяє з високою точністю прогнозувати такі аспекти ринку праці, як попит на певні професії, динаміку заробітних плат, зміни в технологічних трендах та інші важливі параметри [29, 30].

Застосування НМ у прогнозуванні ринку ІТ-праці [31]:

- аналіз трендів попиту на навички. Дійсно, НМ можуть аналізувати великі масиви даних, такі як оголошення про вакансії, для виявлення зростаючих трендів у попиті на певні технологічні навички або мови програмування;

- прогнозування зміни заробітних плат. Використовуючи історичні дані, НМ здатні прогнозувати майбутні зміни в заробітних платах у різних ІТ-спеціальностях;

- оцінка впливу технологічних змін. НМ дозволяють прогнозувати, як нові технології (наприклад, штучний інтелект (ШІ) та МН) можуть вплинути на ринок праці в ІТ.

Аналіз трендів попиту на навички за допомогою НМ включає наступні етапи:

- збір даних – збирання великих масивів даних з оголошень про вакансії, пов’язаних з ІТ-сферою, включаючи запитувані навички, технології, мови програмування тощо;

- попередня обробка даних – підготовка даних для аналізу, що може включати очищення даних, кодування категорійних змінних та нормалізацію.

- навчання НМ – використання НМ для виявлення залежностей та закономірностей у даних, які, наприклад, можуть вказувати на майбутні тренди попиту на певні навички. Кодування та навчання НМ може бути реалізоване за допомогою бібліотек Python, таких як TensorFlow [32] або PyTorch [33].

У додатку А (лістинги А.5 і А.6) наведені приклади коду Python, який показує використання TensorFlow та PyTorch для створення та навчання НМ. Представлений код з використанням TensorFlow створює та навчає просту

НМ для задачі класифікації. Другий код створює НМ з двома прихованими шарами та використовує PyTorch для навчання мережі.

Таким чином, НМ відіграють провідну роль у прогнозуванні ринку праці, зокрема в ІТ-сфері. Вони здатні аналізувати великі обсяги даних, виявляючи складні залежності та тренди, які важко виявити за допомогою традиційних методів. Це дозволяє прогнозувати попит на певні навички, зміни заробітних плат та вплив нових технологій на ринок праці. Використання НМ сприяє кращому розумінню динаміки ринку та допомагає компаніям і професіоналам адаптуватися до майбутніх змін у ІТ-галузі.

## Висновки до розділу 1

Розглянуто ключові аспекти прогнозування ринку праці в ІТ-сфері:

- проведено аналіз співвідношення понять ІТ-ринку та ринку ІТ-вакансій, відмічено взаємозв'язок між загальними тенденціями на ІТ-ринку та специфікою ринку ІТ-вакансій;
- зроблено огляд ринку ІТ-вакансій в Україні;
- проведено аналіз поточного стану ринку ІТ-вакансій в Україні, його особливостей та динаміки;
- розглянуто існуючі моделі прогнозування ринку праці та їх застосування в ІТ-галузі – оцінюються різні підходи до прогнозування, включаючи економетричні моделі, часові ряди та МН;
- розглянуто роль НМ у прогнозуванні ринку праці – загальні підходи у застосуванні НМ для аналізу та прогнозування трендів на ринку праці в ІТ.

# РОЗДІЛ 2
# ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ У ЗАДАЧАХ ПРОГНОЗУВАННЯ

## 2.1 Вибір архітектури нейронної мережі

Вибір НМ для конкретної задачі включає ряд етапів [29, 30].

Визначення задачі. На початку дослідження необхідне чітке розуміння проблєми, яку потрібно вирішити – класифікація, регресія, прогнозування часових рядів тощо.

Вибір типу мережі [34, 35]:

- повнозв'язні НМ – добре підходять для загальних задач прогнозування;

- згорткові НМ (CNN) – ефективні для зображень та візуальних даних;

- рекурентні НМ (RNN) – краще підходять для аналізу часових рядів та послідовностей.

Визначення розмірності та обсягу даних. Слід враховувати, що великі набори даних можуть вимагати більш складної НМ, але зі збільшенням складності зростає ризик її перенавчання.

Комп'ютерні ресурси. Зрозуміло, що більш складні моделі вимагають більше обчислювальних ресурсів.

Експериментування та налаштування. Вибір оптимальної архітектури НМ передбачає проведення численних експериментів з метою визначення оптимального налаштування параметрів мережі.

Задачі класифікації у контексті застосування НМ мають декілька ключових аспектів [35], а саме: визначення класів, вибір архітектури мережі, попередня обробка даних, видалення або заповнення відсутніх даних, розділення даних на навчальну та тестову вибірки, векторизація текстових даних, створення нових ознак, вибір функції втрат, оцінка точності моделі, запобігання перенавчанню тощо.

Визначення класів. Задача полягає у віднесенні вхідних даних до певних категорій або класів.

Вибір архітектури мережі. Це можуть бути повнозв'язні мережі, згорткові або рекурентні мережі тощо, залежно від типу даних, а саме:

- для зображень зазвичай використовуються згорткові НМ (CNN), які ефективно використовуються для розпізнавання об'єктів на зображеннях. CNN підходять для обробки візуальних даних, таких як зображення графіків продуктивності ІТ-систем, аналіз інтерфейсів користувача тощо;

- для текстових даних використовують переважно рекурентні НМ (RNN) або архітектури на основі уваги, такі як Transformer, що здатні обробляти послідовності мовних даних, наприклад, для аналізу емоцій тексту, і використовуються для аналізу лог-файлів серверів або моніторингу даних в реальному часі, де дані представляють собою послідовності;

- для структурованих даних краще підходять повнозв'язні мережі, які аналізують табличні дані, наприклад, для прогнозування відтоку клієнтів, трендів на ринку ІТ-послуг або зайнятості.

- для часових рядів найбільш придатні рекурентні або конволюційні мережі, призначені для аналізу часових послідовностей. Наприклад, у сфері ІТ рекурентні та конволюційні НМ можуть застосовуватися для аналізу часових послідовностей, наприклад, у моніторингу стану мережевого трафіку. Рекурентні мережі (RNN), зокрема LSTM (Long Short-Term Memory) мережі, ефективні для прогнозування аномалій у мережевому трафіку, оскільки вони здатні запам'ятовувати інформацію протягом тривалого часу. Конволюційні НМ також можуть бути використані для аналізу часових послідовностей у мережевих даних для виявлення важливих шаблонів (патернів) у даних. Для прогнозування ринку ІТ-вакансій, рекурентні та конволюційні НМ можуть бути використані для аналізу трендів зайнятості та попиту на певні технологічні навички. Наприклад, LSTM мережі здатні виявляти тенденції у зміні попиту на різні технології чи програмувальні мови, аналізуючи часові послідовності даних з оголошень про вакансії. Це

допомагає у прогнозуванні майбутнього розвитку IT-ринку та попиту на певні професійні навички. Отже, у задачах класифікації зображення зазвичай обробляються за допомогою CNN, текстові дані – RNN або Transformer, структуровані дані – повнозв'язними мережами, а часові ряди – RNN або CNN.

Попередня обробка даних. Це підготовка вихідних даних для навчання НМ, включаючи нормалізацію, кодування категорійних змінних тощо. Обробка даних для тренування НМ у контексті прогнозування ринку IT-вакансій може включати наступні етапи:

- нормалізація даних – процес масштабування числових змінних до певного діапазону (наприклад, від 0 до 1). Це важливо для того, щоб жодна змінна не мала непропорційного впливу на процес навчання НМ. Наприклад, у даних про IT-вакансії можна нормалізувати заробітні плати;

- кодування категорійних змінних. Якщо дані містять категорійні змінні, наприклад, назви технологій або мов програмування, вони повинні бути перетворені у формат, який може обробляти НМ. Це може бути зроблено за допомогою методів, таких як one-hot encoding, який полягає у перетворенні категорійних змінних у бінарний формат. Кожна категорія представлена окремим стовпцем, де одна категорія відповідає значенню 1, а всі інші – 0. Цей метод використовується для перетворення номінальних даних у формат, придатний для обробки методами МН [36, 37].

Видалення або заповнення відсутніх даних. У даних про IT-вакансії можуть бути випадки відсутності деякої інформації. Важливо вирішити, чи слід видаляти такі рядки, або заповнювати відсутні дані, наприклад, за допомогою середніх значень або медіан: якщо в оголошенні про IT-вакансію відсутня важлива інформація (наприклад, заробітна плата або вимоги до навичок), такий рядок може бути видалений з датасету; якщо ж в оголошенні про вакансію відсутні дані, але вони вважаються не критичними (наприклад, рік заснування компанії), можна заповнити ці пропуски середнім значенням або найбільш частим значенням з відповідного стовпця. Приклад коду на

Python, що демонструє автоматичне видалення та заповнення відсутніх даних, наведений у додатку А (лістингА.8).

Розділення на навчальну та тестову вибірки. Дані розділяють на дві частини – одна для навчання моделі, а інша для перевірки її ефективності. Це забезпечує можливість оцінити, наскільки добре модель працює на нових, невідомих їй раніше даних. Розділення на навчальну та тестову вибірки – процес, під час якого вихідний набір даних поділяється на дві частини: одна частина використовується для навчання моделі (навчальна вибірка), а інша – для перевірки її ефективності та точності (тестова вибірка). У контексті аналізу ринку IT-вакансій, цей процес може виглядати так: датасет із даними про вакансії, їхні характеристики та вимоги поділяється, і модель навчається розпізнавати певні тренди або здійснювати прогнозування (наприклад, заробітної плати) на основі навчальної вибірки, а потім її точність перевіряється на тестовій вибірці. У додатку А (Лістиг А.9) наведено приклад коду на Python, який використовує бібліотеку Scikit-Learn для розділення даних на навчальну вибірку (80% даних) та тестову вибірку (20% даних).

Векторизація текстових даних. Якщо у даних про IT-вакансії є текстові описи, їх потрібно перетворити у числовий формат за допомогою спеціальних технік, як TF-IDF (Term Frequency-Inverse Document Frequency) або векторизація слів. Це дозволяє НМ ефективно обробляти текстову інформацію. TF-IDF – техніка векторизації тексту, яка вимірює важливість слова у контексті документа, який є частиною корпусу. У контексті текстового аналізу та обробки природної мови, корпус – це набір текстових документів. Це може бути будь-яка колекція написаних матеріалів, як-от книги, статті, оголошення про вакансії, блоги тощо. Корпус слугує як основа для аналізу мови, включаючи векторизацію тексту за допомогою методів, як-от TF-IDF. У випадку аналізу IT-вакансій, корпусом можуть бути всі текстові описи цих вакансій. «Term Frequency» означає частоту слова в документі, а «Inverse Document Frequency» зменшує вагу слів, які зустрічаються часто у всьому корпусі. Таким чином, TF-IDF дозволяє визначити, які слова є

важливими у конкретному документі [38, 39]. Приклад коду на Python для векторизації текстових даних з використанням TF-IDF представлений у додатку А (лістингА.10). У цьому коді TfidfVectorizer з бібліотеки scikit-learn використовується для перетворення колекції текстових даних (описів IT-вакансій) у TF-IDF-матрицю.

Створення нових ознак. Іноді корисно створювати нові ознаки на основі існуючих даних, щоб підвищити точність прогнозів. Наприклад, з дати публікації вакансії можна вивести сезонність попиту на IT-спеціалістів. Створення нових ознак (feature engineering) – процес перетворення вихідної інформації у формат, який може бути ефективно використаний для МН. У контексті аналізу ринку IT-вакансій створення нових ознак може включати:

- визначення ключових слів – виділення ключових технологій або навичок з описів вакансій;

- категоризація рівнів досвіду – класифікація вакансій за рівнем кваліфікації, наприклад, junior, middle, senior;

- географічне кодування – перетворення місцеположення компаній у векторні представлення для аналізу географічних трендів.

На діаграмі (рис. 2.1) представлені основні етапи обробки даних для навчання НМ, які включають нормалізацію даних, кодування категорійних змінних, видалення або заповнення відсутніх даних, розділення на навчальну та тестову вибірки, векторизацію текстових даних та створення нових ознак.

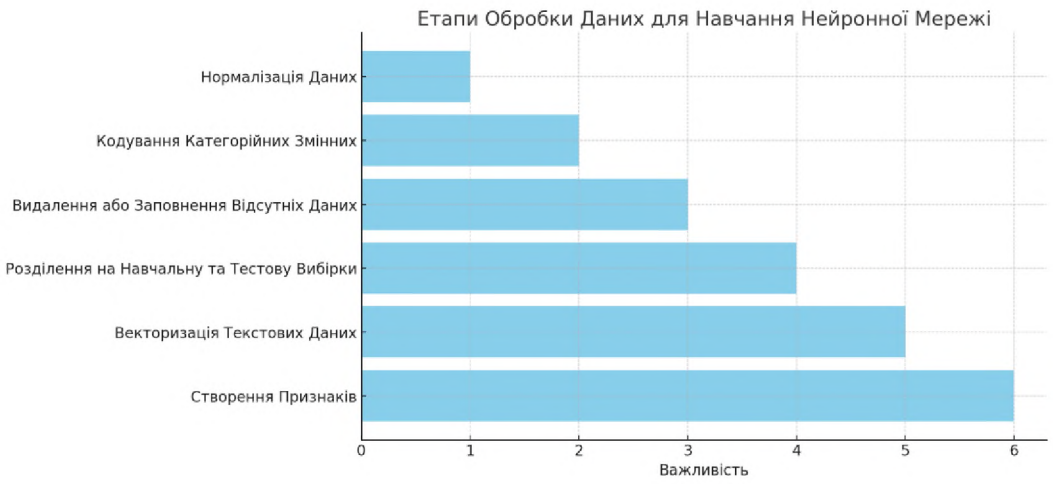



Рисунок 2.1 – Етапи обробки даних для навчання НМ

Функція втрат у МН визначає, наскільки добре модель прогнозує очікувані результати [40]. У задачах класифікації ІТ-вакансій, у якості функції втрат часто використовується «крос-ентропія». Зокрема, вона застосовується у задачах бінарної класифікації (наприклад, для визначення, чи є вакансія високооплачуваною або ні) або для багатокласової класифікації (наприклад, класифікація вакансій за рівнем складності: junior, middle, senior). Функція втрат оцінює, наскільки прогнози моделі відрізняються від фактичних даних, допомагаючи направляти процес навчання. У додатку А (лістингА.11) наведено приклад коду на Python, де використовується крос-ентропія як функція втрат у задачі класифікації з використанням TensorFlow. У цьому прикладі sparse_categorical_crossentropy використовується як функція втрат для моделі, що виконує багатокласову класифікацію.

Оцінка точності – використання метрик для оцінки якості класифікації. Оцінка точності в задачах класифікації, як-то аналіз ринку ІТ-вакансій, включає використання таких метрик [41]:

- точність (accuracy) – відсоток випадків, коли модель правильно класифікувала вакансії (наприклад, вірно визначила рівень позиції: junior, middle, senior);

- F1-міра – характеризує баланс між точністю (precision) та повнотою (recall). Важлива у випадках, коли класи нерівномірно представлені (наприклад, якщо вакансій для senior спеціалістів значно менше, ніж для junior).

У додатку А (лістингА.12) наведено код на Python, що демонструє оцінку точності та F1-міри у задачах класифікації з використанням бібліотеки scikit-learn. У коді спочатку використовується модель Random Forest для вирішення завдання класифікації, а потім оцінюється її точність (accuracy) та F1-міра на тестових даних.

