

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)
Факультет інтелектуальних інформаційних технологій та
автоматизації
(повне найменування інституту, назва факультету (відділення))
Кафедра комп'ютерних наук
(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Інформаційна технологія прогнозування індексу споживчих цін в
умовах нестабільної економіки»

Виконав: студент 2-го курсу, групи 2КН-23м
спеціальності 122 «Комп'ютерні науки»
(шифр і назва напрямку підготовки, спеціальності)

Кравчук А.І.
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН

Арсенюк І.Р.
(прізвище та ініціали)

«05» 12 2024 р.

Опонент: к.т.н., доцент каф. АІТ

Кабачій В.В.
(прізвище та ініціали)
«05» 12 2024 р.

Допущено до захисту

Завідувач кафедри КН

д.т.н., проф. Яровий А.А.

(прізвище та ініціали)

«06» 12 2024 р.

Вінниця ВНТУ - 2024 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти II-й (магістерський)
Галузь знань – 12 «Інформаційні технології»
Спеціальність – 122 «Комп'ютерні науки»
Освітньо-професійна програма – «Системи штучного інтелекту»

ЗАТВЕРДЖУЮ

**Завідувач кафедри КН
Д.т.н., проф. Яровий А.А.**

11.10 2024 року

ЗАВДАННЯ НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Кравчуку Андрію Івановичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Інформаційна технологія прогнозування індексу споживчих цін в умовах нестабільної економіки.

керівник роботи: к.т.н., доцент кафедри КН Арсенюк І.Р.

затверджені наказом вищого навчального закладу від "13" 09 2024 року № 310

2. Строк подання студентом роботи 11.11 2024 року

3. Вихідні дані до роботи: мови програмування JavaScript, Python; середовища розробки Visual Studio Code 2022; інструмент Navicat; файл API з даними індексу споживчих цін періоду який використовується для тестування.

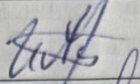
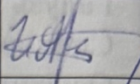
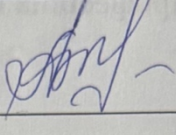
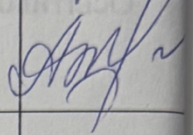
4. Зміст текстової частини:

Вступ, аналіз існуючих рішень поставленої задачі, проектування інформаційної технології прогнозування індексу споживчих цін в умовах нестабільної економіки, програмна реалізація індексу споживчих цін в умовах не стабільної економіки, тестування та аналіз розробленого програмного забезпечення, економічна частина інформаційної технології прогнозування індексу споживчих цін в умовах нестабільної економіки, висновки, список використаних джерел, додатки.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень)

Алгоритм програмного прогнозування індексу споживчих цін, алгоритм первинного аналізу даних, алгоритм прогнозування з Gradient Boosting, архітектура розробленого веб-застосунку, взаємодія користувача з MVC системою, опис таблиць та зв'язків бази даних системи алгоритм роботи модулю прогнозування, прогнозування модифікованого та класичного способів архітектури програмного забезпечення.

Консультанти розділів роботи

Розділ	Прізвище, ініціалита посада консультанта	Підпис, дата	
		завдання видав	виконав прийняв
1-4	Арсенюк О.О., к.т.н., доц. каф. КН		
5	Адлер Н.В., к. т. н., доцент. каф. ЕПВМ		

Дата видачі завдання 02.09 2024 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	При
1	Грунтовне ознайомлення з предметною галуззю	02.09.24-08.09.24	
2	Визначення структури магістерської дисертації; вивчення літератури, пошук додаткової літератури, патентний пошук	08.09.24 - 14.09.24	
3	Робота над першим розділом магістерської дисертації; проведення наукового дослідження	15.09.24-25.09.24	
4	Проведення наукового дослідження; робота над другим розділом магістерської дисертації; розроблення програмного забезпечення	26.09.24-02.10.24	
5	Проведення наукового дослідження; робота над статтею за результатами наукового дослідження	03.10.24-18.10.24	
6	Проведення наукового дослідження; робота над третім розділом магістерської дисертації	19.10.24-28.10.24	
7	Економічна частина	29.10.24-04.11.24	
8	Оформлення пояснювальної записки, графічного матеріалу, презентації	05.11.24-11.11.24	

Студент

Керівник роботи

(підпис)

(підпис)

Кравчук А.І.

Арсенюк І.Р.

АНОТАЦІЯ

УДК 004.8

Кравчук А.І. Інформаційна технологія прогнозування індексу споживчих цін в умовах нестабільної економіки. Магістерська кваліфікаційна робота зі спеціальності 122 – комп'ютерні науки, освітня програма – системи штучного інтелекту. Вінниця: ВНТУ, 2024. 118 с.

Українською мовою. Бібліогр.: 45 назв; рис.: 11; табл. 15.

У магістерській кваліфікаційній роботі розроблено інформаційну технологію прогнозування індексу споживчих цін в умовах нестабільної економіки. Проаналізовано інші рішення прогнозування, запропоновано новий формат програмного продукту для виконання задачі аналізу. Проаналізовано способи машинного навчання з метою виявлення найточнішого. Виконана детальна оцінка та інтерпретація результатів, отриманих в ході дослідження. Наведено архітектуру веб-додатку, схему роботи MVC системи, діаграма бази даних, діаграму роботи модулю прогнозування, графік порівняння точності модифікованого та класичного способів. Результатом роботи є програмне забезпечення, яке використовує модифікований спосіб прогнозування індексу споживчих цін, що дозволяє отримувати більш точні прогнози порівняно із статистичним методом прогнозування індексу споживчих цін.

Графічна частина складається з 8 плакатів.

У економічному розділі, оцінивши інформаційну технологію та ідею, визначено, що розробка відповідає при значенні інтегральної новизни 0,89 - рівню достатня новизна, термін окупності становить 1,12 р., що свідчить про комерційну привабливість науково-технічної розробки. Рівень комерційного потенціалу розробки становить 44,3 бали (високий), що свідчить про комерційну важливість проведення даних досліджень.

Ключові слова: Індекс споживчих цін, машинне навчання, веб- додаток.

ABSTRACT

Kravchuk A.I. Information Technology for Forecasting the Consumer Price Index in Conditions of an Unstable Economy. Master's Qualification Thesis in Specialty 122 – Computer Science, Educational Program – artificial intelligence systems. Vinnytsia: VNTU, 2024. 123 pages.

Language: Ukrainian. References: 42 sources; Figures: 11; Tables: 15.

The master's thesis develops information technology for forecasting the consumer price index under conditions of an unstable economy. The study analyzes commercial solutions for detecting HTTP requests with anomalous behavior and proposes a new software format for task analysis. Machine learning methods were evaluated to identify the most accurate approach. A detailed evaluation and interpretation of the research results are presented. The work includes the architecture of a web application, an MVC system workflow diagram, a database schema, a module algorithm for forecasting, a comparison chart of the accuracy of modified and classical methods, and a problem tree.

The result is software that uses a modified method for forecasting the consumer price index, enabling more accurate predictions compared to traditional statistical methods.

The graphical component consists of 8 posters.

In the economic section, an evaluation of the information technology and concept demonstrated that the development corresponds to a sufficient level of novelty, with an integral novelty value of 0.89. The payback period is 1.12 years, indicating the commercial attractiveness of the scientific and technical development. The commercial potential of the development is rated at 44.3 points (high), underscoring the commercial importance of conducting this research.

Keywords: Consumer Price Index, Machine Learning, Web Application.

ЗМІСТ

ВСТУП	4
1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ПОСТАВЛЕНОЇ ЗАДАЧІ	
1.1 Аналіз предметної області прогнозування індексу споживчих цін	11
1.2 Огляд та аналіз існуючих рішень.....	10
1.3 Постановка задачі дослідження	18
1.4 Висновок до розділу 1	18
2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПРОГНОЗУВАННЯ ІНДЕКСУ СПОЖИВЧИХ ЦІН В УМОВАХ НЕСТАБІЛЬНОЇ ЕКОНОМІКИ	
2.1 Аналіз збору даних для прогнозування індексу споживчих цін	25
2.2 Аналіз математичних моделей прогнозування часових рядів у прогнозуванні індексу споживчих цін	28
2.3 Обґрунтування вибору техніки машинного навчання Gradient Boosting у прогнозуванні індексу споживчих цін.....	26
2.4 Розробка алгоритму роботи програмного забезпечення індексу споживчих цін.....	27
2.5 Висновки до розділу 2.....	36
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНДЕКСУ СПОЖИВЧИХ ЦІН В УМОВАХ НЕ СТАБІЛЬНОЇ ЕКОНОМІКИ	
3.1. Вибір мови програмування для веб-застосунку.....	38
3.2 Програмна реалізація індексу споживчих цін в умовах не стабільної економіки.....	49
3.3 Розробка інтерфейсу користувача	61
3.4 Розробка алгоритм роботи модуля прогнозування індексу споживчих цін в умовах не стабільної економіки.	66
3.5 Висновки до розділу 3	69
4 ТЕСТУВАННЯ ТА АНАЛІЗ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	
4.1 Тестування результатів роботи програми індексу споживчих цін в умовах не стабільної економіки.....	67
4.2 Аналіз розробленого програмного забезпечення індексу споживчих цін в умовах не стабільної економіки.....	74
4.3 Висновки до розділу 4.....	75

5 ЕКОНОМІЧНА ЧАСТИНА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПРОГНОЗУВАННЯ ІНДЕКСУ СПОЖИВЧИХ ЦІН В УМОВАХ НЕСТАБІЛЬНОЇ ЕКОНОМІКИ

5.1 Оцінювання комерційного потенціалу розробки.....	77
5.2 Прогнозування витрат на виконання науково-дослідної роботи.....	78
5.3 Розрахунок економічної ефективності науково-технічної розробки .	83
5.4 Розрахункова ефективність вкладених інвестицій та період їх окупності.....	88
5.5 Висновок економічної частини науково-технічної розробки.....	87
ВИСНОВКИ.....	88
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	95
Додаток А (обов'язковий) Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень.....	97
Додаток Б (обов'язковий) Лістинг програми	98
Додаток В (обов'язковий) Ілюстративна частина.....	111
Додаток Г (довідниковий) Інструкція користувача.....	123

ВСТУП

Актуальність. У ситуації нестабільної та динамічної економіки, яка склалася в Україні через військовий конфлікт, точне програмне прогнозування індексу споживчих цін ІСЦ стає надзвичайно високим. З огляду на значний економічний спад, який призвів до зниження ВВП на 30,4% у 2024 році, а також на серйозні зміни в експорті, включаючи обмеження доступу до ключових портів, існує критична потреба в реалістичних та адаптивних економічних прогнозах для ефективного реагування на економіку виклики та планування [2,3]. Точні прогнози ІСЦ допомагають уряду та бізнесу приймати обґрунтовані рішення, коригувати стратегії управління та формувати політику, що зменшує негативний вплив економічних потрясінь і сприяє.

Економічна невизначеність може спричинити ланцюгові реакції в різних секторах економіки. Так, підвищена невизначеність може змусити компанію скорочувати інвестиції та згортати свою діяльність, а домогосподарства — зменшити витрати і збільшити заощадження через м.