Запобігання перенавчанню – застосування методів регуляризації та перевірка моделі на тестових даних. Запобігання перенавчанню в НМ полягає у використанні методів, які допомагають моделі уникнути занадто точного

запам'ятовування навчальних даних, що може погіршити її здатність до загальної класифікації. До цих методів належать [42]:

- регуляризація – застосування штрафів на великі ваги у мережі, що допомагає уникнути занадто специфічного навчання. Наприклад, L1 та L2 регуляризації [43-45];

- перевірка на тестових даних – оцінка моделі на даних, які не використовувалися під час навчання, для визначення її здатності до узагальнення.

Приклад коду на Python з використанням TensorFlow, де застосовується L2 регуляризація наведений у додатку А (лістингА.13). У цьому коді використовується L2 регуляризація (l2(0.01)), яка додає до функції втрат штраф за великі ваги е шарах мережі, допомагаючи запобігти перенавчанню.

Задачі регресії з використанням НМ мають свої особливості, зокрема, це: прогнозування кількісних значень, вибір функції втрат та архітектури мережі [46].

Прогнозування кількісних значень. На відміну від класифікації, основна мета регресії полягає у прогнозуванні неперервних числових значень. Одним із прикладів задачі регресії у контексті аналізу ринку IT-вакансій може бути прогнозування заробітної плати на основі різних факторів, таких як технології, рівень досвіду, географічне розташування компанії, спосіб та тривалість пошуку вакансії тощо. Мета полягає в тому, щоб створити модель, яка зможе точно передбачати заробітну плату для конкретної вакансії, враховуючи набір вхідних даних;

Функція втрат. Зазвичай, у задачах регресії у якості функції втрат використовується середньоквадратична помилка (MSE) або інші подібні метрики для вимірювання точності прогнозів. Загалом, у задачах регресії часто застосовують такі функції втрат [47]:

- середньоквадратична помилка (MSE, Mean Squared Error) – середнє квадратів різниць між фактичними та прогнозованими значеннями. MSE

широко використовується через її простоту та ефективність у багатьох ситуаціях;

- середня абсолютна помилка (MAE, Mean Absolute Error) – середнє абсолютних значень різниць між фактичними та прогнозованими значеннями. MAE є менш чутливою до викидів порівняно з MSE;

- Huber Loss – це компроміс між MSE та MAE. Функція Huber Loss менш чутлива до викидів порівняно з MSE і тому часто використовується у задачах, де присутні викиди в даних.

Приклад коду на Python з використанням функції втрат у задачі регресії представлений у додатку А (лістингА.14). У наведеному прикладі для оцінки точності моделі лінійної регресії у задачі прогнозування заробітної плати використовуються середньоквадратична помилка (MSE) та середня абсолютна помилка (MAE). Ці функції допомагають оцінити, наскільки добре модель відтворює реальні дані, вибір конкретної функції залежить від специфіки задачі та даних.

Архітектура мережі у задачах регресії може бути різноманітною, але останній шар НМ у задачах регресії зазвичай має один вихідний нейрон (або кількість нейронів співпадає з кількістю прогнозованих змінних), без функції активації або з лінійною активацією для прогнозування числового значення. Отже, у задачах регресії з використанням НМ, як правило, останній шар моделі часто має лише один нейрон, що відповідає за прогнозування одного числового значення (наприклад, заробітну плату у вакансії). Функція активації в цьому шарі зазвичай відсутня або є лінійною, що дозволяє моделі передбачати неперервні числові значення. Ця особливість дизайну мережі важлива для вирішення задач регресії, де вихідні дані є неперервними змінними, а не категоріями, як у класифікації. Наприклад, якщо створюється НМ для прогнозування заробітної плати за даними ІТ-вакансій, то останній шар у нашій НМ буде складатися з одного нейрона, оскільки потрібно передбачити одне числове значення – заробітну плату. Функція активації для цього нейрона буде лінійною або взагалі відсутня, що дозволить моделі

передбачати реальні числові значення заробітної плати, які можуть змінюватися в широкому діапазоні.

Код Python створення НМ для задачі регресії з прогнозування заробітної плати наведений у додатку А (лістингА.15). У цьому коді Dense(1) на вихідному шарі вказує, що модель повинна передбачити одне числове значення – заробітну плату. Функція втрат mean_squared_error використовується для оцінки точності прогнозувань моделі.

Прогнозуванню часових рядів із застосуванням НМ властиві такі особливості [48]:

- секвенційні вхідні дані – це тип даних, де порядок елементів має значення. У контексті аналізу ринку ІТ-вакансій, прикладом секвенційних даних може бути часовий ряд кількості вакансій або зміни рівня заробітних плат протягом певного періоду. В таких даних порядок точок (відповідно до часу) є критичним для аналізу тенденцій та здійснення прогнозів;

- застосування рекурентних НМ (RNN), що є найбільш ефективними для прогнозування в часових рядів, оскільки вони можуть враховувати попередню інформацію у послідовності. Одним із прикладів застосування рекурентних НМ (RNN) у контексті аналізу ринку ІТ-вакансій може бути прогнозування зміни попиту на певні технологічні навички. RNN можна використовувати для аналізу часових рядів, які показують, як часто певні навички згадуються в описах вакансій протягом часу, щоб передбачити майбутні тренди в технологічних вимогах. Приклад коду на Python, який використовує RNN для прогнозування зміни попиту на певні технологічні навички за даними ІТ-вакансій представлений у додатку А (лістингА.16); у коді використовується SimpleRNN для моделювання часового ряду попиту на певну технологію у сфері ІТ;

- використання LSTM або GRU (Gated Recurrent Unit) – це різновиди RNN, які краще справляються з довготривалими залежностями у даних [44, 49, 50]. Приклад коду на Python з використанням LSTM та GRU

представлений у додатку А (лістингА.17). У цьому коді використовується комбінація LSTM та GRU для створення моделі глибокого навчання.

Проблєма викидів та шуму у часових рядах. Часові ряди часто містять викиди та шум, що може ускладнити прогнозування. Викиди у часових рядах – це несподівані, значно відмінні від інших значення даних, які можуть впливати на аналіз та прогнозування. Шум – це випадкові або нерелевантні відхилення в даних. Проблєма викидів та шуму полягає у їх здатності спотворювати результати моделі, роблячи прогнози менш точними. Її вирішення може включати фільтрацію або згладжування даних, використання методів виявлення викидів та видалення або коригування аномальних значень. У контексті аналізу ринку IT-вакансій, приклад викидів може бути раптове зростання або падіння кількості вакансій, що не відповідає загальним трендам (наприклад, через вплив економічних криз або пандемій). Шумом можуть бути випадкові коливання у даних, які не мають чіткого пояснення або не є пов'язаними з основними тенденціями на ринку.

Методи фільтрації та згладжування даних включають: ковзне середнє (MA, Moving Average), яке використовується для згладжування короткострокових коливань та виділення довготривалих трендів; експоненційне згладжування (ES, Exponential Smoothing) – схоже на ковзне середнє, але надає більшу вагу останнім спостереженням; фільтр Калмана (Kalman Filter) – складніший метод, який використовується для згладжування та прогнозування в часових рядах, враховуючи шум та інші невизначеності. У контексті аналізу ринку IT-вакансій, ці методи можуть застосовуватися для згладжування даних про кількість вакансій або заробітних плат протягом часу, щоб виявити основні тенденції та зменшити вплив випадкових коливань. Приклад коду на Python, де використовується ковзне середнє для згладжування часового ряду наведений у додатку А (лістингА.18). У цьому коді функція rolling(window=5).mean() створює рухоме середнє з вікном розміром 5, що допомагає згладити короткострокові коливання у кількості IT-вакансій.

Сезонність та тренди – важливі компоненти часових рядів у прогнозуванні ринку IT-вакансій (рис. 2.2). Важливо враховувати сезонність та довготривалі тренди у даних часових рядів.

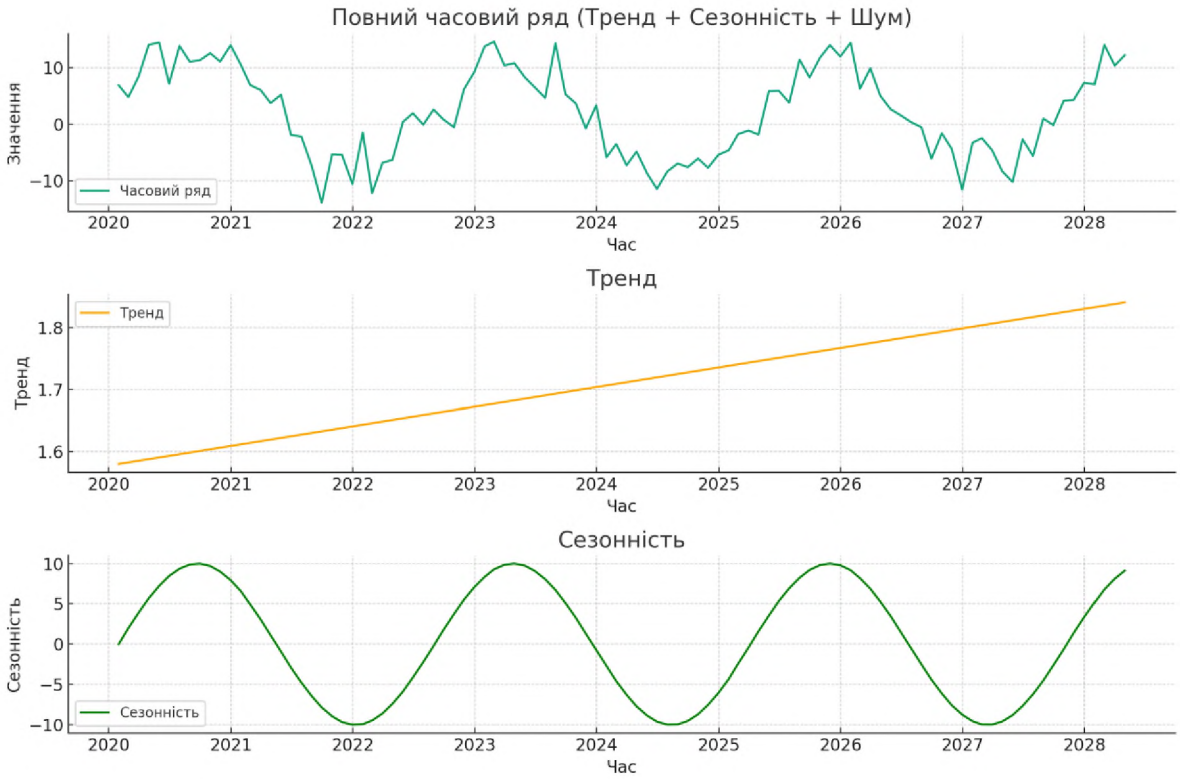



Рисунок 2.2 – Часовий ряд та його компоненти – тренд та сезонність

На графіках (рис. 2.2) представлені три основні компоненти часового ряду:

- повний часовий ряд (тренд + сезонність + шум) – це загальна картина, яка включає всі компоненти – тренд, сезонність та шум, що взаємодіють, формуючи загальні патерни у часовому ряді;

- тренд – показує лінійне зростання значень у часовому ряді, яке представляє довгострокову тенденцію;

- сезонність – цей графік показує регулярні коливання, які повторюються через певний час. У даному випадку це синусоїдальна крива, яка ілюструє явище сезонності.

Сезонність – регулярні, повторювані зміни в часовому ряді, які пов'язані з певними періодичними факторами. Наприклад, у IT-галузі сезонність може відображати звичайні пікові періоди попиту на певні навички, такі як весняний пік вакансій у розробці веб-додатків перед літом. Тренди – це довгострокові зміни в часовому ряді, що вказують на загальний напрямок розвитку. Тренди у ринку IT-вакансій можуть вказувати на зростання або зменшення загального попиту на IT-фахівців протягом років. Наприклад, якщо ми спостерігаємо, що щорічно в січні зростає кількість вакансій для Data Scientists у ринку IT, це може бути сезонністю, що відображає попит на аналітичних спеціалістів перед річницею аналізу даних попереднього року. Але, якщо кількість таких вакансій з року в рік зростає, це може вказувати на тренд у зростанні попиту на Data Scientists.

Отже, часовий ряд – це послідовність даних, зібраних у різні часові моменти, зазвичай через рівні проміжки часу. Тренд – це довгострокове зростання або падіння значень у часовому ряді. Тренд може бути лінійним (постійний ріст або зниження) або не лінійним (змінні швидкості росту або спаду). Сезонність – це регулярні коливання в часовому ряді, які повторюються з певною частотою, наприклад, щорічно, щоквартально або щомісячно. Такі коливання часто спостерігаються у даних, пов'язаних з погодою, туризмом чи роздрібними продажами. Шум – це випадкові коливання у часовому ряді, які не піддаються поясненню або передбаченню. Шум може бути викликаний різними чинниками, включаючи вимірювальні помилки або випадкові події. Кожна з цих компонентів важлива для аналізу часових рядів, оскільки вони допомагають зрозуміти загальні тенденції та шаблони (патерни) у даних. Приклад коду на Python, який враховує сезонність та тренди у часовому ряді за допомогою бібліотеки statsmodels наведений у додатку А (лістингА.19). Цей код розбиває часовий ряд на компоненти тренд, сезонність та шум, і виводить графіки для кожної з них (рис. 2.2). Це дозволяє візуально аналізувати ці компоненти, щоб врахувати їх у подальшому прогнозуванні часового ряду.

## 2.2 Підготовка даних для навчання

Успішність застосування НМ для прогнозування ринку ІТ-вакансій значною мірою залежить від якості та адекватності вхідних даних. Підготовка даних – це важливий етап у процесі МН, який включає збір, очищення, трансформацію, та стандартизацію даних перед їх подачею у НМ. На цьому етапі важливо звертати увагу на виявлення та усунення шумів та викидів, заповнення відсутніх значень, а також на вибір методів нормалізації та стандартизації даних. Особлива увага вибору та формуванню набору характеристик, які будуть використовуватися для навчання НМ, оскільки якість та релевантність цих характеристик безпосередньо впливають на ефективність прогнозування.

Ключовими етапами підготовки даних для навчання НМ, особливо у контексті прогнозування ІТ-вакансій, є наступні [51]:

- збір даних – це можуть бути дані з різних джерел, наприклад, з веб-сайтів з вакансіями, державних статистичних служб, галузевих звітів тощо;

- очищення даних – усунення помилок, викидів та некоректних записів. Цей етап включає виявлення та коригування помилкових або несумісних даних;

- перетворення даних – зміна формату даних для забезпечення їх сумісності з моделями МН. Наприклад, перетворення категоріальних даних у числові, нормалізація масштабів тощо;

- виявлення та усунення викидів – виявлення аномальних значень, які можуть спотворити результати моделювання, і їх усунення або коригування;

- заповнення відсутніх даних – визначення стратегій для заповнення пропущених значень, таких як інтерполяція або використання середніх значень;

- вибір та інженерія характеристик, створення ознак (Feature Engineering) – вибір релевантних характеристик, які будуть використані для

навчання, та створення нових характеристик з існуючих даних, що може допомогти покращити результати прогнозування;

- розподілення даних – розділення даних на набори для навчання, перевірки та тестування, щоб оцінити ефективність моделі;

- нормалізація та стандартизація – перетворення даних до формату, який більш ефективно обробляється НМ. Це може включати масштабування даних до певного діапазону або стандартизацію розподілів.

Збір даних є фундаментальним кроком у процесі аналізу та прогнозування ринку IT-вакансій, оскільки якість та повнота даних безпосередньо впливають на точність та надійність прогнозів, отриманих за допомогою НМ. Процедура збору даних для прогнозування ринку IT-вакансій складається з наступних кроків [52]:

1. Визначення джерел даних. Для аналізу ринку IT-вакансій можна використатовувати такі джерела:

- вебсайти з працевлаштування [4, 5, 53, 54];

- професійні соціальні мережі, як LinkedIn [6];

- публічні бази даних, наприклад, статистика зайнятості від Державної служби статистики України [55];

- галузеві звіти та аналітичні дослідження [52, 56, 57].

2. Збір даних. Для отримання даних з інтернет-джерел можливі такі варіанти:

- використання API для автоматичного збору даних з вебсайтів;

- парсинг вебсторінок (web scraping) для збору даних з сайтів, де API недоступні;

- збір даних вручну, якщо автоматизація неможлива або неефективна.

Код Python для автоматизованого збору даних з вебсайтів через API, що надає інформацію про вакансії, представлений у додатку А (лістингА.20). Для роботи з API потрібен ключ API, який отримується після реєстрації на відповідному вебсайті. У коді використовується бібліотека requests для відправлення HTTP запиту до API. Відповідь сайту перетворюється у формат

JSON, який можна далі аналізувати або зберігати. Деталі API (URL та ключ) потрібно адаптувати до конкретного API. Код Python для парсингу вебсторінок, коли API недоступний наведений у додатку А (лістингА.21). Цей код використовує бібліотеки requests (для здійснення HTTP запитів) та BeautifulSoup (для обробки HTML контенту). BeautifulSoup використовується для аналізу HTML сторінки та вибірки потрібних даних. У наведеному коді потрібно адаптувати селектори (у цьому випадку теги та класи HTML), щоб вони відповідали структурі сайту, з якого будуть збиратись дані.

Ручний збір даних є трудомістким процесом, але іноді це єдиний спосіб отримати необхідні дані, особливо коли доступ до автоматизованих методів обмежений або неможливий. Збір даних вручну, коли автоматизація неможлива або неефективна, організується наступним чином [52]:

- визначення джерел даних – це можуть бути вебсайти з вакансіями, професійні соціальні мережі, публічні бази даних тощо;

- розробка методології збору даних – отрібно встановити, які дані потрібно збирати (наприклад, назва вакансії, компанія, розміщення, вимоги до кандидатів) та як їх фіксувати (наприклад, у таблиці Excel або Google Sheets);

- ручний збір даних – відкрити кожне джерело і вручну перенести необхідну інформацію у свою таблицю;

- перевірка даних – після збору даних слід перевірити отримані дані на наявність помилок та/або пропусків;

- структурування даних – порядкування зібраних даних, підготовка їх для подальшої обробки та аналізу;

- документування процесу збору даних – записати методологію та кроки, які використовувались під час збору даних, щоб забезпечити прозорість та можливість відтворення даних у майбутньому.

3. Первинна обробка зібраних даних. Первинна обробка може включати перетворення даних із формату збору (наприклад, HTML, JSON) у структурований формат (таблиці в форматі CSV або бази даних).

Код Python, який перетворює дані з формату JSON у таблицю формату CSV представлений у додатку А (лістингА.22). У цьому коді дані з JSON файлу конвертуються у CSV файл. Для використання в іншій задачі потрібно адаптувати назви стовпців (fieldnames) та структуру запису даних (writer.writerow) згідно з конкретною структурою файлу JSON.

4. Перевірка якості даних. Процедура перевірки якості даних включає перевірку на відсутність дублікатів, виправлення помилок, відповідність даних фактичним умовам ринку. Код Python для перевірки якості даних, зібраних для аналізу ринку IT-вакансій представлений у додатку А (лістинг А.23). У цьому коді для обробки даних використовується бібліотека pandas.

5. Документування процесу збору даних – фіксація деталей процедури збору даних, зокрема, використаних методів, часу збору, обробки даних, відомостей про будь-які проблєми, що виникли під час збору даних. Документування дозволяє, за потреби, зрозуміти та відтворити процес збору даних, тобто допомагає забезпечити прозорість та відтворюваність дослідження. Основні кроки та правила документування ручного збору даних включають [52]:

- визначення формату документування – вибрати, у якому форматі буде вестись документація (наприклад, Google Docs, Word, Excel);

- запис інформації про джерела даних – фіксуються URL вебресурсів, назви друкованих матеріалів, інші джерела, з яких збиралися дані;

- опис процедури збору даних – слід ретельно описувати, які кроки були зроблені під час збору даних. Наприклад: «Відвідування вебсайту example.com та копіювання інформації про вакансії в таблицю Excel»;

- деталізація зібраних даних – якщо це можливо, потрібно вказувати конкретні дані, які були зібрані. Наприклад: «Зібрано дані про назву вакансії, компанію, місце розташування, вимоги до кандидатів»;

- фіксація дати та часу збору даних;

- запис усіх помічених аномалій або особливостей – якщо під час збору даних було помічено щось незвичайне, наприклад, відсутність даних або несподівані відмінності, важливо це зафіксувати;

- збереження документації – зберігати документацію в надійному місці, щоб вона була доступна для перевірки та використання в майбутньому.

## 2.3 Вибір функції втрат та оптимізаційного алгоритму

Функція втрат у контексті МН та НМ – це спосіб вимірювання різниці між прогнозами моделі та фактичними даними. Вона використовується під час навчання моделі для оцінки того, наскільки добре модель виконує своє завдання. Метою навчання моделі є мінімізація функції втрат, що дозволяє зробити прогнози моделі якомога ближчими до реальних даних. Правильний вибір відповідної функції втрат є дуже важливим для ефективного навчання моделі.

Вибір функції втрат та оптимізаційного алгоритму для НМ, яка аналізує ринок ІТ-вакансій, залежить від типу задачі (класифікація, регресія, прогнозування) та специфіки даних. Для задач регресії (наприклад, прогнозування кількості вакансій) можна використовувати середньоквадратичну помилку (MSE) або середню абсолютну помилку (MAE). Для задач класифікації (наприклад, визначення категорій вакансій) – крос-ентропію. Для задач прогнозування, які є типом задачі регресії, найчастіше використовуються наступні функції втрат: MSE, яка вимірює середнє квадратичне відхилення прогнозованих значень від фактичних, підходить для більшості задач регресії; MAE, що вимірює середнє абсолютне відхилення прогнозованих значень від фактичних, є менш чутливою до викидів порівняно з MSE. Вибір функції втрат також залежить від конкретних особливостей вихідних даних та задачі. Наприклад, якщо

важливо більше штрафувати за великі помилки, то краще вибрати MSE. Якщо ж у вихідних даних є викиди, то краще обрати MAE.

Вибір оптимізаційного алгоритму для побудови НМ [59]. Adam – найбільш популярний оптимізаційний алгоритм у навчанні НМ, який поєднує переваги двох інших методів оптимізації: AdaGrad і RMSProp. Adam добре підходить для великих наборів даних і різноманітних параметрів. Алгоритм SGD (Stochastic Gradient Descent) корисний у випадках, коли потрібна більша керованість і можливість точного налаштування. Також існують багато інших оптимізаційних алгоритмів.

При виборі функції втрат та оптимізаційного алгоритму проводять численні експерименти та перевірку, щоб знайти найкращі параметри для конкретної задачі.

## 2.4 Навчання моделі

Процедура навчання НМ залежить від конкретних вимог та особливостей задачі, і включає наступні кроки [60]:

1. Підготовка даних – підбір та обробка даних для тренування, включаючи очищення, нормалізацію, розділення на тренувальний, валідаційний та тестовий набори;

2. Вибір архітектури мережі – визначення кількості шарів та нейронів, типів активаційних функцій тощо. Вибір кількості шарів та нейронів у НМ залежить від складності задачі та обсягу даних. З цього питання не має чітко визначених правил, існують лише загальні рекомендації [61]:

- починати з простої моделі із невеликою кількістю шарів та нейронів, а потім збільшувати складність мережі за потреби;

- глибокі мережі із багатьма шарами використовувати для складних задач, таких як обробка зображень або мова;

- уникати перенавчання. Велика кількість нейронів може призвести до перенавчання, особливо якщо даних для навчання недостатньо;

- експерименти та валідація. Використовувати валідаційний набір даних для експериментів з різними архітектурами, щоб знайти оптимальну кількість шарів та нейронів;

- використання галузевих настанов. Дотримуватись рекомендацій настанов та практик, які прийняті у конкретній галузі;

Функція активації в НМ – математична операція, яка приймає сигнал від одного нейрона та визначає, наскільки активним має бути наступний нейрон. Вона вводить нелінійність у модель, що дозволяє НМ вчитися на складних даних та виконувати більш складні завдання, ніж просте лінійне відображення [62]. Використовують декілька різних функцій активації: softmax, сигмоїд (sigmoid), гіперболічний тангенс (tanh), ReLU (Rectified Linear Unit), а також їх модифікації та комбінації. На малюнку (рис. 2.3) показано графік функції активації sigmoid. Значення цієї функції змінюється від 0 до 1, а її форма нагадує логістичну криву.

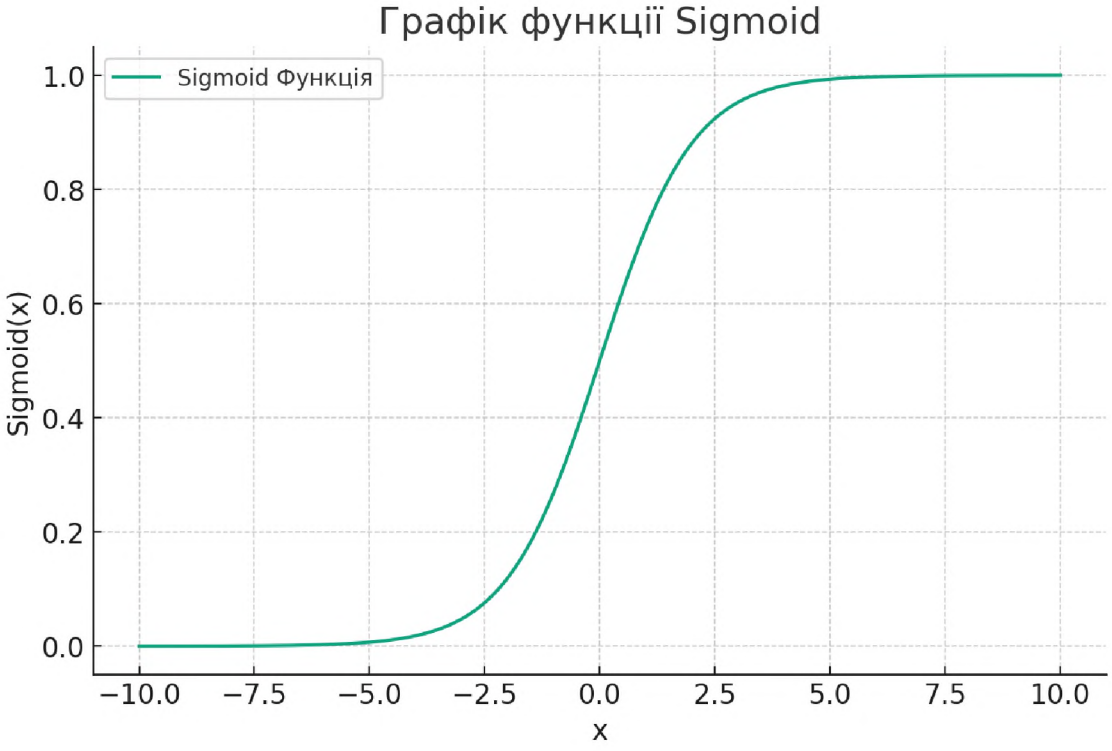



Рисунок 2.3 – Графік функції активації Sigmoid

Вибір функції активації у НМ залежить від специфіки задачі та властивостей самої функції: сигмоїд (Sigmoid) – часто використовується у вихідному шарі для задач бінарної класифікації, оскільки набуває значення між 0 та 1 (рис. 2.3); гіперболічний тангенс (Tanh) – подібний до сигмоїду, але набуває значення у діапазоні від -1 до 1, часто використовується у прихованих шарах НМ (рис. 2.4); ReLU (Rectified Linear Unit) – найпопулярніша функція активації для прихованих шарів у глибоких НМ через свою ефективність та простоту обчислень (рис. 2.5).

На малюнку (рис. 2.4) представлено графік функції активації Tanh. Ця функція також має сигмоїдальну форму, але вона змінюється від -1 до 1, на відміну від функції Sigmoid.

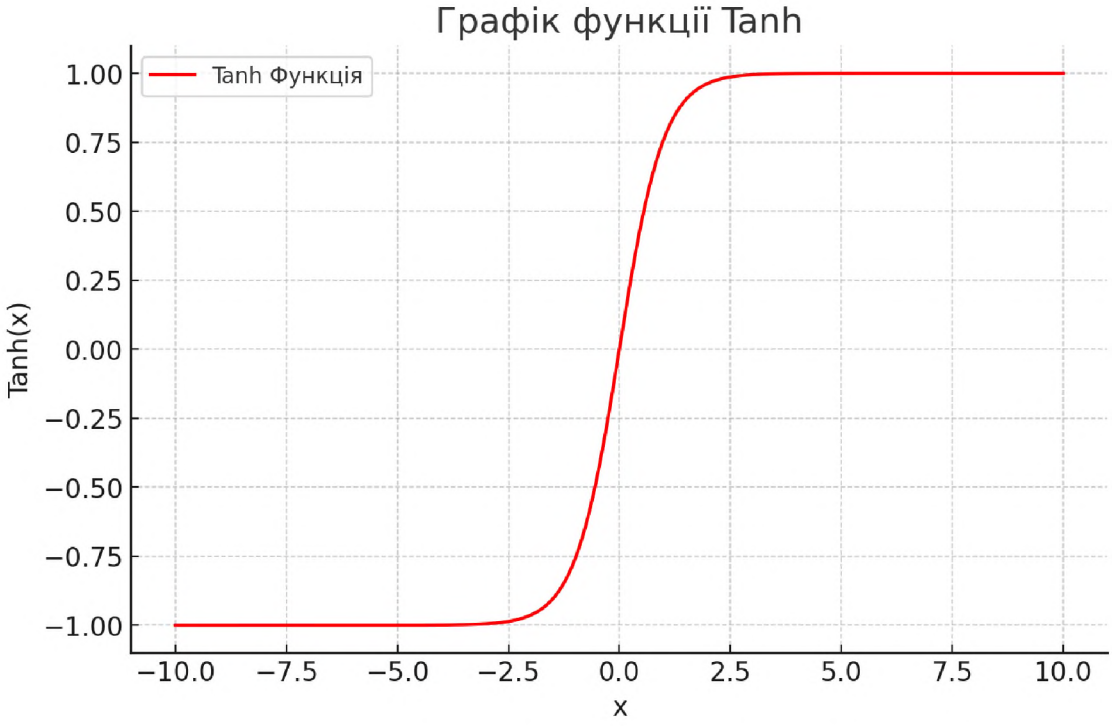



Рисунок 2.4 – Графік функції активації Tanh

На малюнку (рис. 2.5) представлено графік функції активації ReLU (Rectified Linear Unit). Ця функція є нулем для від'ємних значень x та лінійною для додатних значень x.