У світовій практиці програмного прогнозування ІСЦ широко застосовуються статистичні методи, такі як ARIMA, SARIMA, VAR тощо. Однак ці методи мають обмеження при застосуванні складних або нелінійних залежностей між даними для аналізу та прогнозування. Національний банк України (НБУ) застосовує комплексний підхід, який включає аналіз різних макроекономічних показників, а також стикається з подібними труднощами

З урахуванням вищезазначених потреб розробка модифікованого програмного методу прогнозування ІСЦ, що за рахунок переваг статистичних інструментів і методів машинного навчання, є актуальним завданням.

Методом цієї дисертації є розробка програмного забезпечення для підвищення точності прогнозування індексу споживчих цін за допомогою модифікованого методу прогнозування, що дозволяє врахувати вплив факторів нестабільності в економіці. Розробка орієнтована на вдосконалення програмного прогнозування ІСЦ, що є критичним для ухвалення ефективних рішень в умовах постійних змін.

Згідно з дослідженнями Федерального резерву США, невизначеність індексу споживчих цін має суттєвий вплив на економіку. Очікування щодо інфляції можуть вплинути на рішення споживачів та інвесторів, що у своєму випадку може змінити загальний економічний розвиток та стабільність. В сучасних умовах динамічної економіки, для індексу споживчих цін необхідно розробити методи програмного аналізу та прогнозування, які поєднують статистичні моделі з алгоритмами машинного навчання для підвищення точності прогнозів. Тому створення модифікованого методу прогнозування ІСЦ та програмного забезпечення для його реалізації є корисним [2,4].

Зв'язок роботи з науковими програмами, планами, темами. Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Розробка прикладних інтелектуальних інформаційних технологій та систем» та плану наукової та навчально-методичної роботи кафедри.

Мета та завдання дослідження. Підвищення точності прогнозування індексу споживчих цін в умовах нестабільності економіки.

Об'єкт дослідження - процес прогнозування індексу споживчих цін.

Предмет дослідження - програмне забезпечення та інформаційна технологія для прогнозування індексу споживчих цін.

Методи дослідження. В роботі використовуються методи збору даних, методи підготовки даних до обробки, методи модифікації, методи прогнозування, методи тестування.

Наукова новизна одержаних результатів полягає в наступному.

Удосконалено програмний спосіб прогнозування індексу споживчих цін, який використовує моделі Arimax та Sarimax у поєднанні з алгоритмом машинного навчання Gradient Boosting для збільшення точності прогнозування.

Практичне значення

1. Розроблено вдосконалений спосіб індексу споживчих цін, який використовує моделі Arimax та Sarimax у поєднанні з алгоритмом машинного навчання Gradient Boosting для збільшення точності прогнозування;

2. Розроблено вдосконалений алгоритм оцінки індексу споживчих цін;

3. Розроблено програмний засіб оцінки індексу споживчих цін.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням математичних методів під час доведення наукових положень, строгим виведенням аналітичних співвідношень, експериментальними дослідженнями тестування програмної реалізації інформаційної технології прогнозування індексу споживчих цін в умовах нестабільної економіки.

Особистий внесок магістранта. Усі результати, наведені у магістерській кваліфікаційній роботі, отримані самостійно [1].

Апробація результатів роботи. Результати роботи були апробовані на конференції «Молодь в науці: дослідження, проблеми, перспективи (МН2024)», Вінниця, 15 листопада 2024 року - 12 травня 2024 року [1].

Публікації. За результатами дослідження опубліковано тези доповіді на науково-технічній конференції [1.].

1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ПОСТАВЛЕНОЇ ЗАДАЧІ

1.1 Аналіз предметної області прогнозування індексу споживчих цін

Індекс споживчих цін є одним з макроекономічних показників, який відображає загальні тенденції у ціновій динаміці, який піддається впливу різноманітних факторів, таких як експорт, динаміка курсу валют, стан торговельного балансу, управлінські рішення державних структур тощо.

Серед важливих екзогенних метрик, що впливають на ІСЦ, можна виділити:

1. Середній рівень заробітної плати. Цей показник безпосередньо впливає на споживчий попит. Вища заробітна плата може підвищувати попит на товари та послуги, сприяючи зростанню цін на товари і відповідно сприяючи зростанню ІСЦ.
2. Рівень безробіття та зайнятості. Відображення загального стану економіки, ці показники мають інверсну кореляцію з ІСЦ. Зниження рівня безробіття зазвичай супроводжується підвищенням ІСЦ.
3. Грошова маса та кредитні умови. величина грошової маси та умови кредитування мають вплив на ІСЦ через монетарні канали, зокрема через зміну вартості грошей та доступності кредитних ресурсів.
4. Ціни на сировину та енергоресурси. Ці показники безпосередньо впливають на вартість виробництва товарів та послуг, що надалі впливає на ІСЦ.
5. Валютний курс. Особливо важливий для країн, що активно здійснюють імпорт та експорт. Зміна валютного курсу може призвести до імпортованої інфляції, особливо у випадках валютної волатильності.

Індекс споживчих цін (ІСЦ) в Україні піддається значній сезонності. Це явище можна спостерігати, наприклад, на прикладі так званих "борщових наборів" – набору продуктів, необхідних для приготування традиційного українського борщу. Взимку ціни на овочі, що входять до борщового набору, як правило, зростають через зниження їх доступності та підвищення витрат на їх зберігання та транспортування. Навпаки, влітку, коли овочі доступні в достатку і їх вирощування стає економічно вигіднішим, ціни на них знижуються.

Така сезонна динаміка впливає на ІСЦ, оскільки борщовий набір становить значну частину споживчої корзини більшості українців. Врахування цих сезонних коливань є важливим для розуміння загальних тенденцій у динаміці споживчих цін. Сезонність також має бути врахована при прогнозуванні ІСЦ, адже вона впливає на загальну картину інфляції в країні. Відтак, ефективне управління та прогнозування економічної динаміки в Україні вимагає глибокого аналізу та врахування сезонних чинників, які впливають на ціни на основні споживчі товари та послуги. Враховуючи ці обставини, сучасні програмні методи прогнозування індексу споживчих цін повинні включати можливість відстеження сезонних коливань.

Національний банк України (НБУ) використовує комплексний підхід для прогнозування ІСЦ, який включає аналіз різних макроекономічних показників та використання статистичних даних. Прогнози НБУ базуються на даних, опублікованих Державною службою статистики України, а також на власних аналітичних оцінках.

Для програмного прогнозування ІСЦ НБУ використовує такі моделі, як ARIMA, SARIMA, а також більш складні моделі, такі як VAR чи моделі з багатьма змінними, що дозволяють враховувати широкий спектр економічних індикаторів. Використання цих моделей дозволяє НБУ аналізувати макроекономічні тенденції та робити прогнози, базуючись на великому обсязі даних та різноманітних економічних показниках [5].

Для прогнозування ІСЦ НБУ аналізує різні фактори, що впливають на рівень цін, такі як подорожчання продуктів харчування, зміни на валютному ринку, інфляційні очікування, а також зовнішні та внутрішні економічні умови. Так, в січні 2023 року спостерігалось зниження споживчої інфляції до 26,0% у річному вимірі, що було нижчим за прогноз НБУ, при цьому основними факторами, що вплинули на інфляцію, були: збільшення пропозиції на ринку продуктів харчування, стабільність на валютному ринку, поліпшення інфляційних очікувань [6].

Таким чином, прогнозування ІСЦ НБУ є багатоаспектним процесом, що враховує як поточні економічні показники, так і різні фактори, які можуть вплинути на рівень цін в майбутньому.

Міжнародний досвід також довгий час демонстрував використання різних статистичних методів у програмному прогнозуванні ІСЦ. Центральні банки та економічні установи використовують методи ARIMA, SARIMA та інші, такі як VAR, FAVAR та Bayesian VARs, для програмного прогнозування індексів споживчих цін [7,8].

Спроможність програмно прогнозувати ІСЦ з високою точністю є ключовим аспектом для ефективної монетарної політики та економічного планування.

В свою чергу, методи машинного навчання пропонують рішення, здатні адаптуватися до змін та виявляти нелінійні взаємозв'язки. Станом на зараз з'являється позитивна тенденція в світовій практиці щодо інтеграції таких методів для прогнозування ІСЦ. Так в Євросоні використовуються машинні технології навчання, такі як Quantile Regression Forest, для прогнозування ІСЦ [9]. Цей підхід дозволяє аналізувати вклад різних змінних у прогнозування ІСЦ та ідентифікувати потенційні нелінійні зв'язки між змінними та ІСЦ. Використання таких методів також супроводжується певними викликами, зокрема проблемами перенавчання [10].

Методи машинного навчання пропонують значні переваги такі як висока адаптивність до нових даних, здатність виявляти складні та неочевидні для

статистичних методів зв'язки у великих наборах даних, а також вирішення нелінійних задач. Статистичні методи також мають свої переваги, включаючи надійність у прогнозуванні при відомих зв'язках між даними та простоту інтерпретації результатів. Оптимальним рішенням може бути об'єднання цих двох підходів, використовуючи статистичні методи для структурованого аналізу та машинне навчання для виявлення нових взаємозв'язків та шаблонів у даних.

1.2 Огляд та аналіз існуючих рішень

На сьогоднішній день, доступних програмних сервісів та платформ для прогнозування (ІСЦ) досить обмежено, що може пояснюватися тим, що більшість інструментів розробляються для внутрішнього використання організаціями, державними установами або фінансовими компаніями. Розробка подібних платформ вимагає значних інвестицій та специфічних знань, що обмежує їх поширення у відкритому доступі.

Однак, серед відкритих платформ та програмних рішень, які доступні для прогнозування ІСЦ, можна виділити кілька ключових.

Stata є потужним інструментом для аналізу даних та прогнозування (ІСЦ) завдяки своїм сильним можливостям у сфері економетрії та статистики [11]. Основні переваги та недоліки Stata для прогнозування ІСЦ можна резюмувати наступним чином.

Аналіз часових рядів: Stata підтримує такі моделі, як ARIMA та SARIMA, що дозволяють аналізувати тренди та сезонні коливання в часових рядах. Ці інструменти дають змогу визначити закономірності в історичних даних ІСЦ для точнішого прогнозування майбутніх змін.

Економетричний аналіз: Програма дає можливість здійснювати детальний аналіз впливу різних економічних змінних на ІСЦ, включаючи регресії, коефіцієнти еластичності та тестування гіпотез. Це допомагає визначити фактори, які найбільше впливають на зміну ІСЦ.

Інтеграція даних: Можливість імпортувати дані з різних джерел дозволяє збирати та аналізувати інформацію з національних і міжнародних баз даних. Легкість обробки великих обсягів даних є важливим аспектом для прогнозування макроекономічних показників.

Візуалізація даних: Stata пропонує зручні інструменти для графічного відображення даних, що допомагає наочно представити результати аналізу і виявити ключові тренди в зміні ІСЦ.