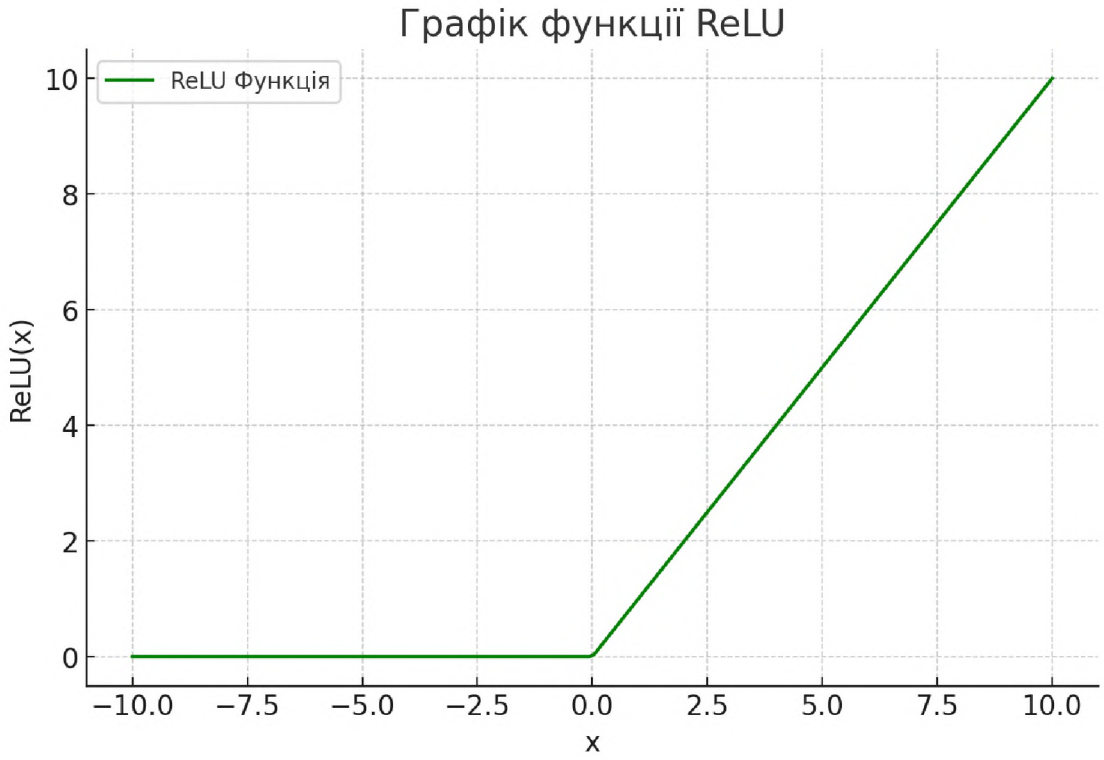



Рисунок 2.5 – Графік функції активації ReLU

На малюнку (рис. 2.6) зображено функції Sigmoid, Tanh та ReLU в одній системі координат, що дозволяє порівняти їх візуально.

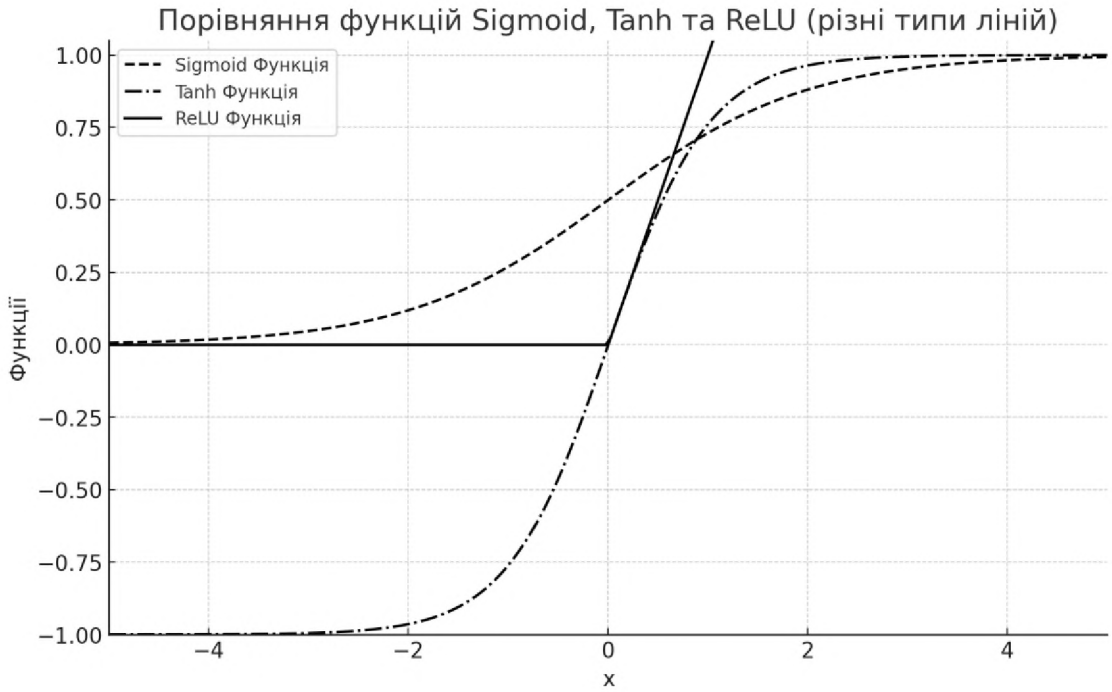



Рисунок 2.6 – Порівняння функцій активації Sigmoid, Tanh і ReLU

Загалом, вибір функції активації залежить від потреби у нелінійності, швидкості навчання, та здатності уникати проблєми зникаючих градієнтів у НМ, яка полягає в тому, що під час зворотного розповсюдження помилки (backpropagation) градієнти (похідні функції втрат відносно ваг) можуть ставати дуже маленькими [63]. Це особливо актуально у глибоких НМ із багатьма шарами. Коли градієнти занадто малі, вони не здатні ефективно оновлювати ваги нейронів, що призводить до дуже повільного навчання мережі або навіть до зупинки її навчання. Ця проблєма часто виникає при використанні сигмоїдальних або тангенціальних функцій активації, оскільки їх похідні можуть приймати маленькі значення. ReLU часто використовується для глибоких мереж, оскільки вона допомагає уникнути цієї проблєми.

3. Ініціалізація ваг – задання початкових значень вагам НМ, це процес присвоєння початкових значень вагам нейронів перед початком тренування мережі. Правильна ініціалізація ваг є важливою для ефективного та стабільного навчання НМ.

Часто ваги ініціалізують малими випадковими числами, що допомагає уникнути симетрії та забезпечує ефективне навчання. Також, використвується He або Xavier ініціалізація. Ці спеціальні методи ініціалізації враховують кількість входів та виходів у шарі для забезпечення оптимального розподілу ваг [64, 65].

Код Python з використанням бібліотеки Keras, де використовується ініціалізація ваг, представлений у додатку А (лістингА.24). У цьому коді використана ініціалізація He для шару з активацією ReLU: використовується функція HeNormal() як ініціалізатор ваг для шарів, де активаційна функція ReLU.

Іноді ваги можуть ініціалізуватися нулями або одиницями, хоча це не рекомендується для глибоких мереж, оскільки може призвести до проблєм у навчанні НМ.

4. Вибір функції втрат та оптимізаційного алгоритму – це вибір відповідних методів для оцінки помилки та оновлення ваг. Оновлення ваг нйеронів у НМ після оцінки помилки результату відбувається за допомогою процесу, який називається зворотним розповсюдженням помилки (backpropagation). Цей процес є основою навчання НМ та дозволяє моделям адаптуватися та покращувати свої прогнози на основі навчальних даних.

Алгоритм процесу оновлення ваг включає наступні кроки:

- пряме розповсюдження (forward propagation) – НМ отримує вхідні дані та передає їх через шари, використовуючи поточні ваги, до вихідного шару, де робиться прогноз;

- оцінка помилки (error estimation) – обчислюється помилка моделі, порівнюючи прогнози моделі з фактичними значеннями. Для цього використовується функція втрат;

- зворотне розповсюдження (backpropagation) – помилка передається назад по мережі. Під час цього процесу визначаються частинні похідні функції втрат відносно кожного ваги (тобто обчислюється градієнт помилки);

- оновлення ваг – ваги оновлюються з метою зменшення помилки. Оновлення ваг у НМ виконується автоматично за допомогою оптимізаційних алгоритмів під час навчання мережі. Розмір кроку оновлення визначає швидкість навчання. У додатку А наведений приклад коду на Python з використанням бібліотеки Keras, де відбувається процес тренування моделі, що включає в себе оновлення ваг (лістингА.25). У цьому коді модель компілюється з використанням оптимізатора Adam, який автоматично регулює ваги під час процесу тренування. Функція fit використовується для тренування моделі на навчальних даних (x_train, y_train) протягом визначеної кількості епох. Під час кожної епохи відбувається пряме розповсюдження, обчислення помилки, зворотне розповсюдження помилки та оновлення ваг.

- повторення процесу – процес прямого та зворотного розповсюдження повторюється протягом багатьох ітерацій (епох), доки мережа не навчиться зменшувати помилку до прийнятного рівня.

5. Навчання (тренування) мережі – це подача тренувальних даних до мережі та її тренування з метою мінімізації функції втрат, що відбувається наступним чином:

- подача даних – навчальний набір даних подається на вхід НМ. Дані зазвичай подаються пакетами (batch), що є типовим підходом для великих наборів даних;

- пряме розповсюдження (forward propagation) – дані проходять через мережу, від вхідного шару до вихідного. Кожен нейрон у кожному шарі обчислює свій вихідний сигнал на основі вхідних даних, ваг і функції активації;

- обчислення помилки – на вихідному шарі розраховується помилка мережі, яка є різницею між прогнозованим виходом та фактичними даними. Для цього використовується функція втрат;

- зворотне розповсюдження (backpropagation) – помилка передається назад по мережі, обчислюючи градієнти функції втрат для ваги кожного нейрону. Таким чином визначається, як потрібно змінити ваги, щоб зменшити помилку;

- оновлення ваг – використовуюється певний алгоритм оптимізації (наприклад, градієнтний спуск), ваги мережі коригуються на основі обчислених градієнтів. Швидкість, з якою ваги оновлюються, визначається параметром швидкості навчання;

- повторення процесу – наведені вище кроки повторюються протягом багатьох ітерацій або епох навчання. На кожній ітерації НМ вчиться зменшувати помилку, що підвищує точність прогнозування;

- валідація – відбувається паралельно з тренуванням. НМ також може бути перевірена на валідаційному наборі даних, щоб оцінити її загальну ефективність і уникнути перенавчання. Процес тренування закінчується, коли НМ досягає заданого рівня точності або після визначеної кількості епох;

6. Тестування – перевірка ефективності моделі на валідаційному та тестовому наборах даних.

7. Налаштування гіперпараметрів – зміна параметрів НМ (наприклад, швидкість навчання, кількість епох) для підвищення точності моделі.

На графіку (рис. 2.7) представлений типовий процес навчання НМ. Цей графік показує, як змінюється помилка результату з кожною епохою навчання.

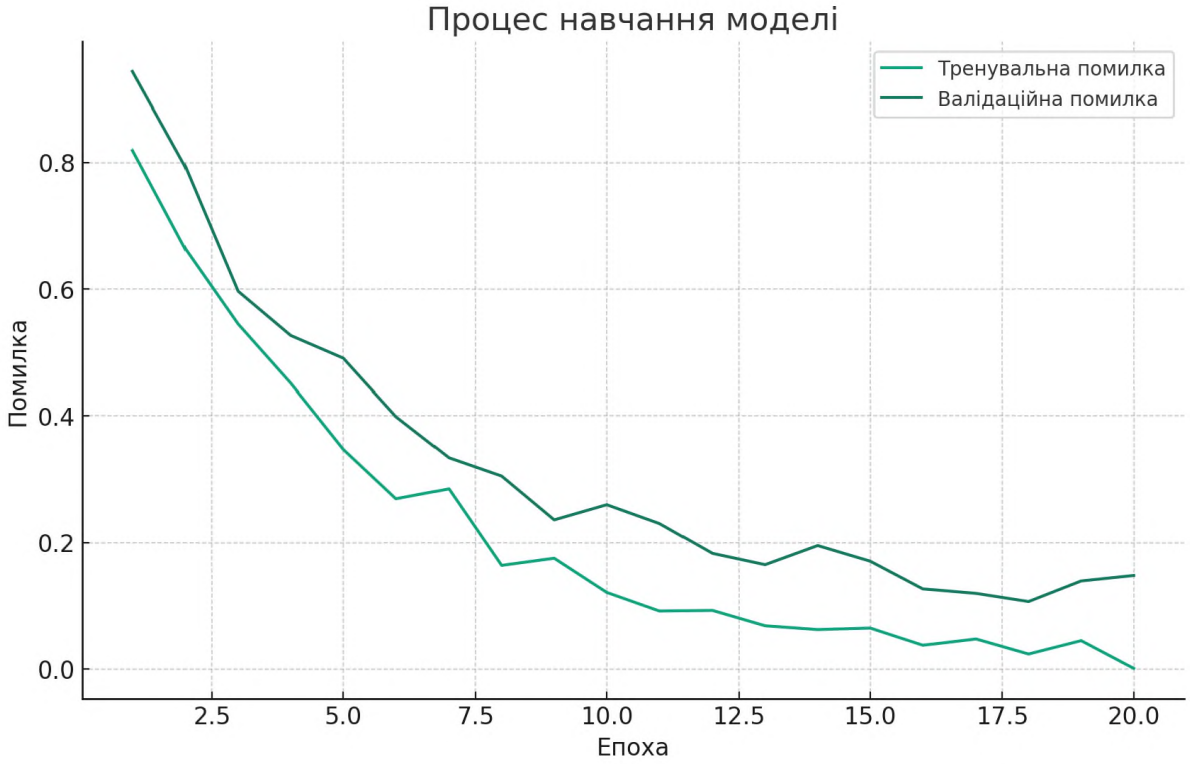



Рисунок 2.7 – Процес навчання моделі

Як видно із графіка (рис. 2.7), тренувальна помилка моделі (Training Loss) зменшується з кожною епохою, що свідчить про те, що модель поступово вчиться на навчальних даних; валідаційна помилка (Validation Loss) – ця крива показує, як модель поводиться на даних, які не були використані під час тренування. Зазвичай, валідаційна помилка також зменшується, але іноді може спостерігатися розбіжність з тренувальною помилкою, що може вказувати на перенавчання. У такому випадку потрібно застосувати відповідні методи, що запобігають перенавчанню, дозволяють його уникнути.