Переваги Stata для прогнозування ІСЦ. Надійність і точність: Програма забезпечує високий рівень точності в статистичних розрахунках. Широкий функціонал: Великий набір інструментів для аналізу дозволяє виконувати детальні економетричні дослідження. Користувацька підтримка: Stata має широку спільноту користувачів і безліч навчальних ресурсів, що полегшує освоєння програми. Недоліки Stata для прогнозування ІСЦ.

Висока вартість: Stata є комерційним продуктом, що може бути значною перешкодою для індивідуальних користувачів або невеликих організацій. Обмеження у машинному навчанні: Програма менш підходить для завдань, які вимагають застосування складних методів машинного навчання, порівняно з іншими сучасними платформами. Складність інтерфейсу: Для нових користувачів інтерфейс Stata може здатися менш інтуїтивним, що ускладнює її освоєння.

Stata є відмінним вибором для традиційного економетричного аналізу та прогнозування ІСЦ, особливо у випадках, коли потрібна точність і надійність розрахунків. Проте, для більш просунутих завдань, таких як обробка великих даних або використання методів машинного навчання, можливо, варто розглянути інші програмні рішення.

EViews є спеціалізованим інструментом для економетричного аналізу та роботи з часовими рядами, що робить його особливо корисним для прогнозування індексу споживчих цін (ІСЦ) [12]. Завдяки потужним методам аналізу, EViews дозволяє проводити точні дослідження на основі

даних про ІСЦ та інші економічні показники. Основні функції EViews для прогнозування ІСЦ.

Аналіз часових рядів: Пакет підтримує широкий спектр моделей, таких як ARIMA, SARIMA, VAR (векторна авторегресія) і SVAR (структурна векторна авторегресія). Ці моделі допомагають аналізувати тренди та коливання ІСЦ, враховуючи різні економічні фактори.

Економетричне моделювання: EViews дає змогу моделювати як екзогенні, так і ендогенні змінні, що дозволяє більш точно прогнозувати ІСЦ у різних економічних умовах. Вбудовані інструменти для тестування стабільності моделей та оцінки точності прогнозів.

Інтерфейс та візуалізація: EViews має зручний графічний інтерфейс, який спрощує роботу з моделями і даними. Широкі можливості візуалізації допомагають краще зрозуміти отримані результати та представити прогнозовані значення. Переваги EViews для прогнозування ІСЦ.

Висока точність прогнозів: Моделі EViews відомі своєю точністю, оскільки вони враховують різноманітні економічні чинники та історичні дані про ІСЦ.

Гнучкість: Користувачі мають можливість вибору різних методів і моделей для адаптації до специфіки досліджень.

Інтуїтивний інтерфейс: Зручний для користувачів інтерфейс спрощує роботу з великими обсягами даних і складними моделями. Недоліки EViews для прогнозування ІСЦ.

Висока вартість: Як і більшість комерційних програм, EViews є дорогим, що може стати бар'єром для невеликих організацій або індивідуальних користувачів.

Складність для новачків: Хоча інтерфейс інтуїтивний, для ефективного використання програмного забезпечення потрібне глибоке розуміння економетрії та статистичних методів.

бмеження в складних аналізах : Деякі складніші методики можуть вимагати додаткового програмування або інтеграції з іншими інструментами для досягнення необхідної гнучкості.

EViews є чудовим інструментом для прогнозування ІСЦ завдяки своїй потужній підтримці економетричних моделей і аналізу часових рядів. Це ідеальне рішення для тих, хто шукає баланс між точністю прогнозів і легкістю використання. Проте, через високу вартість та вимоги до рівня підготовки користувача, EViews може не підійти для всіх.

SAS (Statistical Analysis System) є потужним програмним забезпеченням для аналізу даних, яке знаходить широке застосування в економіці, фінансах та багатьох інших сферах [13].

Його можливості для прогнозування індексу споживчих цін (ІСЦ) підкріплені розвиненими статистичними методами та аналітичними інструментами. Основні функції SAS для прогнозування ІСЦ.

Розширений аналіз часових рядів: SAS пропонує потужні інструменти для роботи з часовими рядами, включаючи моделі ARIMA та SARIMA, які є критично важливими для виявлення трендів та сезонних коливань ІСЦ.

Мультиваріативний аналіз: Цей інструмент дозволяє вивчати взаємозв'язки між кількома економічними змінними, що допомагає краще зрозуміти динаміку ІСЦ та фактори, що його впливають.

Панельний аналіз даних: SAS підтримує аналіз панельних даних, що важливо для дослідження ІСЦ в різних регіонах або демографічних групах, забезпечуючи більш детальну та специфічну інформацію.

Інтеграція з базами даних та Big Data: SAS здатен працювати з великими обсягами даних, інтегруючись з різними джерелами інформації, що робить його ідеальним для аналізу складних наборів даних.

Візуалізація та звітність: Програмне забезпечення має потужні інструменти для візуалізації даних, що дозволяє ефективно представляти результати аналізу ІСЦ і спрощує їх інтерпретацію. Переваги SAS для прогнозування ІСЦ.

Гнучкість у виборі методів: SAS надає широкий спектр аналітичних методів, які можна адаптувати під конкретні потреби користувачів.

Висока точність моделювання: Завдяки використанню передових статистичних та аналітичних методів, SAS забезпечує високий рівень точності у прогнозах ІСЦ.

Потужні інструменти аналізу даних: SAS пропонує складні алгоритми для ретельного аналізу та обробки даних, що важливо для точності прогнозування.

Ефективна візуалізація: Графічне представлення даних допомагає користувачам зрозуміти тенденції та взаємозв'язки, що підвищує якість аналізу. Недоліки SAS для прогнозування ІС.

Висока вартість ліцензії: SAS є дорогим програмним забезпеченням, що може стати серйозним бар'єром для багатьох організацій, особливо невеликих.

Складність для новачків: Для ефективного використання SAS необхідні глибокі знання в статистиці та аналітиці, що може ускладнити його освоєння.

Вимоги до ресурсів: SAS може вимагати значних обчислювальних ресурсів для обробки великих наборів даних, що може бути обмеженням для деяких користувачів.

SAS є потужним інструментом для аналізу та прогнозування ІСЦ, надаючи різноманітні методи і функції для детального аналізу. Однак його ефективність у великій мірі залежить від рівня підготовки користувачів та доступності ресурсів, що може бути важливим фактором при виборі цього програмного забезпечення для конкретних задач.

SPSS (Statistical Package for the Social Sciences) є одним з найпопулярніших статистичних пакетів, що використовується у багатьох областях, включаючи соціальні науки, маркетинг, охорону здоров'я та освіту [14]. Його можливості для прогнозування індексу споживчих цін

(ІСЦ) є важливими для економічного аналізу та прийняття рішень. Основні функції SPSS для прогнозування ІСЦ.

Аналіз часових рядів: SPSS надає інструменти для аналізу часових рядів, включаючи моделі ARIMA, що є критично важливими для прогнозування ІСЦ та виявлення сезонних трендів.

Мультиваріативний аналіз: Це дозволяє проводити комплексний аналіз взаємодії між різними економічними змінними, що може допомогти у вивченні факторів, які впливають на ІСЦ.

Економетричне моделювання: SPSS підтримує різні економетричні моделі для вивчення і прогнозування економічних явищ, включаючи ІСЦ.

Інтуїтивний інтерфейс: SPSS відомий своїм зручним і зрозумілим інтерфейсом, що робить його доступним навіть для користувачів без глибоких знань у статистиці. Переваги SPSS для прогнозування ІСЦ.

Легкість використання: Завдяки простому інтерфейсу, SPSS є доступним для широкого кола користувачів, включаючи тих, хто не є спеціалістами в галузі статистики.

Адаптивність аналітичних методів: SPSS пропонує різноманітні статистичні тести та методи аналізу, що дозволяє користувачам адаптувати свої дослідження під конкретні потреби.

Візуалізація даних: Пакет містить потужні інструменти для візуалізації результатів, що полегшує інтерпретацію даних та прогнозів ІСЦ. Недоліки SPSS для прогнозування ІСЦ.

Вартість: SPSS є платним продуктом, що може обмежити його доступність для деяких користувачів або організацій, особливо для невеликих бізнесів.

Обмежена підтримка складних моделей: Деякі складні статистичні методи та моделі можуть бути недоступні або вимагати додаткових модулів, що може обмежити гнучкість аналізу.

Вимоги до обробки даних: Обробка великих наборів даних може бути обмеженою або вимагати значних обчислювальних ресурсів, що може ускладнити роботу з великими масивами даних.

SPSS є потужним інструментом для статистичного аналізу та прогнозування ІСЦ, особливо корисним для користувачів, які цінують простоту використання та широкий спектр аналітичних методів. Однак його використання може бути обмеженим через вартість та обмеження в обробці великих обсягів даних. У зв'язку з цим, організаціям і користувачам варто ретельно зважувати свої потреби і ресурси перед вибором цього програмного забезпечення для аналізу даних.

Bloomberg Terminal — це високопрофесійне програмне забезпечення, яке широко використовується в фінансовому секторі, зокрема інвестиційними банками, хедж-фондами та іншими фінансовими установами [15]. У контексті прогнозування індексу споживчих цін (ІСЦ) Bloomberg Terminal пропонує ряд унікальних можливостей і функцій, що роблять його важливим інструментом для професіоналів. Основні функції Bloomberg Terminal для прогнозування ІСЦ.

Доступ до великих обсягів даних: Bloomberg Terminal забезпечує доступ до величезної бази даних фінансових, економічних та інших пов'язаних даних, критично важливих для аналізу та прогнозування ІСЦ.

Аналітичні інструменти: Пакет включає розширені аналітичні інструменти для проведення глибокого аналізу даних, які можуть охоплювати історичні дані ІСЦ, макроекономічні індикатори, екзогенні змінні тощо.

Реальний час та історичні дані: Bloomberg Terminal надає інформацію в режимі реального часу та доступ до історичних даних, що дозволяє аналізувати тренди ІСЦ. Переваги Bloomberg Terminal для прогнозування ІСЦ.

Широкий обсяг даних: Неперевершений доступ до широкого спектру фінансових та економічних даних, що сприяє точності прогнозів.

Глибина аналітики: Висока деталізація та глибина аналітичних засобів дозволяють проводити комплексний аналіз впливу різних факторів на ІСЦ.

Надійність та актуальність: Дані, які надходять у Bloomberg Terminal, є дуже надійними та актуальними, що є ключовим для прогнозування ІСЦ.

Недоліки Bloomberg Terminal для прогнозування ІСЦ:

Висока вартість: Bloomberg Terminal є одним із найбільш дорогих рішень на ринку, що обмежує його доступність для деяких користувачів та організацій.

Складність використання: Для ефективного використання Bloomberg Terminal потрібні специфічні знання та навички, що може бути складним для нових користувачів.

Залежність від платформи: Використання Bloomberg Terminal вимагає доступу до спеціального обладнання або програмного забезпечення Bloomberg.

Програмні системи, такі як Stata, EViews, SAS, SPSS та Bloomberg Terminal, мають певні переваги та обмеження:

Stata: Сильні можливості для аналізу часових рядів та економетричного аналізу, але висока вартість та обмеження в машинному навчанні.