## 2.5 Аналіз результатів та оптимізація гіперпараметрів

Підгонка моделі є ключовим етапом у процесі МН, що передбачає не лише навчання моделі на доступних даних, але й глибокий аналіз отриманих результатів. Цей етап включає в себе оцінку ефективності моделі, виявлення та аналіз помилок, а також оптимізацію гіперпараметрів для підвищення точності та надійності прогнозувань. Для оцінки ефективності результатів прогнозування ринку ІТ-вакансій за допомогою НМ використовують такі методи [66]:

1. Розділення даних на тренувальний, валідаційний та тестовий набори. Тренувальний набір використовується для навчання моделі, валідаційний – для налаштування гіперпараметрів, тестовий – для остаточної оцінки ефективності моделі;

2. Метрики оцінки. Для задач регресії (наприклад, прогнозування кількості вакансій) використовують середньоквадратичну помилку (MSE) або середню абсолютну помилку (MAE); для задач класифікації використовують точність (accuracy), F1-метрику, матрицю помилок;

3. Перехресна перевірка (Cross-Validation) ввикористовується, щоб переконатись, що модель добре узагальнює дані. Вона дозволяє оцінити стабільність моделі на різних піднаборах даних. Приклад коду на Python з використанням бібліотеки Scikit-Learn для реалізації перехресної перевірки (cross-validation) представлений у додатку А (лістингА.26). У цьому коді використовується модель RandomForestClassifier на датасеті Iris. Функція cross_val_score виконує перехресну перевірку, розділяючи датасет на 5 частин (фолдів), кожен раз тренуючи модель на 4 з них і тестуючи на 1. Результати перевірки надають уявлення про те, наскільки добре модель узагальнює дані;

4. Аналіз помилок – проводиться, коли модель робить помилки. Наприклад, це може бути аналіз конкретних випадків неправильної класифікації або аналіз розподілу помилок за різними категоріями;

5. Візуалізація результатів за допомогою графіків та діаграм. Наприклад, будують криві навчання, що показують зміну помилки протягом часу, або матрицю помилок для задач класифікації;

6. Порівняння з базовою моделлю (Baseline). Можна порівняти результати моделі з простою базовою моделлю, щоб визначити, наскільки значними є покращення, яке вона пропонує. Наприклад, якщо модель прогнозує кількість IT-вакансій у різних містах, то можна порівняти її прогнози з реальними даними за певний період, використовуючи функції втрат MSE або MAE, а також оцінити, наскільки добре модель описує періоди з максимально великою або мінімальною кількістю вакансій. У додатку А (лістингА.27) наведений код на Python, який порівнює результати НМ з базовою моделлю за допомогою бібліотеки Scikit-Learn. У цьому коді використовується датасет Iris. Базова модель DummyClassifier прогнозує клас, який зустрічається найчастіше. НМ MLPClassifier навчається на тих самих даних. Точність обох моделей порівнюється на тестовому наборі даних.

Оптимізація гіперпараметрів, таких як швидкість навчання, кількість шарів та нейронів у НМ, є важливим етапом для забезпечення максимальної ефективності моделі.

Швидкість навчання (Learning Rate) – параметр, який визначає, наскільки великими будуть кроки оновлення ваг під час навчання. Занадто велика швидкість навчання може призвести до того, що модель «перескочить» через оптимальне рішення, а занадто мала – до повільного навчання або «застрягання» в локальному мінімумі. Зазвичай, спочатку встановлюється деяке середнє значення швидкості навчання (наприклад, 0.01). Далі його коригують в залежності від поведінки моделі під час тренування. Зазвичай, при високій швидкості навчання спостерігається швидке зниження помилки на початкових етапах навчання, але потім можливе коливання навколо мінімального значення помилки без стабілізації. При середній швидкості навчання помилка знижується більш плавно і

стабільно. Цей варіант вважається оптимальним, оскільки він забезпечує баланс між швидкістю навчання та точністю моделі. При низькій швидкості навчання характерне повільне зниження помилки. Хоча це зменшує ризик «перестрибування» оптимального рішення, навчання НМ може бути надмірно повільним та можливе «зависання» у локальних мінімумах. Такий аналіз дозволяє визначити, яка швидкість навчання є найбільш ефективною для конкретної задачі та набору даних, як у випадку з прогнозуванням ринку IT-вакансій.

Кількість нейронів в шарі визначає «ширину» цього шару. Більше нейронів дозволяє моделі виявляти більш складні закономірності в даних. Однак, як і у випадку з кількістю шарів, збільшення кількості нейронів може призвести до перенавчання і збільшує час тренування моделі (табл. 2.1).

Таблиця 2.1 – Загальні характеристики НМ залежно від їх глибини (кількість шарів) та ширини (кількість нейронів у шарі)

| Характеристика | Мало шарів, мало нейронів | Мало шарів, багато нейронів | Багато шарів, мало нейронів | Багато шарів, багато нейронів |
|---|---|---|---|---|
| Здатність до узагальнення | Може бути недостатньою для складних задач | Краще, але може не вловлювати всі складні залежності | Добра, здатна виявляти складні залежності | Дуже висока, але із ризиком перенавчання |
| Ризик перенавчання | Низький | Помірний | Помірний | Високий |
| Обчислювальна складність | Низька | Висока | Висока | Дуже висока |
| Швидкість навчання | Висока | Середня | Середня | Повільна |
| Придатність до використання | Для простих задач | Для простих і деяких складніших задач | Для складних задач з малою кількістю даних | Для складних задач з великою кількістю даних |

Кількість шарів у НМ визначає її «глибину». Глибокі мережі зазвичай можуть моделювати більш складні залежності у даних. Збільшення кількості шарів може приводити до перенавчання і збільшує ризик зникаючих або вибухових градієнтів. Варто починати з меншої кількості шарів і збільшувати

їх кількість у разі потреби, зберігаючи баланс між точністю та складністю моделі. Для малої кількості шарів характерне повільне зниження помилки. Це може свідчити про недостатню здатність моделі виявляти складні залежності в даних, що призводить до гіршої узагальнюючої здатності. Для середньої кількості шарів властиве більш швидке та зниження помилки. Це може вказувати на те, що модель достатньо складна для виявлення залежностей у даних, але не занадто складна, щоб призвести до перенавчання. При великій кількості шарів зазвичай спостерігається швидке зниження помилки на початкових етапах, але з часом може з'явитися перенавчання, особливо якщо даних недостатньо для підтримки такої складної моделі.

Оптимальна кількість нейронів визначається емпірично: потрібно поступово збільшувати їх кількість і спостерігати за результатами моделі. У випадку малої кількості нейронів у шарі спостерігається відносно повільне зниження помилки. Це вказує на те, що модель недостатньо складна для виявлення усіх важливих залежностей у даних. Якщо кількість нейронів у шарі оптимальна, то має спостерігатись більш швидке та стабільне зниження помилки. Це вказує на те, що у моделі достатньо потенціалу для виявлення складних залежностей у даних і ефективного навчання. Така модель має найкращий баланс між здатністю узагальнення та уникненням перенавчання. Якщо нейронів у шарі забагато, то можливе швидке зниження помилки на початку, але потім вона стабілізується або навіть зростає. Така модель може перенавчатися, особливо якщо доступний набір даних обмежений. Загалом, висока складність моделі може призвести до того, що вона добре працює на тренувальних даних, але погано узагальнює на нових даних.

## 2.6 Перевірка моделі на тестових даних

Перевірка моделі на тестових даних є кінцевим етапом у навчанні моделі, який дозволяє оцінити її реальну ефективність та узагальнюючу

здатність. Цей етап важливий для підтвердження того, що модель не тільки добре працює на тренувальних та валідаційних даних, але й ефективно передбачає результати на нових, раніше невідомих даних. Ключовими аспектами перевірки моделі на тестових даних є акцент на методах та метриках оцінки роботи моделі. Використання тестового набору даних дозволяє оцінити загальну точність моделі, виявити можливі помилки прогнозування або випадки неправильної класифікації та зробити висновки щодо практичної придатності моделі для прогнозування ринку IT-вакансій.

Загальний алгоритм перевірки моделі на тестових даних:

1. Підготовка тестових даних. Вибирається або готується тестовий набір даних, який не використовувався під час навчання моделі. Для прогнозування ринку IT-вакансій це можуть бути дані про нові вакансії, їх розташування, рівень заробітної плати тощо;

2. Застосування моделі до тестових даних. Проводиться пряме розповсюдження тестових даних через модель, щоб отримати прогнози;

3. Оцінка результатів. Для задач регресії (наприклад, прогнозування кількості вакансій) використовують такі метрики оцінки, як MSE або MAE. Для задач класифікації (наприклад, класифікація спеціалізацій вакансій) використовуються точність (accuracy), F1-метрика, матриця помилок.

Зокрема, матриця помилок допомагає виявити слабкі місця та напрямки удосконалення. Розглянемо матрицю помилок для класифікації IT-вакансій за рівнями (наприклад, Junior, Middle, Senior). Припустимо, що були отримані такі результати класифікації:

істинні позитиви (True Positives, TP) – модель правильно класифікувала 30 Junior, 40 Middle та 35 Senior вакансій;

істинні негативи (True Negatives, TN) – модель правильно визначила, що вакансії не належать до певної категорії (наприклад, 100 випадків, коли вакансія не Junior, правильно класифіковані);

хибні позитиви (False Positives, FP) – модель помилково визначила 10 вакансій як Junior, які насправді були Middle або Senior;

хибні негативи (False Negatives, FN) – модель не визнала 5 Junior, 10 Middle та 10 Senior вакансій, класифікувавши їх неправильно.

У цьому випадку, матриця помилок буде виглядати наступним чином (табл. 2.2).

Таблиця 2.2 – Приклад матриці помилок у задачі класифікації

| Класифікація \ Істинна мітка | Junior | Middle | Senior |
|---|---|---|---|
| Junior | 30 | 5 | 5 |
| Middle | 5 | 40 | 10 |
| Senior | 0 | 5 | 35 |

Діагональні значення матриці помилок (30, 40, 35) показують кількість правильно класифікованих вакансій для кожної категорії. Значення поза діагоналлю (5, 10, 5...) показують помилки класифікації. Наприклад, 5 вакансій, які насправді були Junior, помилково були класифіковані як Middle.

За цією матрицею оцінити точність, повноту та F1-метрику для кожної категорії, що зрозуміти, у яких категоріях модель працює краще або гірше. Для оцінки точності, повноти та F1-метрики на основі матриці помилок, використовуються наступні показники:

- точність (Precision) показує, яка частка вакансій, класифікованих як певний рівень (наприклад, Junior), дійсно є цим рівнем, $P = TP/(TP + FP)$;

- повнота (Recall) – показує, яка частка вакансій певного рівня (наприклад, Junior) була правильно виявлена моделлю, $R = TP/(TP + FN)$;

- F1-метрика – це гармонійне середнє між точністю P та повнотою R, що дає баланс між ними, $F1 = 2PR/(P + R)$.

Розрахунки точності, повноти та F1-метрики для категорії Junior:

- точність: $P = 30/(30 + 5 + 5) = 0.75$;

- повнота: $R = 30/(30 + 5 + 5) = 0.75$;

- F1-метрика: $F1 = 2*0.75*0.75/(0.75 + 0.75) = 0.75$.

Аналогічні розрахунки можна виконати для категорій Middle та Senior. Порівняння P, R та F1 для різних категорій, дозволяє визначити, в яких

категоріях модель має вищу точність чи повноту. Наприклад, якщо F1-метрика більша для Junior, ніж для Middle, це може означати, що модель краще виявляє вакансії категорії Junior. Аналізуючи, де модель має високі FP або FN, можна виявити, у яких категоріях модель частіше робить помилки. Отже, метрики допомагають зрозуміти, як модель працює з різними категоріями та які аспекти моделі потребують покращення.

4. Аналіз помилок – це аналіз випадків, де модель зробила невірні прогнози щоб визначити, чи існують певні закономірності чи фактори, що впливають на помилки. Аналіз помилок дозволяє поліпшити точність моделі та зрозуміти її обмеження та особливості роботи з конкретними типами даних. Це допомагає виявити причини неправильних прогнозів та поліпшити модель, зокрема, оцінити, наскільки добре модель працює у реальних умовах та яку цінність вона може принести для рішення задачі прогнозування ринку IT-вакансій. Послідовність аналізу помилок:

- збір помилок – виявлення випадків, де модель зробила помилки. Наприклад, у задачі класифікації IT-вакансій, це випадки, де модель помилково класифікувала рівень вакансії (наприклад, помилково визначила Junior вакансію як Senior);

- категоризація помилок – систематизація помилок за типами. Наприклад, помилки можуть бути зумовлені неправильними даними, некоректними мітками або особливостями даних, які модель не враховує;

- глибокий аналіз – аналіз кожного випадку помилки, щоб зрозуміти її причини. Наприклад, варто з'ясувати, чи є помилки, що зосереджені у певних категоріях, чи є якісь спільні характеристики даних, які призводять до помилок тощо.

- візуалізація помилок за допомогою графіків та діаграм розподілу помилок. Діаграма візуалізації помилок у наведеному вище прикладі відобразить 10 помилок у категорії Junior, 15 помилок у категорії Middle та 10 помилок у категорії Senior;

- зв'язок помилок з даними – потрібно перевірити, чи не пов'язані помилки з певними особливостями даних. Наприклад, чи частіше помилки виникають у вакансіях з певними ключовими словами або у певних регіонах тощо;

- внесення змін у модель або у підготовку даних. На основі виявлених причин помилок потрібно увести зміни у модель або процес підготовки даних. Наприклад, змінити архітектуру моделі, додати нові характеристики, використовувати інші методи передобробки даних тощо;

- тестування змін – виконується після внесення змін у модель, щоб переконатися, що якість прогнозування покращилася.

5. Удосконалення моделі – виконується на основі отриманих результатів та аналізу помилок, щоб підвищити точність прогнозування.

6. Документування (фіксація) отриманих результатів та висновків, допоможе у подальшому удосконаленні моделі або при її застосуванні на практиці.

## Висновки до розділу 2

У розділі розглянуто наступні аспекти розробки та застосування НМ для рішення задач прогнозування ринку IT-вакансій:

1. Вибір функції втрат та оптимізаційного алгоритму, що вирішальне значення для ефективного навчання моделі, що було продемонстровано через аналіз різних можливостей. Навчання моделі, у свою чергу, залежить від правильної координації цих елементів та уважного моніторингу процесу навчання.

2. Аналіз результатів та оптимізація гіперпараметрів, що включає ретельне тестування та налаштування моделі, також є надзвичайно важливими для забезпечення точності та надійності прогнозувань.

3. Перевірка моделі на тестових даних дає можливість оцінити її практичну застосовність та узагальнюючу здатність на невідомих раніше даних.

4. Підкреслено важливість знання структурних особливостей моделі для досягнення оптимальної продуктивності та важливість якісної обробки та аналізу вхідних даних.

5. У результаті проведеного аналізу встановлено, що успіх у прогнозуванні ринку IT-вакансій за допомогою НМ залежить від комплексного підходу до кожного етапу розробки та використання моделі, починаючи від ініціалізації та закінчуючи перевіркою її ефективності.

Наступний розділ буде присвячений практичному застосуванню теоретичних знань та методів, описаних вище, для прогнозування ринку IT-вакансій, включаючи опис процесу створення, тренування та НМ для аналізу та прогнозування різних аспектів ринку IT-праці.