EViews: Спеціалізується на економетричному аналізі та використанні часових рядів, проте має високу вартість і складність для новачків.

SAS: Пропонує гнучкість у виборі методів і високу точність моделювання, але вимагає значних інвестицій у ліцензію та потужних обчислювальних ресурсів.

SPSS: Відомий своєю легкістю використання та гнучкістю аналітичних методів, але має обмеження в підтримці складних моделей.

Bloomberg Terminal: Надає детальні та точні прогнози з великою базою даних, але має високу вартість і складність використання.

Ці програмні системи, хоча й ефективні у традиційному статистичному аналізі та прогнозуванні, часто обмежені в інтеграції сучасних методів машинного навчання та автоматизації процесів. Це створює вагому потребу

в розробці нового програмного забезпечення, яке б інтегрувало статистичні методи з методами машинного навчання, що дозволило б користувачам отримувати більш точні та адаптивні прогнози.

1.3 Постановка задачі дослідження

Необхідна розробка спеціального вебзастосунку, що дозволяє поєднати традиційні та сучасні підходи для підвищення точності прогнозування ІСЦ. Підчас розробки потрібно, врахування особливостей даних ІСЦ, використання сучасних статистичних і машинних методів, оптимізація параметрів моделей, покращити аналіз та прогнозування макроекономічних показників, сприяючи економічній стабільності.

Основними джерелами історичних даних, які використовуються в традиційних методах ІСЦ є економічні показники України є НБУ, Світовий банк, Trading Economics. Доступ до даних можливий через API, для отримання структурованих даних. Ці дані будуть використані в нашому модифікованому підході до програмного прогнозування ІСЦ, який поєднує статистичні методи та алгоритми машинного навчання. Основні етапи: збір і попередня обробка даних, аналіз стаціонарності, побудова оптимізованих за Акаїке моделей ARIMAX/SARIMAX, перевірка мультиколінеарності, тестування моделей, уточнення прогнозу за допомогою Gradient Boosting.

1.4 Висновок до розділу 1

Індекс споживчих цін (ІСЦ) є важливим економічним показником, який відображає загальні тенденції цінової динаміки в країні. Він залежить від багатьох факторів, таких як доходи населення, рівень безробіття, грошово-кредитна політика, валютні курси тощо. Для України характерна сезонність ІСЦ, що значною мірою пов'язана із коливаннями цін на продукти харчування.

Національний банк України та міжнародні організації традиційно використовують статистичні методи для прогнозування ІСЦ. Проте методи машинного навчання демонструють певні переваги завдяки своїй здатності адаптуватися до нових даних і моделювати нелінійні залежності. Оптимальним підходом може бути комбінація статистичних і машинних методів, що дозволить підвищити точність прогнозів ІСЦ.

Серед існуючих статистичних пакетів, таких як Stata, EViews, SAS, SPSS та Bloomberg Terminal, є потужні інструменти для аналізу часових рядів, побудови економетричних моделей та проведення прогнозування. Їх переваги включають високу точність обчислень, широкий вибір аналітичних методів і зручні засоби візуалізації. Проте ці рішення часто є досить дорогими, складними у використанні та обмеженими в застосуванні сучасних методів машинного навчання:

Stata пропонує сильні можливості для аналізу часових рядів та економетричного аналізу, але має високу вартість і обмежені можливості у машинному навчанні.

EViews спеціалізується на економетричному аналізі та використанні часових рядів, однак його висока вартість і складність можуть ускладнити використання для новачків.

SAS забезпечує гнучкість у виборі методів і високу точність моделювання, проте потребує значних інвестицій у ліцензію та потужних обчислювальних ресурсів.

SPSS відомий своєю простотою у використанні та різноманіттю аналітичних методів, але має обмеження щодо підтримки складних моделей.

Bloomberg Terminal пропонує детальні та точні прогнози з величезною базою даних, проте його висока вартість і складність використання можуть бути перешкодою для багатьох користувачів.

Зроблено постановку задачі.

Необхідна розробка спеціального вебзастосунку, що дозволяє поєднати традиційні та сучасні підходи для підвищення точності прогнозування ІСЦ. Підчас розробки потрібно, врахування особливостей даних ІСЦ, використання сучасних статистичних і машинних методів, оптимізація параметрів моделей, покращити аналіз та прогнозування макроекономічних показників, сприяючи економічній стабільності.

Таким чином, актуальним є створення програмного забезпечення для прогнозування ІСЦ, яке ґрунтується на модифікованому підході, що поєднує статистичні методи та машинне навчання. Це дозволить підвищити точність прогнозів, враховуючи вплив факторів нестабільності в економіці. Таке програмне рішення забезпечить більш гнучке і комплексне прогнозування ІСЦ з підвищеною точністю.

2. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПРОГНОЗУВАННЯ ІНДЕКСУ СПОЖИВЧИХ ЦІН В УМОВАХ НЕСТАБІЛЬНОЇ ЕКОНОМІКИ

2.1 Аналіз збору даних для прогнозування індексу споживчих цін

Для прогнозування індексу споживчих цін ІСЦ важливо зосередитися на таких джерелах історичних даних та екзогенних змінних.

Національний банк України НБУ та Державна служба статистики України: ці установи публікують актуальні макроекономічні дані, включаючи інформацію про ІСЦ, що може бути корисним для аналізу фінансової стабільності та монетарної політики.

World Bank: надає різноманітні економічні індикатори по Україні (рис. 2.1), включаючи дані про ІСЦ, що можуть бути використані для дослідження.

Trading Economics: пропонує велику кількість економічних даних, включаючи ІСЦ та інші важливі індикатори, що впливають на економіку України.

Отримання та обробка цих даних може здійснюватися за допомогою різних методів, таких як API, вебскрапінг або завантаження даних з локальних джерел. Хоча вебскрапінг є ефективним для екстракції даних з вебсторінок, він має певні обмеження, пов'язані зі стабільністю вебінтерфейсу та можливим обмеженням доступом. Тому зазвичай більш практичним і надійним варіантом є використання API, які надають дані в зручних для обробки форматах, таких як JSON або XML. Приклади отримання даних у форматах JSON або XML ілюструються на рисунках 2.2 та 2.3, що спрощує доступ до історичних даних для аналізу ІСЦ.

Використання цих методів забезпечує адаптивність та ефективність у зборі та аналізі даних, що є критично важливим для точного прогнозування індексу споживчих цін та проведення економічних досліджень.

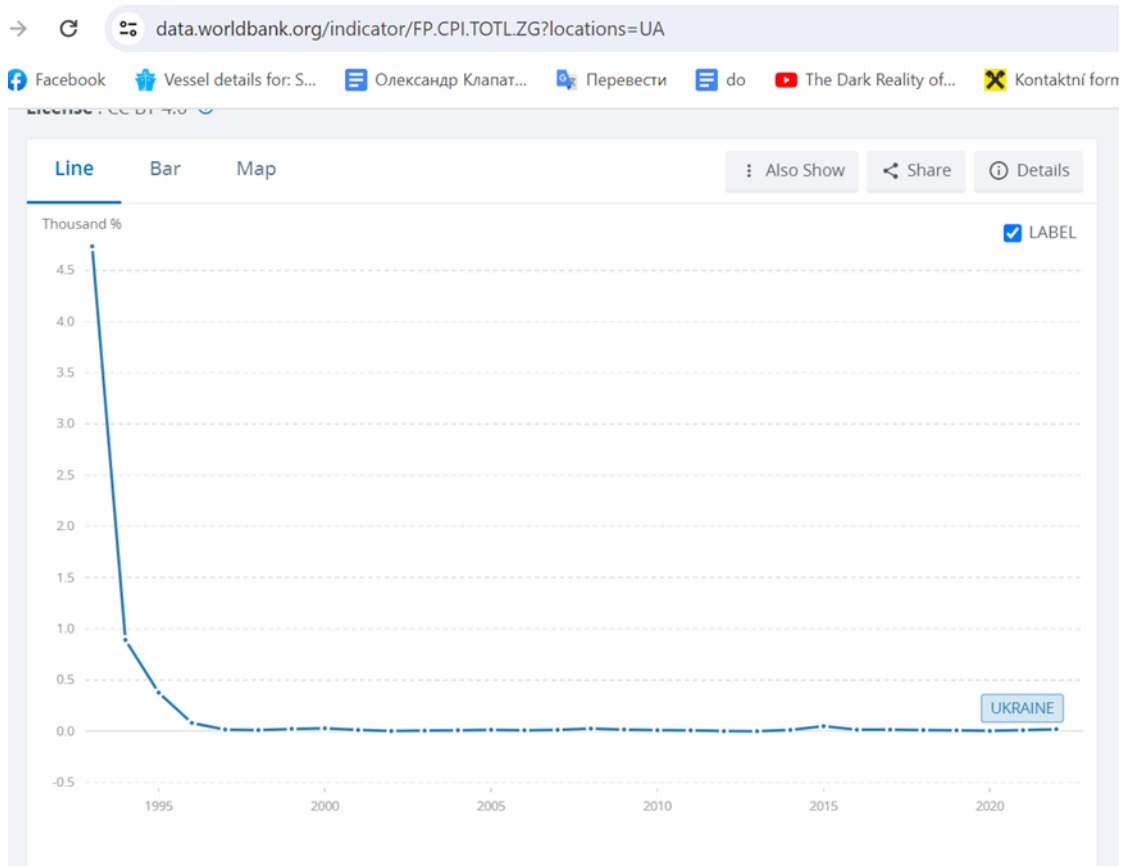


Рисунок 2.1— Дані про ІСЦ в Україні на сайті «worldbank.org»

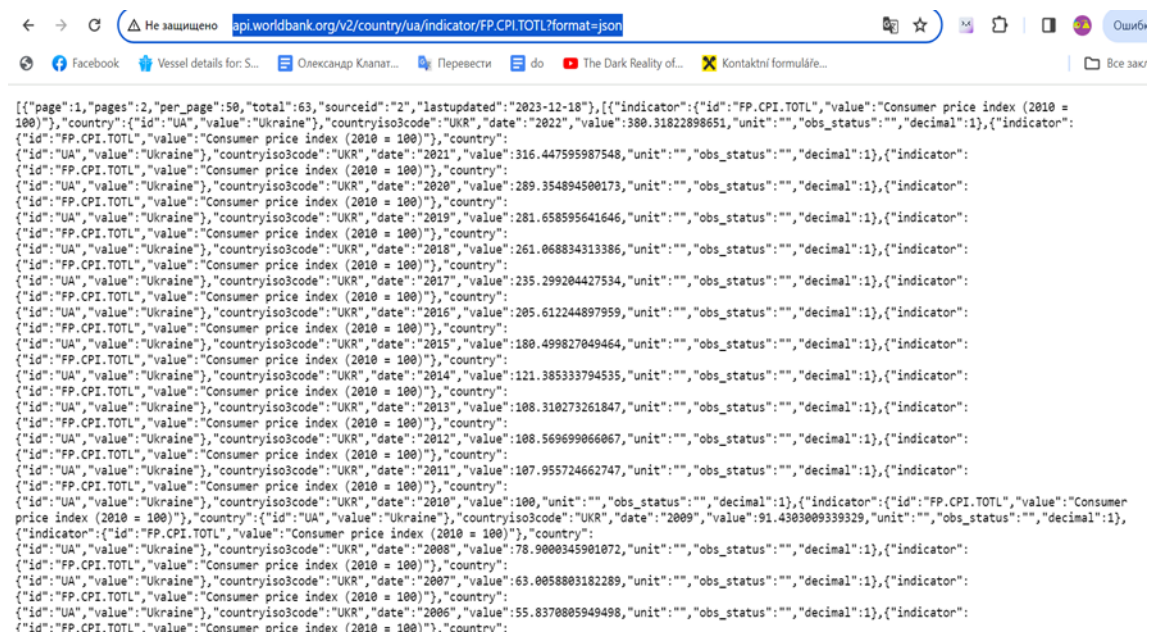


Рисунок 2.2 — Отримання даних від API сайту «worldbank.org»