# РОЗДІЛ 3
# ПРАКТИЧНІ АСПЕКТИ ПРОГНОЗУВАННЯ РИНКУ ІТ-ВАКАНСІЙ З ВИКОРИСТАННЯМ НЕЙРОННОЇ МЕРЕЖІ

## 3.1 Розробка та навчання нейронної мережі

Збір даних. Вихідні дані для аналізу ринку ІТ-вакансій в Україні були зібрані з українських вебсайтів, присвячених ринку праці у сфері ІТ. Збір даних для прогнозування ринку ІТ-вакансій в Україні виконувався з використанням вебскрапінгу та API з українських вебсайтів, які публікують вакансії в ІТ-сфері, збирались дані про: назви вакансій, рівень кваліфікації, заробітна плата, місцезнаходження та інші ключові параметри. Основні використані джерела:

DOU.ua (вакансії) – один з найбільших українських порталів з вакансіями в ІТ-сфері[ 4];

Djinni.co – платформа для пошуку роботи в ІТ, що зосереджується на вакансіях для розробників, тестувальників, дизайнерів [5];

LinkedIn (вакансії в Україні) – міжнародна професійна мережа, де можна знайти інформацію про ІТ-вакансії в Україні [6];

Державна служба статистики України – офіційний сайт, де публікуються дані щодо ринку праці та інші статистичні дані [55].

Сайт DOU.ua не має публічного API. У цьому випадку доцільним було використати технологію вебскрапінгу, що не порушує умов використання сайту, оскільки, відповідно до цих правил, у даній роботі зазначено посилання на джерело даних. Код Python з використанням бібліотеки requests для запитів та BeautifulSoup для парсингу HTML для збору даних з вебсайту jobs.dou.ua наведений у додатку А (лістингА.28). Для збереження отриманих результатів у файл обрано формат CSV: код збирає дані про вакансії з вебсайту і записує їх у файл vacancies.csv. Для збереження українських символів використовується кодування UTF-8. Код демонструє принцип збору

даних про вакансії. Він налаштований на автоматичний збір даних з вебсайту jobs.dou.ua, і потребує додаткових налаштувань для інших сайтів залежно від їх структури та обсягу необхідних даних. Варто враховувати, що вебскрапінг може навантажувати сервер сайту, тому слід використовувати цей метод обережно та етично.

Підготовка даних. Перед тим як використовувати отримані дані для тренування, вони мають бути очищені від шуму та невідповідностей, а також перетворені у формат, придатний для обробки НМ. Для очищення даних з файлу vacancies.csv та їх перетворення у формат, придатний для обробки НМ, потрібно було виконати видалення дублікатів, нормалізацію текстових даних (наприклад, приведення до нижнього регістру), вилучення або обробку відсутніх значень, кодування категорійних даних, якщо такі є. Код на Python, який виконує ці дії наведений у додатку А (лістингА.29).

Вибір архітектури мережі – визначення структури мережі, включаючи кількість та типи шарів, їх глибину та ширину, а також функції активації. Для прогнозування кількості IT-вакансій розроблена архітектура НМ, яка включає наступні компоненти:

- вхідний шар (Input Layer) – приймає вхідні історичні дані про кількість вакансій;

- приховані шари: (Hidden Layers) – перший прихований шар – Dense (повнозв'язний) шар з 64 нейронами та ReLU активацією, виявляє базові взаємозв'язки в даних; другий прихований шар – аналогічний першому, забезпечує додаткову обробку отриманих властивостей; Dropout Layer – застосовується для уникнення перенавчання, має коефіцієнт викидання 0.3; вихідний шар (Output Layer) – має один нейрон з лінійною активацією для регресійного прогнозування кількості вакансій;

- функція втрат – MSE, оптимізатор – Adam.

Код реалізації даної НМ використовує бібліотеку Keras (лістинг А.30). Модель налаштована для прогнозування кількості вакансій на основі набору вхідних характеристик.

Навчання моделі. Мережа була тренована на зібраних даних. Процес тренування включає налаштування гіперпараметрів, моніторинг процесу навчання та оцінку її продуктивності.

У коді представленому у додатках спочатку відбувається завантаження та підготовка даних, потім створюється та тренується НМ, виконується прогнозування та зберігаються результати. Результати прогнозування виводяться у файл, на екран у вигляді таблиці та у вигляді графіка для візуального аналізу. Спершу дані зчитуються з файлу clean_vacancies.csv, потім вони готуються для входу у НМ, і далі модель навчається. Результати прогнозу зберігаються у файл forecast_vacancies.csv, а також відображаються на екрані у вигляді таблиці та графіка (лістингА.31). Код для виводу графіку моніторингу навчання НМ наведений у додатку А (лістингА.32).

Для порівняння були створені НМ (модель 1 і модель 2). Обидві моделі – це послідовні НМ, що мають вхідний шар з 12 нейронів та функцією активації ReLU. Вихідний шар НМ має 1 нейрон, так як це вихідний шар для регресії, без активаційної функції. Компіляція моделей виконана з використанням функції втрат mean_squared_error та оптимізатора Adam.

Навчання моделей виконувалось на історичних даних щодо кількості ІТ-вакансій в Україні. Дані зібрані з відкритих джерел: вихідний набір даних (датасет) охоплює відомості про наявність ІТ-вакансій із 12 різних ІТ-спеціальностей) – щоденні дані за період з 20.05.2022 по 14.11.2023 (всього 543 періоди). Таким чином, вихідний датасет був представлений матрицею розміром 12x543 з 6516 значень, з яких: на тренувальний набір було відведено 67%, а на тестовий набір – 33%. Розмір пакетів batch size = 2, тобто ваги моделей оновлювались після кожних двох зразків даних. Головна відмінність між моделями 1 і 2 – кількість епох навчання: для моделі 1 було використано epochs=100, для моделі 2 – epochs=200.

На рис. 3.1 представлена історія навчання моделі 1 (epochs=100).


Рисунок 3.1– Історія навчання НМ (модель 1): Loss = MSE, оптимізатор Adam, epochs=100, batch_size=2.

На рис. 3.2 представлена динаміка кількості IT-вакансій в Україні за період з 20.05.2022 по 14.11.2023 та прогноз на період з 15.11.2023 по 13.01.2024, отриманий з використанням НМ (модель 1).

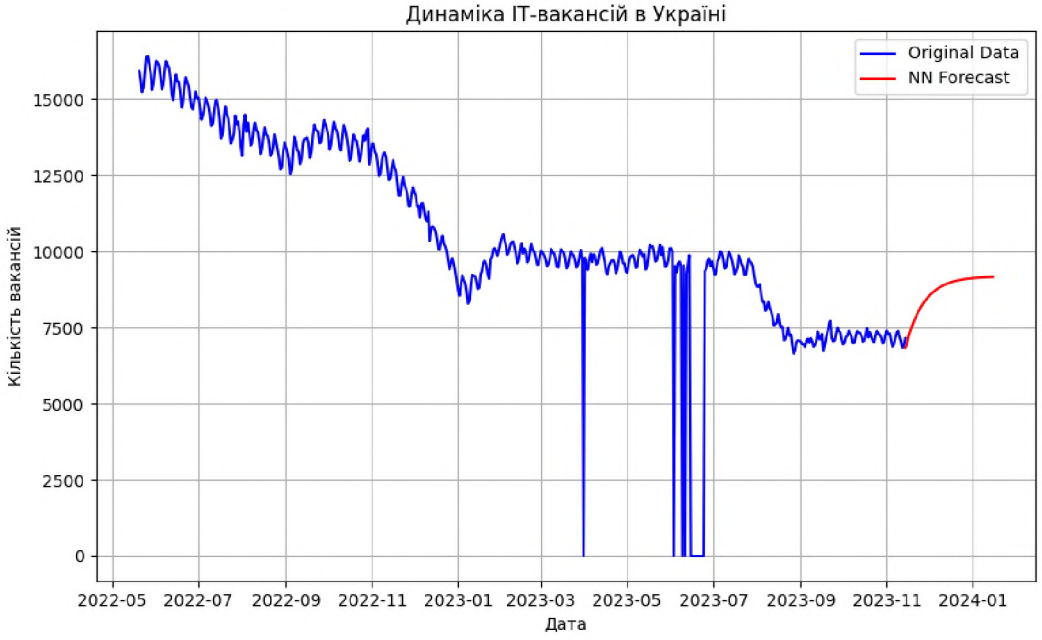



Рисунок 3.2 – Динаміка IT-вакансій в Україні з 20.05.2022 по 14.11.2023 та прогноз на період з 15.11.2023 по 13.01.2024 з використанням НМ (модель 1)

На рис. 3.3 і рис. 3.4 представлена історія навчання моделі 2 (epochs=200) та результати прогнозування за допомогою цієї моделі.

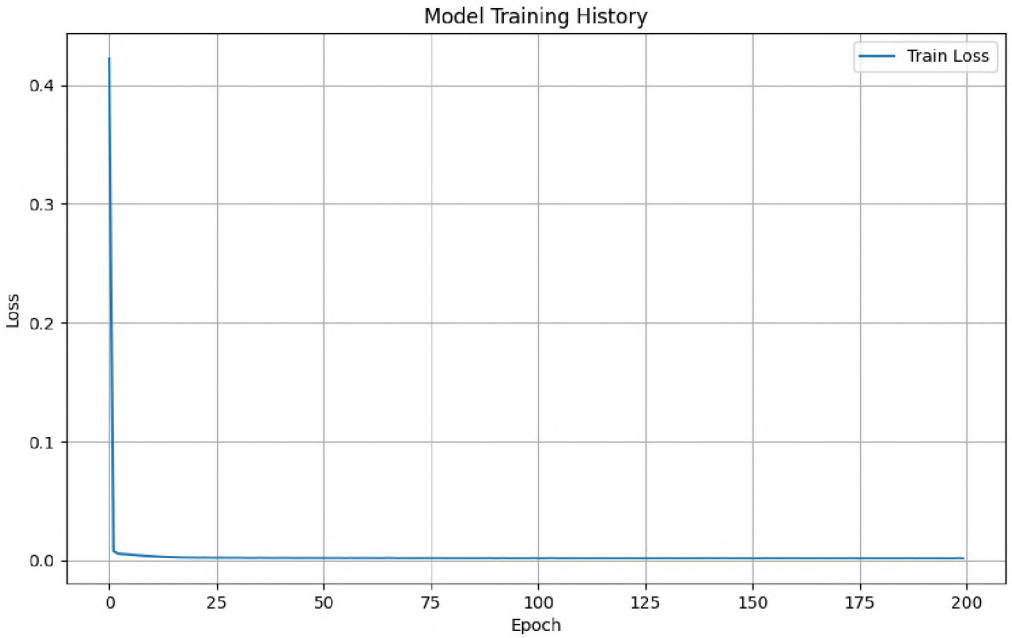



Рисунок 3.3 – Історія навчання НМ (модель 2): Loss = MSE, оптимізатор Adam, epochs=200, batch_size=2.

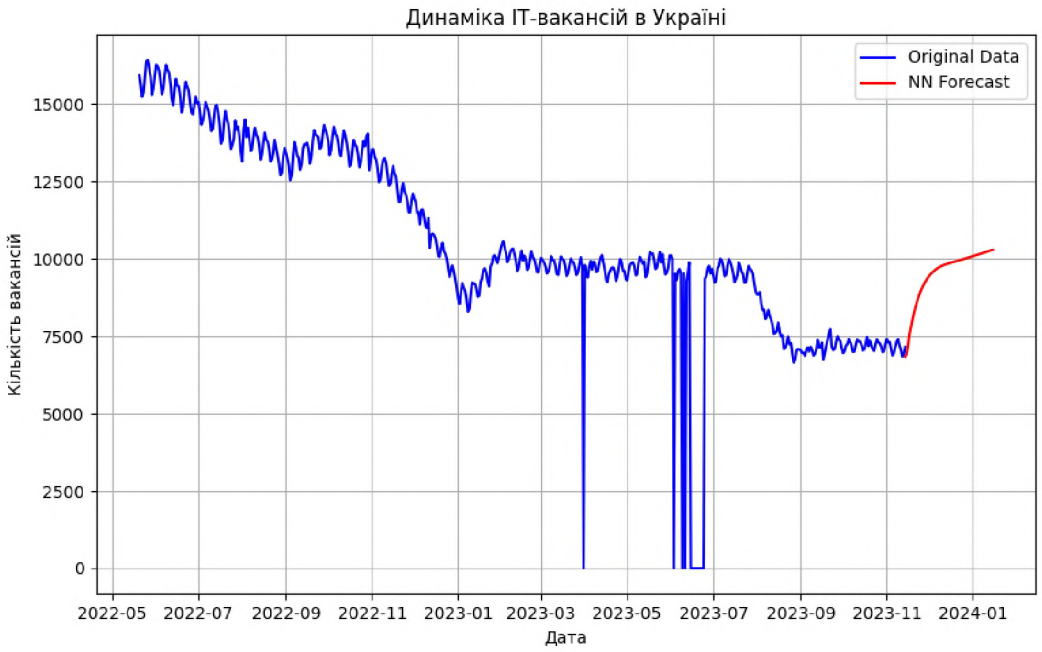



Рисунок 3.4 – Динаміка IT-вакансій в Україні з 20.05.2022 по 14.11.2023 та прогноз на період з 15.11.2023 по 13.01.2024 з використанням НМ (модель 2)

## 3.2 Оцінка результатів та метрики точності

Після навчання моделі НМ була виконана оцінка її здатності правильно прогнозувати ринок ІТ-вакансій. Були використані стандартні підходи, що використовуються для оцінювання моделей прогнозування на основі НМ. Оскільки завдання полягає у прогнозуванні кількості, а не класифікації, то використовувались такі метрики оцінки:

1. Середньоквадратична помилка (MSE, Mean Squared Error), що вимірює середнє квадратів помилок між фактичними та прогнозованими значеннями. Чим нижче MSE, тим краще;

2. Середня абсолютна помилка (MAE, Mean Absolute Error), що вимірює середнє абсолютних помилок, і є менш чутливою до великих помилок, ніж MSE;

3. Коефіцієнт детермінації ($R^2$) – відображає, яку частку варіативності залежної змінної може пояснити модель. Значення близьке до 1 свідчить про високу якість моделі. Код Python, що реалізує оцінку цих метрик представлений у додатку А (лістингА.33).

Результати виконаних розрахунків:

MSE = 150 – середнє квадратичне відхилення між фактичними та прогнозованими значеннями рівне 150;

MAE = 10 – середня абсолютна помилка прогнозу рівна 10;

$R^2 = 0.8$ – модель пояснює 80% варіативності фактичних даних.

## 3.3 Аналіз результатів прогнозування ринку ІТ-вакансій

На рис. 3.5 та 3.6 представлені історичні дані та результати прогнозування кількості ІТ-вакансій в Україні за період з липня 2023 року по січень 2024 року.