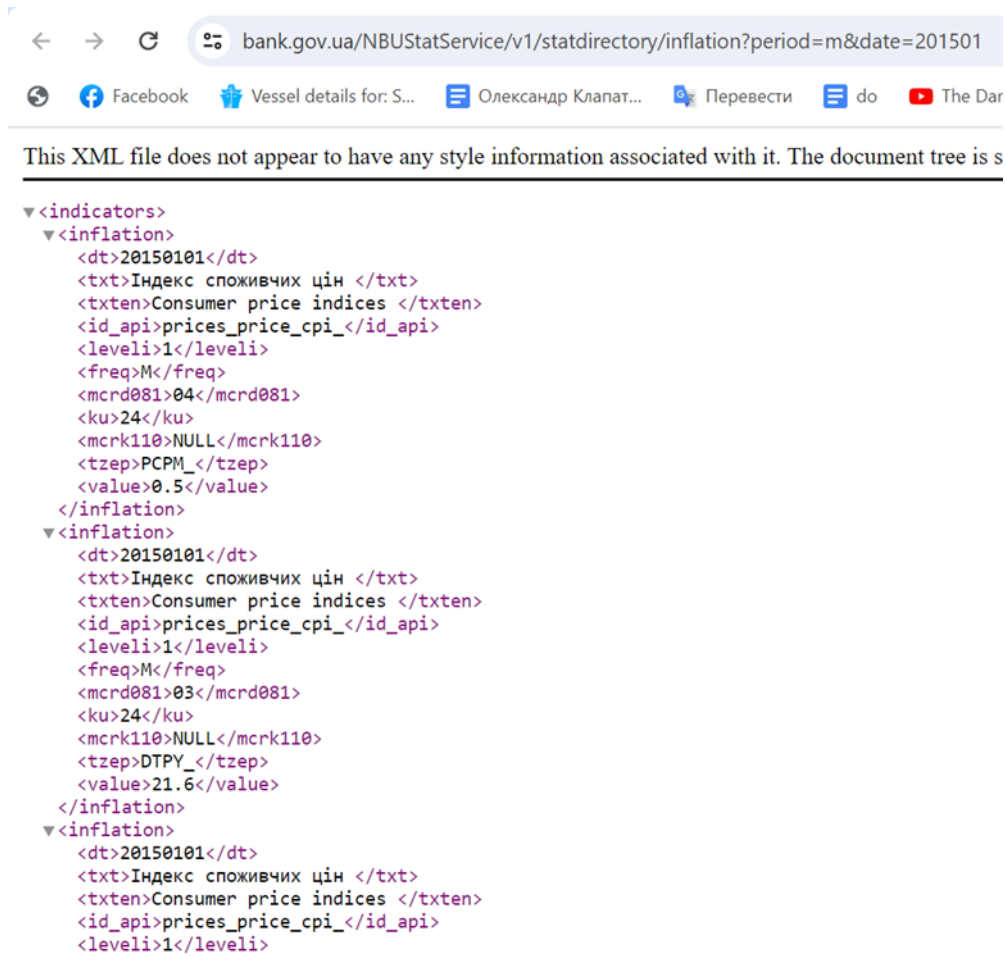


Рисунок 2.3 — Отримання даних від API сайту НБУ

API World Bank: Щоб отримати дані про індекс споживчих цін (ІСЦ) через API World Bank, користувачі повинні вказати конкретні параметри запиту, які ідентифікують необхідні економічні індикатори. API World Bank зазвичай надає дані у форматі JSON, що є зручним для читання і легко інтегрується в більшість мов програмування та платформ. Важливою перевагою є можливість доступу до широкого спектру індикаторів, що робить цей API корисним для аналізу та порівняння економічних тенденцій.

API Національного банку України (НБУ): API НБУ пропонує детальну інформацію про макроекономічні показники, зокрема про ІСЦ та інші метрики, які можуть бути корисними в рамках дослідження. Використання API НБУ потребує знання точних параметрів запиту та, можливо, API ключа для доступу. Дані зазвичай надаються у форматах JSON або XML, залежно

від специфікацій API, і потребують подальшої обробки для аналітичних цілей.

2.2 Аналіз математичних моделей прогнозування часових рядів у прогнозуванні індексу споживчих цін

Модель ARIMA (Автогресивна Інтегрована Ковзна Середня) є розширенням моделі ARMA, яка поєднує авторегресійні (AR) і ковзні середні (MA) компоненти, додаючи інтеграційний елемент (I) для роботи з нестационарними часовими рядами. Модель ARIMA порядку (p, d, q) можна описати рівнянням [16]:

$$(1 - B)^d X_t = C + \phi_1 X_{t-1} + \dots + \phi_p X_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q},$$

де B є оператор зміщення (наприклад, $BX_t = X_{t-1}$), а d є порядком інтеграції.

Модель ARIMA особливо корисна для прогнозування індексу споживчих цін (ІСЦ), оскільки дозволяє працювати з нестационарними даними. Однак, правильний вибір параметрів p, d і q є критичним і може бути складним завданням.

Модель SARIMA (Сезонна Автогресивна Інтегрована Ковзна Середня) розширює ARIMA, додаючи сезонні компоненти для обліку повторюваних шаблонів у даних. SARIMA модель порядку $(p, d, q)(P, D, Q)$ визначається наступним рівнянням [17]:

$$(1 - B^S)^D X_t = c + (\phi_1 B + \dots + \phi_p B^p)(1 - \theta B^S)^D \varepsilon_t + (\theta_1 B + \dots + \theta_q B^q)(1 - \theta B^S)^D \varepsilon_t,$$

де S є періодичністю сезону, а (P, D, Q) є порядками сезонних компонентів

Модель SARIMA враховує сезонні варіації в даних, що дозволяє поліпшити точність прогнозування в умовах наявності циклічних коливань. Це робить її корисною для роботи з часовими рядами, які мають виражену сезонність. Проте ця модель може бути складною для оцінки та інтерпретації, особливо коли існує висока сезонність або наявні кілька різних сезонних компонентів.

ARIMAX – це авторегресійна інтегрована модель з ковзним середнім, яка враховує вплив екзогенних змінних на прогнозування. Модель ARIMAX порядку (p, d, q) з екзогенними змінними формулюється наступним чином [19]:

$$(1 - B)^d X_t = C + \phi_1 X_{t-1} + \dots + \phi_p X_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q} + \beta_1 Z_{i,t} + \dots + \beta_n Z_{n,t},$$

де $Z_{i,t}$ – екзогенні змінні.

Модель ARIMAX враховує екзогенні змінні, що дозволяє їй краще адаптуватися до зовнішніх факторів, які впливають на інфляцію. Проте включення екзогенних змінних може ускладнити процес моделювання та вибір оптимальних параметрів. Крім того, виникає потреба в наявності додаткових даних. Для використання моделі ARIMAX необхідно мати доступ до відповідних даних про екзогенні змінні, що може бути викликом у деяких випадках.

Модель SARIMAX, що є розширенням SARIMA, інтегрує екзогенні змінні для покращення прогнозування. SARIMAX модель порядку $(p, d, q)(P, D, Q)$ з екзогенними змінними формується наступним чином [19]:

$$(1 - B)^d (1 - B^S)^D X_t = c + (\phi_1 B + \dots + \phi_p B^p)(1 - \theta B^S)^D \varepsilon_t + (\theta_1 B + \dots + \theta_q B^q)(1 - \theta B^S)^D \varepsilon_t + \beta_1 Z_{i,t} + \dots + \beta_n Z_{n,t}.$$

Модель SARIMAX має такі переваги в прогнозуванні індексу споживчих цін (ІСЦ):

Урахування сезонності та екзогенних факторів. Модель дозволяє моделювати сезонні коливання в інфляційних даних і включати зовнішні фактори, що впливають на економіку.

Підвищена точність. Інтеграція екзогенних змінних сприяє більш точним прогнозам, адаптованим до специфічних умов.

Проте, SARIMAX також має деякі недоліки в прогнозуванні ІСЦ:

Складність та вимоги до ресурсів. Модель може бути вимогливою з точки зору обчислювальних ресурсів і необхідних даних.

Потенційна перекомплікація моделі. Є ризик, що модель може стати занадто складною, що призведе до перенавчання і зниження її загальної ефективності [20].

ARIMAX та SARIMAX є розширеннями традиційних моделей ARIMA та SARIMA, які інтегрують зовнішні (екзогенні) змінні. Це надає моделям більшої адаптивності та підвищує їх точність.

2.3 Обґрунтування вибору техніки машинного навчання **Gradient Boosting** у прогнозуванні індексу споживчих цін

Gradient boosting — це метод машинного навчання, який використовує ансамбль слабких моделей (зазвичай дерев рішень) для створення сильної комбінованої моделі [21]. Процес роботи градієнтного бустингу можна описати етапами. Навчання першої моделі на даних для прогнозування цільової змінної. Обчислення залишків — різниці між передбаченнями моделі та фактичними значеннями цільової змінної. Навчання наступної моделі, яка намагається передбачити ці залишки. Повторення процесу кілька разів, де на кожному кроці нова модель навчається на залишках попередніх моделей. Інальне прогнозування є сумою передбачень усіх моделей.

Формульно позначимо:

- $F_m(x)$ — передбачення на m -тому кроці.
- y — справжній таргет.
- ρ_m — швидкість навчання.

Тоді, залишки на m -тому кроці обчислюються як:

$$r_m = y - F_{m-1}(x) .$$

Нова модель намагається передбачити залишки, а її передбачення додаються до попередніх прогнозів з урахуванням швидкості навчання. Формально це можна виразити так:

$$F_m(x) = F_{m-1}(x) + \rho_m \times model_m(x) .$$

Переваги:

- Висока точність. Часто забезпечує кращі результати в порівнянні з іншими методами.
- Моделює нелінійні залежності. Ефективно виявляє складні зв'язки в даних.
- Важливість ознак. Може автоматично оцінювати важливість різних економічних показників.
- Регуляризація. Включає вбудовану регуляризацію для запобігання перенавчанню.