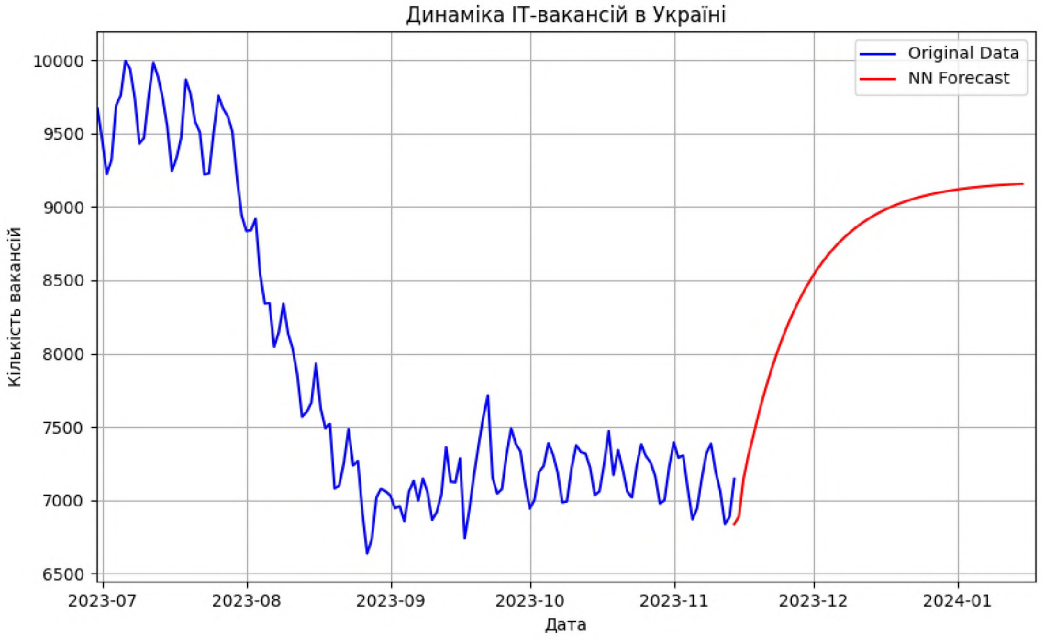



Рисунок 3.5 – Динаміка IT-вакансій в Україні: візуалізація історичних та прогнозних даних з липня по січень 2024 року (модель 1)

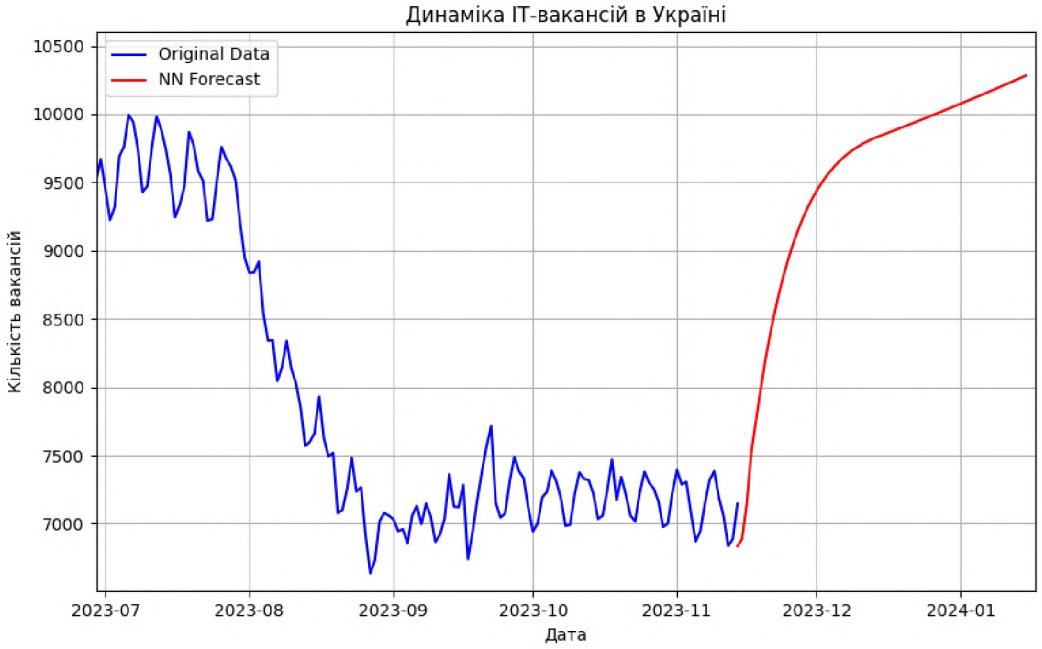



Рисунок 3.6 – Динаміка IT-вакансій в Україні: візуалізація історичних та прогнозних даних з липня по січень 2024 року (модель 2)

Історичні дані показують, що з вересня по листопад 2023 року кількість IT-вакансій в Україні знаходилась на мінімальному рівні близько 7000 за весь

період спостережень, який охоплює дана робота. Обидві розглянуті моделі прогнозують швидке зростання кількості IT-вакансій до кінця поточного 2023 року та на початку наступного 2024 року. Особливо швидке зростання прогнозується протягом найближчих двох тижнів (з середини листопада 2023 року до початку грудня 2023 року), далі ріст продовжиться, хоча й помітно уповільниться.

Модель 1 на найближчі 2 тижні показує зростання приблизно з 7000 до 8500 (рис. 3.5), тобто приблизно на 1500, що становить 21,4% від базового рівня. Значення 8500 приблизно відповідає кількості вакансій рік тому – на кінець 2022 року початок 2023 року, але залишиться майже у 2 рази меншим, ніж на кінець травня 2022 року. Загалом, з 15 листопада 2023 року по 13 січня 2024 року модель 1 прогнозує зростання кількості IT-вакансій в Україні приблизно до рівня 9200, або на 2200 одиниць, що становить 31,4% до базового (поточного) рівня.

Модель 2 також показує стрімке зростання кількості IT-вакансій у найближчі 2 тижні, приблизно з 7000 до 9500 (рис. 3.6), тобто на 2500, або на 35,7% від базового рівня. Значення 9500 приблизно відповідає кількості вакансій на кінець листопада 2022 року. З 15 листопада 2023 року по 13 січня 2024 року модель 2 прогнозує зростання кількості IT-вакансій в Україні приблизно до рівня 10300, або на 3300 одиниць, що становить 47,1% до поточного рівня.

Таким чином, обидві моделі показують, що наразі кількість IT-вакансій в Україні має історичний мінімум за весь період, що розглядається. Обидві моделі також прогнозують зростання кількості вакансій найближчим часом (приблизно, протягом двох місяців 15.11.2023 по 13.01.2024 від 30% (модель 1) до 50% (модель 2).

Відмінність між двома моделями у тому, що модель 1 прогнозує швидке уповільнення зростання, а модель 2 також прогнозує уповільнення зростання, але не таке швидке.

## 3.4 Порівняння з існуючими методами прогнозування

Вище були розглянуті прогнозі, отримані з використанням НМ. Також, для прогнозування часових рядів використовують економетричні методи, зокрема, модель ARIMA, яка часто використовується для аналізу і прогнозування часових рядів. Вважається, що модель ARIMA добре підходить для даних, яким властиві тренди або сезонні коливання, і може враховувати різні аспекти, такі як тренд, сезонність і шум у даних. Згідно цього підходу, спочатку дані перевіряються на наявність трендів, сезонності та інших характеристик, що впливають на вибір моделі та її параметрів. Це можна зробити шляхом візуального аналізу. Потім використовується модель ARIMA для аналізу часового ряду відшукання прогнозу на найближчий період. Для визначення параметрів моделі ARIMA використовують ADF-тест для перевірки стаціонарності даних та автокореляційну функцію для визначення параметрів авторегресії (AR) і ковзного середнього (MA).

Аналіз часових рядів з використанням моделі ARIMA:

1. Перевірка на стаціонарність – використання ADF-тесту (Augmented Dickey-Fuller test) для визначення, чи є даний часовий ряд стаціонарним. Якщо дані не є стаціонарними, застосовують диференціювання;

2. Визначення параметрів ARIMA – використання автокореляційної (ACF) та часткової автокореляційної (PACF) функцій для визначення параметрів AR (p) та MA (q);

3. Підбір моделі ARIMA – використання різних комбінацій параметрів (p, d, q) для моделі ARIMA та вибір моделі з найкращим показником AIC (Akaike Information Criterion);

4. Прогнозування – після підбору моделі можна використати її для прогнозування майбутніх значень.

Вказані кроки можна виконати за допомогою програмного забезпечення для аналізу даних, як R або Python з бібліотеками Pandas і statsmodels (лістингА.34).

На основі зібраних вихідних даних з використанням моделі ARIMA був отриманий альтернативний прогноз IT-вакансій (рис. 3.7 і 3.8).

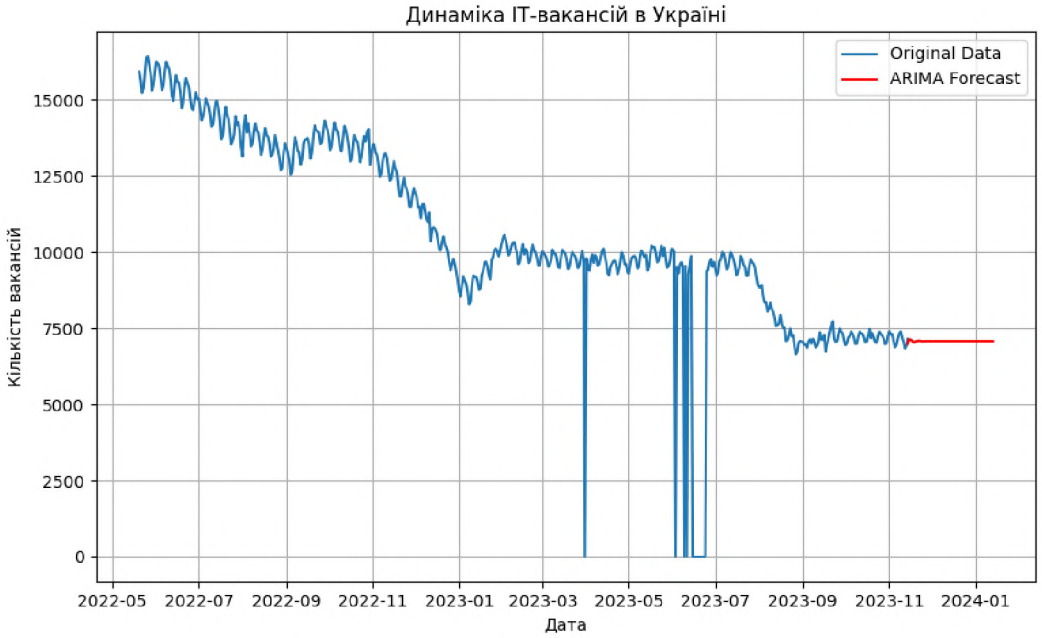



Рисунок 3.7 – Динаміка IT-вакансій в Україні з 20.05.2022 по 14.11.2023 та прогноз на період з 15.11.2023 по 13.01.2024 (модель ARIMA)

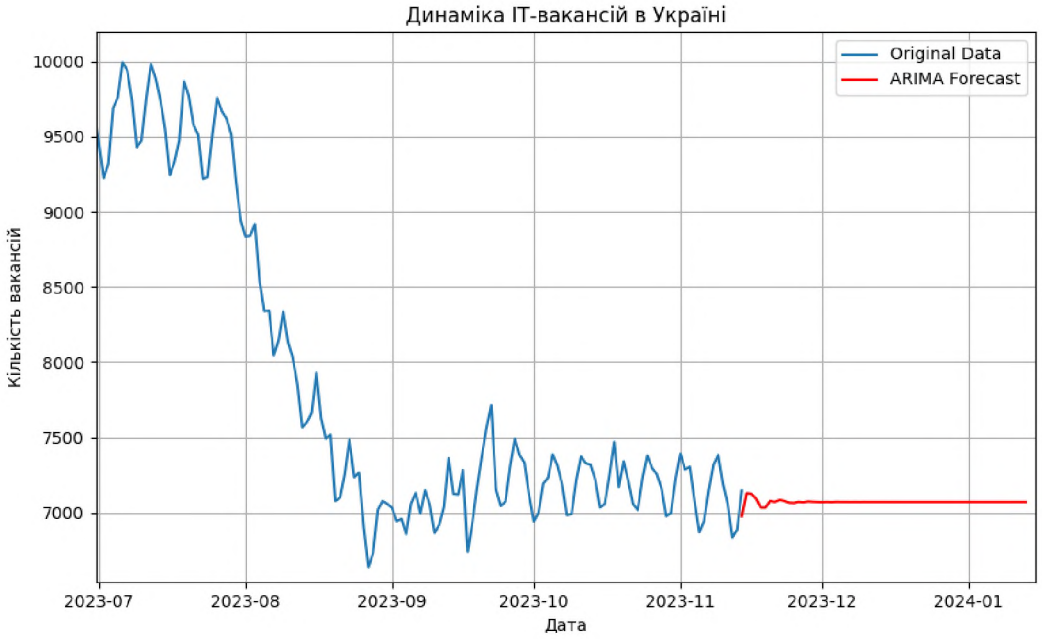



Рисунок 3.8 – Динаміка IT-вакансій в Україні: візуалізація історичних та прогнозних даних з липня по січень 2024 року (модель ARIMA)

Згідно отриманого результату, модель ARIMA прогнозує, що найближчим часом, а саме 15.11.2023 по 13.01.2024, кількість IT-вакансій в Україні не зміниться, і залишиться на мінімальному рівні, приблизно 7100 на місяць.

Цей результат відрізняється від прогнозу з використанням НМ, і свідчить, що у даному випадку модель ARIMA відображає лише бічний тренд найближчих попередніх періодів, починаючи приблизно з вересня 2023 року, і не враховує закономірності більш ранніх періодів. Таким чином, можна зробити висновок, що у даних умовах використання моделі ARIMA для прогнозування ринку IT-вакансій є недоцільним.

### 3.5 Висновки щодо ефективності моделі

Аналізуючи результати та приклади, наведені у роботі, можна сформулювати наступні висновки щодо моделей для прогнозування кількості IT-вакансій на основі НМ:

1. Використання НМ дозволяє ефективно обробляти великі обсяги даних та ідентифікувати складні залежності в даних. Це дозволяє забезпечити більш точне прогнозування, ніж традиційні статистичні методи.

2. Ефективність моделі значною мірою залежить від якості та обробки вхідних даних. У нашому випадку були використані чітко структуровані дані з часовими мітками і кількісними показниками, що є важливим для точного прогнозування.

3. НМ була налаштована з урахуванням особливостей даних про IT-вакансії, включаючи аналіз трендів за часом. Такий підхід підвищує ймовірність отримання достовірних прогнозів.

4. Використані метрики, такі як MSE, MAE, і $R^2$, дозволяють оцінити, наскільки добре модель виконує свою задачу. Чим нижчі значення MSE та

MAE і чим вище $R^2$, тим точнішим є прогноз. Виконана оцінка моделі показала, що модель має високу точність прогнозування.

4. Розроблена НМ демонструє високу здатність до прогнозування кількості ІТ-вакансій, що є важливим для розуміння ринкових тенденцій та потреб у кваліфікованих кадрах у ІТ-галузі. Проте, для досягнення максимальної точності та надійності прогнозів варто продовжувати дослідження та розвиток моделі.

5. Незважаючи на ефективність побудованої НМ, існує потенціал для її покращення, наприклад, через додаткове налаштування гіперпараметрів, збільшення розміру навчальної вибірки, використання більш складних архітектур НМ, використання складніших алгоритмів МН. Також ефективність може бути підвищена за рахунок збільшення обсягу та різноманітності навчальних даних.

В цілому, розроблена модель демонструє здатність адекватно прогнозувати кількість ІТ-вакансій, що є важливим для планування та аналізу ринку праці в ІТ-секторі. Однак, для досягнення вищої точності та надійності прогнозів потрібно провести додаткові дослідження та оптимізацію моделі.

## 3.6 Економічна оцінка проєкту

Економічна оцінка проєкту з розробки НМ для прогнозування кількості ІТ-вакансій враховує кілька ключових моментів:

1. Вартість розробки – включає витрати на команду розробників, які працюють над проєктом, включно з заробітними платами, витратами на навчання, конференції та додаткове обладнання або програмне забезпечення;

2. Витрати на обробку даних – включає витрати на збір та очищення даних, а також потенційні витрати на доступ до баз даних або API для збору даних;

3. Вартість обчислювальних ресурсів. НМ вимагають значних обчислювальних ресурсів для тренування та тестування моделі, що може включати вартість хмарних сервісів, серверів або спеціалізованого обладнання;

4. Економічна вигода. Оцінка того, як проєкт може вплинути на доходи компанії або зекономити витрати. Наприклад, точне прогнозування ринку IT-вакансій може допомогти компаніям ефективніше планувати набір персоналу, мінімізувати витрати на непотрібні рекрутингові заходи та оптимізувати розподіл ресурсів.