Недоліки:

- Час навчання. Моделі Gradient Boosting можуть потребувати багато часу для навчання.
- Налаштування. Вимагає детального налаштування гіперпараметрів для досягнення оптимальних результатів.
- Складність інтерпретації. Хоча можна визначити важливість змінних, сама модель є менш інтерпретованою в порівнянні з простим деревом рішень.

Gradient Boosting може бути особливо ефективним для прогнозування ІСЦ завдяки своїй здатності моделювати складні залежності в даних. Проте його використання потребує глибокого розуміння та досвіду в налаштуванні моделі.

2.4 Розробка алгоритму роботи програмного забезпечення індексу споживчих цін

Розроблений модифікований спосіб зосереджується на виявленні та використанні глибинних закономірностей в історичних даних індексу

споживчих цін (ІСЦ). Цей підхід інтегрує різноманітні екзогенні змінні та застосовує алгоритми машинного навчання для підвищення точності прогнозів. Основні етапи цього методу включають:

1. Первинний аналіз даних. Вивчення та очищення даних для подальшого аналізу.
2. Автокореляційний аналіз. Оцінка залежностей в даних та виявлення патернів.
3. Перевірка на мультиколінеарність. Аналіз кореляцій між незалежними змінними для запобігання проблемам у моделюванні.
4. Вибір моделі прогнозування (ARIMAX або SARIMAX) та оптимізація. Визначення найкращої моделі для даних та її налаштування.
5. Уточнення прогнозу за допомогою Gradient Boosting. Поглиблення прогнозування з урахуванням нових факторів і залежностей.

Цей підхід суттєво підвищує наше розуміння даних та взаємозв'язків у часових рядах, що дозволяє глибше аналізувати особливості та тенденції, які в них виявляються. Він також є ключовим для створення надійної та точної моделі прогнозування, здатної адекватно оцінювати майбутні тренди та патери. Завдяки цьому методу ми можемо не лише виявляти закономірності в історичних даних, а й ефективно використовувати ці знання для прогнозування майбутніх подій.

Наступним кроком буде детальний огляд конкретного методу, який застосовується в цьому процесі, що дозволить краще усвідомити його переваги та потенційні сфери використання.

Зобразимо алгоритм за допомогою діаграми діяльності:

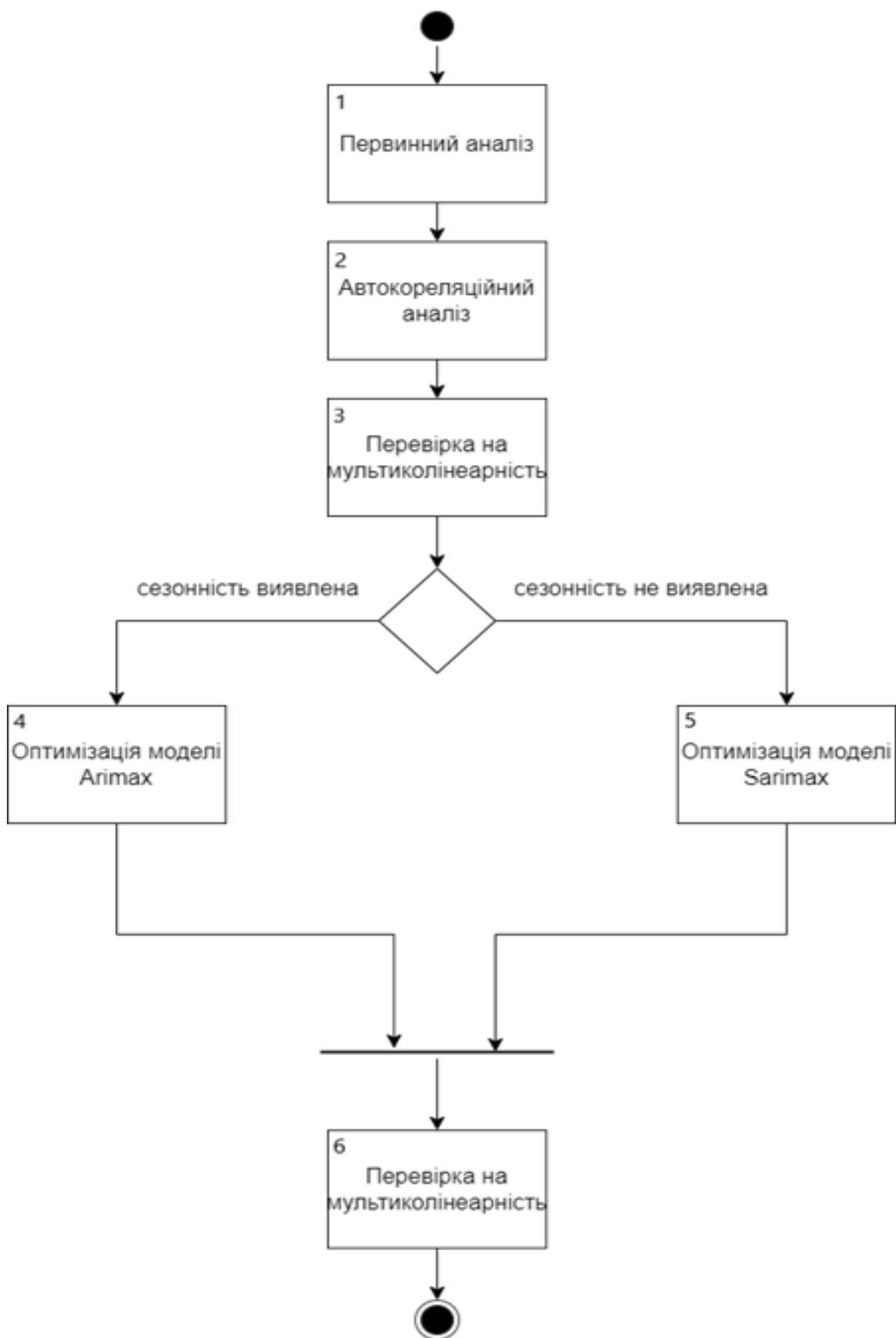


Рисунок 2.3 — Алгоритм програмного прогнозування індексу споживчих цін

Первинний аналіз даних.

Перевірка стаціонарності часових рядів має на меті визначити, чи є ряд стаціонарним. Якщо він не стаціонарний, можна застосувати різницю для його перетворення в стаціонарний.

Основна гіпотеза тесту полягає в тому, що ряд має одиничний корінь, що свідчить про не стаціонарність. Якщо гіпотезу відхиляють, це вказує на стаціонарність ряду [22].

$$\Delta Y_t = a + \beta t + \gamma Y_{t-1} + \delta_1 \Delta Y_{t-1} + \epsilon_t,$$

де ΔY_t – перша різниця ряду, a – константа, β – коефіцієнт при тренді, γ – коефіцієнт при затримці ряду, δ_1 – коефіцієнт при затримці першої різниці, і ϵ_t – білий шум.

Розкладання часових рядів

Виділення тренда, сезонності та залишків передбачає розподіл часових рядів на три основні компоненти: тренд, сезонність та залишки. Це дозволяє алгоритму більш точно моделювати кожен з цих компонентів окремо.

Розкладення часового ряду (STL):

$$Y_t = T_t + S_t + R_t,$$

де Y_t – спостереження в момент часу t , T_t – трендова компонента, S_t – сезонна компонента, і R_t – залишкова (випадкова) компонента. Розглянемо більш детально:

Трендова компонента відображає довгострокову тенденцію в часовому ряду, яка може мати як лінійний, так і нелінійний характер. Вона допомагає визначити загальний напрямок руху даних протягом тривалого періоду.

Сезонна компонента показує сезонні коливання або патерни в даних, що повторюються через певні регулярні інтервали часу, такі як щомісячні чи щорічні.

Залишкова компонента охоплює випадкові або непередбачувані коливання, які не піддаються поясненню через трендову або сезонну компоненти. Ця складова відображає елементи "шуму" в часовому ряду [23].

Наведений вище процес здійснює більш точний аналіз і прогнозування поведінки часового ряду, розділяючи його на зрозумілі та доступні для аналізу компоненти. Це дозволяє більш ефективно моделювати індекс споживчих цін, окремо враховуючи вплив тренду, сезонності та випадкових коливань. Алгоритм роботи первинного аналізу зображений на рисунку 2.3.

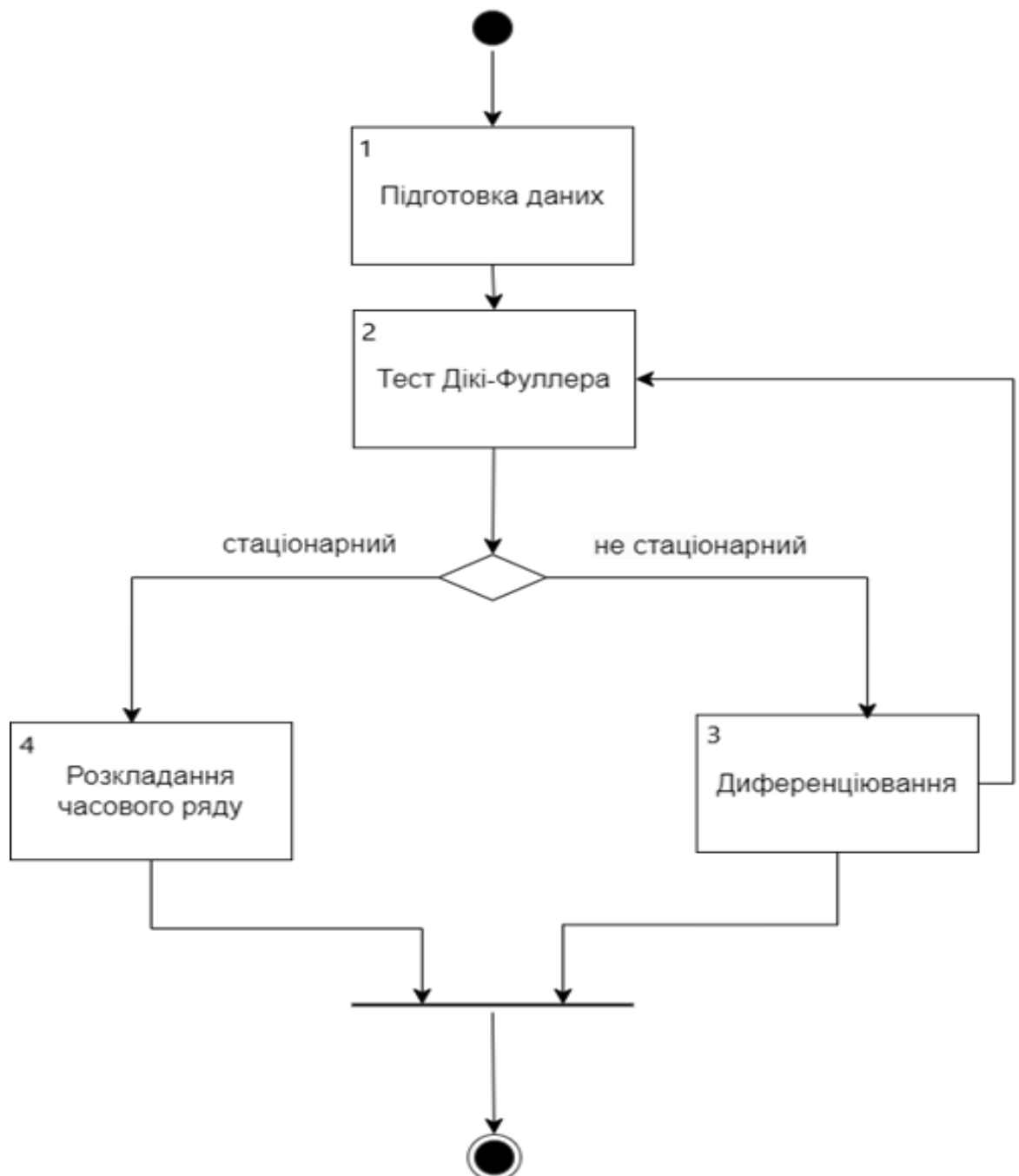


Рисунок 2.3 — Алгоритм первинного аналізу даних

Автокореляційний аналіз

Автокореляційний аналіз дозволяє виявити автокореляційні залежності в часовому ряді, що є важливими для визначення авторегресійних компонентів у моделі.