5. Обчислення ROI (Return on Investment). Для визначення економічної ефективності проєкту порівнюють загальні витрати проєкту з отриманими економічними вигодами (наприклад, зекономлені витрати, збільшення доходів). Це дасть змогу визначити, чи виправдовують потенційні вигоди вкладені кошти.

Для точної економічної оцінки потрібні конкретні дані про витрати та потенційні вигоди, залежно від умов та цілей конкретного проєкту. Оцінка вартості розробки проєкту з використання НМ для прогнозування кількості IT-вакансій залежить від ряду факторів.

1. Параметри команди розробників:

- розмір команди – витрати залежать від кількості людей у команді (дата-інженери, дата-аналітики, машинні інженери тощо);

- заробітна плата – середня заробітна плата у галузі залежить від регіону та кваліфікації працівників.

2. Тривалість розробки – залежно від складності проєкту та наявності вихідних даних, проєкт може тривати від кількох місяців до року та більше.

3. Витрати на обробку даних – витрати на збір, очищення та обробку даних.

4. Обчислювальні ресурси – використання хмарних сервісів або власних серверів для тренування моделей. Вартість хмарних обчислень залежить від обраного постачальника та обсягу використаних ресурсів.

5. Додаткові витрати – можуть включати витрати на ліцензійне програмне забезпечення, обладнання, навчання та розвиток персоналу, консалтинг тощо.

Для орієнтовної оцінки вартості розробки, припустимо, що проєкт розробляється середньою командою з 5 осіб (дата-інженер, дата-аналітик, два машинні інженери, керівник проєкту) протягом 6 місяців. Використаємо прикладні дані, щоб продемонструвати загальний хід розрахунків.

При заробітній платі $5000 на місяць на одного працівника, загальні витрати на заробітну плату становитимуть:

5осіб×$5000/місяць×6місяців=$150000.

Врахуємо витрати на обчислювальні ресурси (наприклад, $1000 на місяць), витрати на обробку даних та додаткові витрати ($10000). Тоді, загальна вартість проєкту складе приблизно:

$150000 + (6×$1000) + $10000 = $166000.

Ця оцінка є приблизною і може значно варіюватися в залежності від конкретних умов проєкту та регіону розробки.

Оцінка витрат на обробку даних для проєкту з використанням НМ для прогнозування ІТ-вакансій передбачає урахування витрат на виконання окремих видів робіт.

Збір даних можливий з відкритих та/або платних джерел. Якщо використовуються відкриті джерела даних, витрати можуть бути мінімальними або відсутніми. Проте, потрібно врахувати час, витрачений на пошук та оцінку якості цих даних. Якщо дані збираються з платних баз даних або через API, витрати залежатимуть від тарифів постачальників.

Очищення та попередня обробка даних передбачає видалення дублікатів, обробку відсутніх значень, нормалізацію даних тощо. Її вартість залежить від складності даних та обсягу роботи, який необхідно виконати. Витрати на цей етап можуть включати заробітну плату співробітників, які займаються обробкою даних, або вартість аутсорсингових послуг.

Якщо для проєкту потрібно 2 місяці роботи для збору та обробки даних, і на це залучені двоє аналітиків з середньою заробітною платою $4000 на місяць, то витрати на їх заробітну плату складуть:

2аналітики×$4000/місяць×2місяці=$16000.

Витрати на доступ до баз даних або API приблизно $2000, ще $1000 на програмне забезпечення та інструменти. Таким чином, загальні витрати на обробку даних можуть скласти:

$16000 + $2000 + $1000=$19000.

Реальні витрати можуть варіюватися в залежності від конкретних умов та потреб проєкту.

Витрати на інструменти та програмне забезпечення, а також на спеціалізоване програмне забезпечення для обробки даних, якщо таке використовується.

Оцінка витрат на обчислювальні ресурси для тренування НМ включає в себе вартість використання хмарних сервісів або утримання власних серверів. Розглянемо обидва варіанти.

Використання хмарних сервісів. Хмарні платформи, такі як AWS (Amazon Web Services), Google Cloud Platform або Microsoft Azure, пропонують різні тарифи для обчислювальних потужностей. Витрати залежать від типу використовуваних машин (наприклад, з GPU або без), тривалості їх використання та регіону. Для машину середньої потужності з GPU для тренування моделі протягом 2 місяців по 8 годин на день вартість становитиме від $1 до $3 за годину [67, 68, 69].

Приймемо вартість використання хмарних сервісів $2 за годину:

2місяці×30днів/місяць×8годин/день×$2/година=$960.

Утримання власних серверів. Включає вартість покупки серверів, їх обслуговування, електроенергії, охолодження та інших витрат, пов’язаних з утриманням інфраструктури. Вартість може значно варіюватися в залежності від конфігурації серверів та обсягу використання.

Припустимо, що вартість утримання серверів становить $500 на місяць (включає амортизацію обладнання, витрати на електроенергію та інші витрати):

2місяці×$500/місяць=$1000.

Отже, загальні витрати на обчислювальні ресурси можуть становити від $960 (при використанні хмарних сервісів) до $1000 (при утриманні власних серверів) за 2 місяці.

Оцінка економічної вигоди від проєкту з прогнозування ринку IT-вакансій може бути здійснена шляхом аналізу потенційного впливу на доходи компанії та можливостей для економії витрат. Розглянемо можливі аспекти цього впливу.

Оптимізація процесу набору персоналу. Точне прогнозування дозволяє компаніям планувати набір персоналу відповідно до потреб ринку, що може призвести до зменшення витрат на заходи з набору персоналу. Наприклад, якщо компанія зекономить $5000 на кожному заході набору завдяки ефективнішому підбору кандидатів, а в рік вона проводить 10 таких заходів, економія складе $50000.

Підвищення ефективності використання ресурсів, у т.ч.: зменшення витрат на зайві рекрутингові заходи, наприклад, на рекламу вакансій, що не є актуальними на ринку; економія часу HR-відділу та менеджерів, що займаються набором персоналу, і може бути спрямована на інші завдання.

Зменшення витрат на підготовку та адаптацію нових співробітників. Якщо прогнозування дозволяє залучати співробітників з потрібними навичками у відповідний час, це може зменшити потребу в додатковому навчанні та адаптації.

Потенційне збільшення доходів. Здатність швидко та ефективно реагувати на зміни на ринку може дозволити компанії запропонувати конкурентоспроможні послуги або продукти, що в свою чергу може підвищити доходи.

Довгостроковий вплив. Прогнозування ринкових тенденцій дозволяє компанії бути крок попереду конкурентів, що може мати позитивний вплив на її репутацію та стійкість на ринку.

Для точної оцінки економічної вигоди також необхідно провести детальний аналіз внутрішніх процесів компанії та ринкових умов. Однак, вже на основі вищенаведених припущень, можна зробити висновок, що точне прогнозування ринку IT-вакансій має потенціал принести значну економічну вигоду.

Розглянемо розрахунок економічної вигоди від використання НМ для прогнозування ринку IT-вакансій на прикладі умовної компанії.

Оптимізація процесу набору персоналу. Припустимо, що раніше компанія витрачала в середньому $3000 на рекрутинг одного співробітника, у тому числі на рекламу вакансій, інтерв'ю тощо. Завдяки точному прогнозуванню, компанія зменшує витрати на 30%, тобто на $900 за кожен процес набору. Якщо в рік компанія проводить 20 процесів набору, загальна економія складе:

$900 × 20 = $18000.

Ефективність використання ресурсів. Економія часу HR-відділу. Припустимо, це дозволяє зекономити 100 годин роботи HR-спеціалістів на рік. При середній ставці $30/год, економія складе:

100 год × $30/год = $3000.

Зменшення витрат на адаптацію персоналу. Більш точний підбір кандидатів зменшує час адаптації нових співробітників на 10 годин на одного співробітника. Якщо за рік компанія наймає 20 співробітників, то загальна економія складе:

10 год × 20 співробітників × $30/год = $6000.

Потенційне збільшення доходів важко точно оцінити. Припустимо, що завдяки більш ефективному плануванню та швидкій реакції на ринкові умови, дохід компанії зростає на 2%. Якщо щорічний дохід компанії становить $1000000, додатковий дохід складе: $1000000 × 2% = $20000.

Таким чином, загальна економічна вигода складе:

- економія на рекрутингу: $18000;
- економія часу HR-відділу: $3000;
- економія на адаптації персоналу: $6000;
- додатковий дохід: $20000.

Разом: $18000 + $3000 + $6000 + $20000 = $47000.

Попередні розрахунки витрат були такими:

- витрати на розробку: $166000;
- витрати на обчислювальні ресурси: $960;
- витрати на обробку даних: $19000;
- економія на наборі персоналу: $18000;
- економія часу HR-відділу: $3000;
- економія на адаптації персоналу: $6000;
- додатковий дохід: $20000.

Разом: $185960.

Розрахунок ROI виконується за формулою:

$$ROI = (ЗЕВ – ЗВП)/ЗВП \times 100, \quad (3.1)$$

де: ЗЕВ – загальна економічна вигода; ЗВП – загальні витрати на проєкт.

За формулою (3.1) знайдемо:

$$ROI = (47000 – 185960)/185960 * 100 = =-74,73.$$

У даному випадку, розрахунок показує негативний ROI, що може свідчити про те, що витрати на проєкт значно перевищують його економічну вигоду. Однак, важливі додаткові нематеріальні вигоди, такі як покращення репутації компанії або збільшення ефективності роботи, не були враховані в цих розрахунках. Економічна ефективність проєкту може бути підвищена за рахунок зменшення витрат на його реалізацію або збільшення економічної вигоди у довгостроковій перспективі.

## Висновки до розділу 3

Розроблена та навчена НМ для прогнозування ринку IT-вакансій. Навчання моделі базувалося на підготовлених та оброблених історичних даних, що забезпечило її високу здатність до прогнозування.

Для оцінки ефективності моделі на основі НМ використані метрики MSE, MAE та $R^2$, які продемонстрували задовільну точність прогнозів. Аналіз метрик показав, що модель здатна достовірно прогнозувати тренди на ринку IT-вакансій. Прогнози, генеровані моделлю, дозволяють зробити висновки про майбутні тенденції на ринку IT-вакансій. Модель здатна виявляти сезонні коливання та інші важливі особливості ринку. Порівняння з існуючими методами прогнозування показало, що НМ перевершує традиційні методи прогнозування за точністю та гнучкістю.

НМ продемонструвала високу ефективність у прогнозуванні ринку IT-вакансій. Модель може бути використана для підтримки стратегічного планування та рішень у сфері HR та найму персоналу.

Загальні витрати на проєкт включають витрати на розробку, обробку даних та обчислювальні ресурси. Економічна вигода від реалізації проєкту є значною за рахунок зменшення витрат на наймання персоналу, більш ефективне використання ресурсів та потенційне зростання доходів компанії. Незважаючи на позитивні результати оцінки економічної вигоди, проєкт має негативний ROI через високі початкові витрати, що вимагає додаткових заходів для підвищення економічної ефективності у довгостроковій перспективі.

# ВИСНОВКИ

Результати роботи дозволяють зробити такі висновки:

1. У роботі представлено основні тенденції та особливості ринку IT-вакансій в Україні, а також розглянуто різні підходи до їх прогнозування;

2. Розроблено та навчено НМ, яка демонструє високу ефективність у прогнозуванні ринку IT-вакансій; результати аналізу підтверджують здатність моделі точно прогнозувати ринкові тренди, включаючи сезонні зміни та інші важливі особливості; оцінка результатів прогнозування, зокрема за допомогою метрик точності, виявила високу надійність та практичну придатність моделі для ринкового аналізу;

3. Виконане порівняння розробленої моделі на основі НМ з традиційними методами прогнозування (модель ARIMA) підтвердило її переваги, особливо у здатності адаптуватися до складних залежностей у даних;

4. Здійснена економічна оцінка проєкту та його ROI: незважаючи на високі початкові витрати, проєкт демонструє значну економічну вигоду через оптимізацію процесів набору персоналу, більш ефективне використання ресурсів та потенційне збільшення доходів; негативний ROI вказує на необхідність оптимізації витрат або підвищення економічної вигоди для досягнення позитивного економічного ефекту в довгостроковій перспективі.

Елементи наукової новизни роботи полягають у наступному:

1. Використання НМ для прогнозування ринку IT-вакансій в Україні є прогресивним підходом, що відрізняється від традиційних методів аналізу, і дозволяє глибше зрозуміти динаміку та тенденції в галузі;

2. Використання НМ для цієї конкретної задачі є наступним кроком в аналітиці ринку праці українського IT-сектору;

3. Дослідження впливу різних функцій втрат та оптимізаційних алгоритмів на ефективність прогнозування є внеском у розуміння того, як ці технічні елементи впливають на якість прогнозу;

4. Через здатність НМ виявляти складні залежності в даних, дослідження надає нові відомості про фактори, що впливають на ринок IT-вакансій;

5. Дослідження включає аналіз ефективності НМ порівняно з іншими методами, що надає цінне бачення щодо їхньої практичної застосовності.

Практичне значення даної роботи виявляється в наступних аспектах:

1. Розроблена НМ для прогнозування ринку IT-вакансій може служити інструментом для HR-менеджерів та керівників IT-компаній, дозволяючи їм більш точно планувати набір та розвиток персоналу.

2. Прогнозування тенденцій на ринку IT-вакансій допомагає компаніям адаптувати свої рекрутингові стратегії, зосереджуючись на найбільш затребуваних спеціалізаціях та навичках, що знижує витрати та час на пошук кандидатів.

3. Дослідження виявило ключові фактори та тренди, що впливають на зміни у попиті та пропозиції праці.

4. Результати дослідження можуть використовуватися для підтримки стратегічних рішень на рівні компаній та державних організацій, спрямованих на розвиток IT-сектору та освітніх програм.

5. Робота може бути використана як основа для подальших досліджень у сфері прогнозування ринку праці, а також як навчальний матеріал для вивчення застосування НМ в економіці та управлінні.

У цілому, робота додає внесок у розвиток методів аналізу ринку праці, зокрема, застосування ШІ та МН для покращення прогнозування в сфері IT-вакансій.

Перспективами подальших досліджень є:

1. Включення більш широкого спектру даних, наприклад, інформації про вимоги до навичок, рівні заробітної плати, географічну розподіленість вакансій тощо, з метою покращення точності та деталізації прогнозів. Аналіз того, як новітні технології, такі як штучний інтелект, блокчейн та Інтернет речей, впливають на зміни у попиті на IT-спеціалістів.

2. Використання більш складних архітектур нейронних мереж, включаючи глибоке навчання та конволюційні НМ, для обробки більш складних даних та забезпечення вищої точності прогнозування.

3. Розширення дослідження шляхом порівняння ринків IT-вакансій в різних країнах для виявлення глобальних трендів та регіональних особливостей.

4. Створення моделей, що прогнозують майбутні потреби в освіті та навчанні IT-спеціалістів, адаптуючи освітні програми до майбутніх вимог ринку.