Автокореляційна функція (ACF):

$$ACF(k) = \frac{\sum_{t=k+1}^N (Y_t - \bar{Y})(Y_{t-k} - \bar{Y})}{\sum_{t=1}^N (Y_t - \bar{Y})^2},$$

де k – лаг автокореляції, N – кількість спостережень, і $\bar{Y}$ – середнє значення часового ряду [24].

ACF (автокореляційна функція) вимірює зв'язок між спостереженнями в часовому ряді з різницею в кілька періодів. Це є важливим для визначення порядку авторегресії в моделі ARIMA.

Використання ACF в моделях ARIMAX або SARIMAX допомагає встановити структуру авторегресійних (AR) та ковзаючих середніх (MA) компонентів моделі, оскільки воно демонструє, наскільки сильні і тривалі автокореляційні залежності. Це критично для вибору відповідних параметрів моделі, що дозволяє зробити її більш точною та адаптованою до специфіки даних індексу споживчих цін.

Перевірка на мультиколінеарність.

Варіаційний інфляційний фактор (ВІФ) є інструментом, що застосовується для вимірювання ступеня мультиколінеарності у наборі множинних змінних регресії. Він оцінює, наскільки дисперсія однієї незалежної змінної підвищується через її взаємодію чи кореляцію з іншими незалежними змінними. ВІФ використовується в класичній регресії найменших квадратів, де він відображає масштаб мультиколінеарності в моделі. Високі значення ВІФ для певних змінних свідчать про значну мультиколінеарність, що вимагає корекції або комбінування колінеарних змінних [25].

Формула ВІФ виглядає наступним чином:

$$VIF_i = \frac{1}{1-R_i^2},$$

де VIF_i – Варіаційний Інфляційний Фактор для i -ої незалежної змінної, а R_i^2 – коефіцієнт детермінації, отриманий з регресії i -ої незалежної змінної на всі інші незалежні змінні.

Цей коефіцієнт демонструє, наскільки добре i -та незалежна змінна може бути описана іншими незалежними змінними в моделі. Чим вищий ВІФ, тим більшою є мультиколінеарність, а отже, більша ймовірність того, що ця змінна залежить від інших змінних у моделі [26].

Вибір моделі прогнозування та оптимізація моделі.

Вибір моделі для прогнозування здійснюється в залежності від виявленої сезонності: використовується модель ARIMAX, якщо сезонність відсутня, і SARIMAX, якщо вона присутня. Обидві моделі оптимізуються за критерієм Акаїке (AIC). За допомогою AIC визначаються оптимальні параметри моделі, причому чим нижче значення AIC, тим кращою вважається модель. AIC слугує критерієм якості моделі, вимірюючи відносну кількість інформації, яку втрачає модель.

Для даної моделі він визначається так:

$$AIC = 2k - 2\ln(L),$$

де k – кількість параметрів моделі, L – максимальне значення функції правдоподібності для моделі [27].

У нашому підході до прогнозування індексу споживчих цін застосування критерію Акаїке (AIC) є критично важливим, оскільки він не тільки винагороджує моделі за високу якість апроксимації, але й накладає штрафи за надмірне використання параметрів, що допомагає уникнути перенавчання. Це дозволяє знайти оптимальну рівновагу між точністю прогнозування та складністю моделі.

Варто зазначити, що модель з найменшим значенням AIC зазвичай вважається найефективнішою, оскільки вона досягає найкращого балансу між точністю прогнозування та кількістю використовуваних параметрів. Це,

в свою чергу, зменшує ризик перенавчання моделі і підвищує її прогностичні можливості.

Отже, у контексті моделей ARIMAX або SARIMAX, які можуть включати різні екзогенні змінні та мати складні структури, критерій Акаїке допомагає знайти оптимальний баланс між складністю моделі і її здатністю точно прогнозувати індекс споживчих цін.

Прогнозування з Gradient Boosting

Прогнозування з використанням Gradient Boosting дозволяє виявити нелінійні залежності та уточнити прогноз для залишкових компонентів (залишків). Цей процес можна розділити на кілька етапів:

1. Створення базової моделі. На першому етапі розробляється проста базова модель, яка прогнозує цільову змінну, в даному випадку — індекс споживчих цін.

2. Обчислення залишків. Наступним кроком є обчислення залишкових компонентів — різниці між фактичними і прогнозованими значеннями індексу споживчих цін. Ці залишки відображають похибки, які базова модель не змогла врахувати.

3. Побудова нової моделі для залишків. Далі створюється нова модель, яка прогнозує невраховані залишки. Ця модель фокусується на виправленні похибок, допущених попередньою моделлю.

4. Оновлення прогнозу. Прогнози нової моделі додаються до прогнозів базової моделі, що покращує загальний прогноз індексу споживчих цін. Цей процес ітеративно повторюється кілька разів, де кожна наступна модель намагається виправити залишкові похибки попередньої, поступово підвищуючи точність загального прогнозу.

5. Комбінування прогнозів. На останньому етапі всі моделі комбінуються для формування фінального прогнозу. Оскільки кожна модель спрямована на виправлення похибок попередніх моделей, комбінований прогноз зазвичай є більш точним і надійним.

Алгоритм прогнозування з Gradient Boosting зображено на рисунку 2.4.

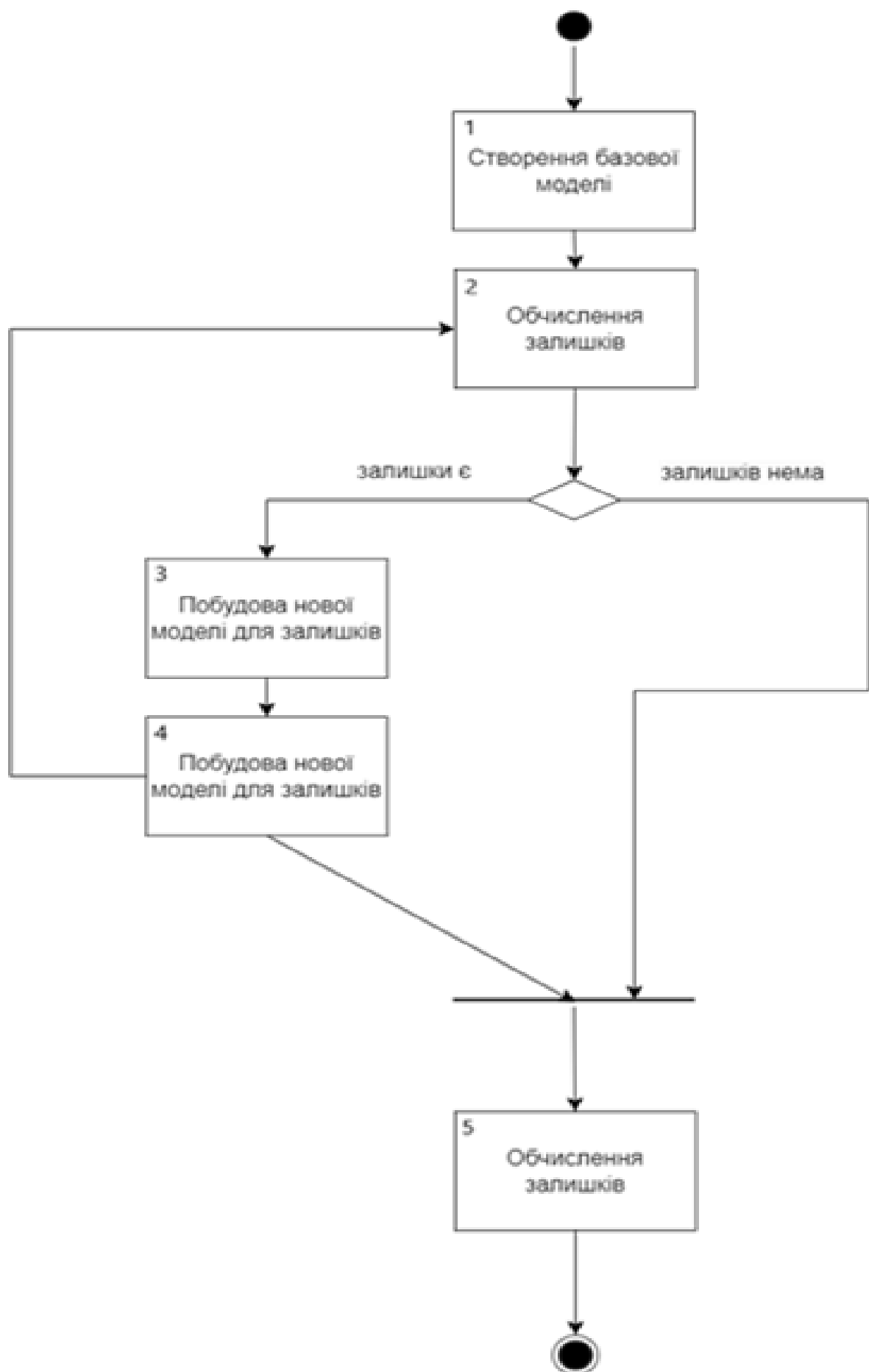


Рисунок 2.4 — Алгоритм прогнозування з Gradient Boosting

Особливості застосування Gradient Boosting на цьому етапі включають:

Фокус на залишкових компонентах. Gradient Boosting дозволяє зосередитися на складних аспектах даних, які були пропущені або недостатньо враховані в попередніх моделях, таким чином підвищуючи загальну точність прогнозування.

Підвищення точності прогнозування. Метод підсилює загальний прогноз, враховуючи недоліки попередньої моделі і коригуючи її похибки.

Адаптивність. Gradient Boosting може адаптуватися до різних шаблонів і змін у даних, що робить його ефективним інструментом для поліпшення прогнозів у динамічних економічних умовах.

Цей етап є ключовим для підвищення точності кінцевого прогнозу індексу споживчих цін, адже враховує всі потенційні фактори та динаміку, які можуть на нього впливати.

2.5 Висновки до розділу 2

Джерелами актуальних історичних даних про ІСЦ та інші економічні метрики, необхідні для точного прогнозування в Україні, є: Національний банк України, Державна служба статистики України, Світовий банк та Trading Economics. Ці організації регулярно публікують оновлені статистичні дані про макроекономічні показники, включаючи ІСЦ.

Доступ до даних може бути здійснений через API, вебскрапінг або завантаження з локальних джерел. Найбільш надійним і ефективним підходом є використання API, які надають дані в структурованих форматах, таких як JSON чи XML. Отримані дані ІСЦ та супутні економічні індикатори можуть бути використані для точного прогнозування ІСЦ в Україні завдяки запропонованому модифікованому методу програмного прогнозування.

Запропонований підхід об'єднує сучасні статистичні методи та алгоритми машинного навчання для підвищення точності прогнозів.

Основні етапи розробленого методу включають: збір та попередня обробка даних, аналіз стаціонарності та декомпозиція часових рядів, автокореляційний аналіз, побудова оптимізованих за критерієм Акаїке моделей ARIMAX або SARIMAX, перевірка мультиколінеарності екзогенних факторів, навчання та тестування моделей, а також покращення базового прогнозу за допомогою Gradient boosting.

Перевагами цього підходу є врахування специфіки даних ІСЦ та використання сучасних статистичних і машинних алгоритмів, що дозволяє підвищити точність прогнозування. Впровадження запропонованого методу може суттєво покращити якість аналізу та прогнозування макроекономічних показників, що сприятиме економічній стабільності країни.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНДЕКСУ СПОЖИВЧИХ ЦІН В УМОВАХ НЕ СТАБІЛЬНОЇ ЕКОНОМІКИ

3.1 Обґрунтування вибору мови програмування та засобів розробки

При розробці веб-застосунку ключовим етапом є вибір відповідних технологій, зокрема мов програмування, клієнтських і серверних фреймворків, а також систем управління базами даних (СУБД). Цей вибір суттєво вплине на продуктивність, безпеку та масштабованість готового продукту.

1. Вибір мови програмування для веб-застосунку.

Мова програмування повинна задовольняти кільком критеріям: гнучкість для створення складних веб-застосунків, багатий вибір доступних бібліотек, постійна підтримка та висока продуктивність. JavaScript виділяється як найкращий вибір завдяки своїй універсальності, великій спільноті та безлічі бібліотек.

JavaScript — це мова програмування, яка стала стандартом для створення інтерактивних вебзастосунків [30]. Вона характеризується широкою підтримкою та регулярними оновленнями. Розглянемо детальніше її переваги та недоліки

Переваги JavaScript:

Універсальність. JavaScript можна використовувати як для клієнтської, так і для серверної частини (через Node.js), що дозволяє розробникам працювати з однією мовою на всіх етапах розробки.

Інтерактивність веб-інтерфейсів. Завдяки JavaScript можливо створювати швидкі та чуйні користувацькі інтерфейси для вебзастосунків.

Велика кількість фреймворків та бібліотек. Існує безліч фреймворків, таких як React, Vue та Angular, які значно полегшують розробку складних вебзастосунків.

Активна спільнота. JavaScript має одну з найбільших і найактивніших спільнот, що забезпечує безліч ресурсів для навчання та підтримки.

Недоліки JavaScript:

Фрагментація. Різноманіття фреймворків та бібліотек може призвести до фрагментації навичок, а також технічних труднощів при переході з одного фреймворку на інший.

Проблеми з безпекою. Оскільки JavaScript працює на клієнтській стороні, він піддається ризикам, таким як крос-сайтове скриптування (XSS) та іншим вебзагрозам.

Непослідовність у браузерях. Різні браузери можуть по-різному інтерпретувати JavaScript-код, що може викликати проблеми сумісності.

Динамічна типізація. Хоча динамічна типізація забезпечує гнучкість, вона також може призвести до труднощів у виявленні помилок на етапі розробки.

У загальному, JavaScript є відмінним вибором для створення інтерактивних вебзастосунків, пропонуючи розробникам широкі можливості. Проте слід враховувати потенційні виклики, пов'язані з безпекою та сумісністю.

З урахуванням специфіки веб-розробки, JavaScript виділяється як найкращий вибір завдяки своїй універсальності, великій спільноті та безлічі бібліотек. Основні переваги JavaScript, які підтверджують цей вибір, включають:

Універсальність. JavaScript може використовуватися для розробки як фронтенду, так і бекенду, що дозволяє розробникам працювати з однією мовою на всіх етапах розробки вебзастосунку.

Велика спільнота. JavaScript має одну з найбільших спільнот розробників, що забезпечує доступ до великої кількості ресурсів для навчання, підтримки та співпраці.

Багатство бібліотек та фреймворків. Існує величезна кількість відкритих бібліотек та фреймворків для JavaScript, які спрощують і пришвидшують процес розробки.

Інтеграція з технологіями веброзробки. JavaScript легко інтегрується з HTML та CSS, що є основою для створення динамічних вебсторінок.

Підтримка сучасних вебстандартів. JavaScript постійно оновлюється, щоб відповідати новітнім вебстандартам, що гарантує його сумісність із сучасними технологіями.

Можливість створення реактивних та інтерактивних інтерфейсів. JavaScript дозволяє розробникам створювати швидкі та відгукуючі інтерфейси користувача, підвищуючи зручність використання вебзастосунків.

Сумісність з браузерами. JavaScript підтримується всіма сучасними веб-браузерами, що робить його ідеальним для кросплатформної розробки.

Ці переваги роблять JavaScript відмінним вибором для розробки вебзастосунків, які вимагають швидкості, гнучкості та високої інтерактивності. Завдяки цим факторам, JavaScript визначається як ключовий інструмент у сучасній веброзробці.

2. Вибір мови для модуля прогнозування

Для впровадження модифікованого методу прогнозування індексу споживчих цін (ІСЦ) важливо вибрати мову програмування, що найкраще відповідатиме специфіці вашого проекту, враховуючи потреби в обробці даних, продуктивність та зручність розробки. Розглянемо потенційні мови програмування, які можуть бути корисними для цього проекту.

Python — це високоабстрактна, інтерпретована мова програмування, яка славиться своєю читабельністю та гнучкістю [31]. Вона підтримує різні парадигми програмування, такі як об'єктно-орієнтоване, процедурне та функціональне програмування. Python широко використовується для розробки вебдодатків, автоматизації завдань, аналізу даних, штучного інтелекту та багатьох інших напрямків. Однією з ключових переваг є велике

співтовариство та різноманітність бібліотек, які роблять його надзвичайно потужним інструментом.

Серед переваг Python можна відзначити наявність багатьох бібліотек для аналізу даних і машинного навчання, таких як Pandas, NumPy, SciPy, Scikit-learn, TensorFlow та Keras. Ці інструменти забезпечують все необхідне для обробки та аналізу даних, що є критично важливим для прогнозування ІСЦ.

Крім того, простота синтаксису Python, а також велика спільнота сприяють швидкому розвитку проектів і легкості в пошуку рішень. Це робить Python ідеальним вибором для швидкого прототипування та реалізації складних алгоритмів.

Однак, варто зазначити, що Python може не бути найшвидшим рішенням для виконання обчислень у порівнянні з компільованими мовами, такими як C++ або Java. Хоча цей недолік частково компенсується завдяки використанню оптимізованих бібліотек, які можуть значно підвищити продуктивність під час виконання складних обчислень.

Таким чином, Python є потужним та зручним вибором для проектів, пов'язаних із прогнозуванням індексу споживчих цін, завдяки своїй гнучкості, широкій підтримці бібліотек та активній спільноті.

R — це мова програмування та середовище для статистичних обчислень і візуалізації даних, широко використовуване серед статистиків і аналітиків даних для аналізу інформації та розробки статистичного програмного забезпечення. R славиться своїми можливостями у візуалізації даних та має велику кількість пакетів для спеціалізованого статистичного аналізу [31].

Основною перевагою R є її фокус на статистичному аналізі та візуалізації, що робить її ідеальним інструментом для статистичного моделювання та аналізу трендів, які є ключовими для прогнозування індексу споживчих цін (ІСЦ). Мова має численні пакети, які дозволяють

проводити спеціалізований аналіз даних, що полегшує роботу з великими обсягами інформації.

Проте, одним з недоліків R є те, що вона менш зручна для загального програмування та розробки великих програмних систем. Як і Python, R може демонструвати знижену продуктивність для деяких типів обчислень [32].

Scala — це високорівнева мова програмування, яка підтримує об'єктно-орієнтоване та функціональне програмування. Вона розроблена для лаконічності та ефективності, надаючи потужні можливості для розробки програмного забезпечення. Scala працює на Java Virtual Machine (JVM) та є сумісною з Java, що робить її популярним вибором для веброзробки та великих додатків [33].

Перевага Scala полягає в її здатності обробляти великі дані завдяки інтеграції з Apache Spark, що може бути особливо корисно, якщо проект передбачає роботу з великими обсягами інформації або вимагає масштабування.

Однак, недоліком Scala є її складність у вивченні та використанні, особливо для тих, хто не знайомий із концепціями функціонального програмування.

Для нашого проекту, що передбачає реалізацію модифікованого способу прогнозування індексу споживчих цін, Python виявляється найкращим вибором. Ця мова програмування ідеально підходить для обробки та аналізу даних завдяки своїй широкій екосистемі бібліотек, які спеціалізуються на статистиці, машинному навчанні та візуалізації. Python забезпечує необхідну обчислювальну продуктивність і залишається легким у вивченні та використанні, що робить його ідеальним для нашого дослідження.

3. Вибір Фреймворків для frontend частини вебзастосунку

При виборі фреймворків для frontend частини вебзастосунку основними варіантами є Vue, React та Angular. Давайте розглянемо ключові переваги та недоліки кожного з цих популярних фреймворків.

Vue.js, або просто Vue, — це прогресивний JavaScript фреймворк, призначений для побудови інтерфейсів користувача та односторінкових вебдодатків (SPA). Vue спроектований для поступової інтеграції та легко впроваджується в існуючі проекти [34].

Переваги:

- Легкість навчання та використання: Vue простий у навчанні і має інтуїтивно зрозумілу структуру та документацію.
- Гнучка архітектура: Розробники можуть легко адаптувати фреймворк під конкретні потреби проекту.
- Легкість інтеграції: Vue можна легко інтегрувати з існуючими проектами та бібліотеками.

Недоліки:

- Обмежена підтримка: Порівняно з Angular та React, Vue має меншу підтримку від великих організацій.
- Спільнота: Хоча спільнота Vue швидко росте, вона все ще менша, що ускладнює пошук ресурсів.
- Складність інтеграції: Інколи інтеграція Vue з іншими бібліотеками може бути складнішою через обмежену підтримку.

Angular — це платформа та фреймворк для створення односторінкових вебдодатків, розроблений Google. Angular використовує HTML як шаблонну мову та розширює його синтаксис[35].

Переваги:

- Повнофункціональний фреймворк: Angular надає широкий набір інструментів для розробки, включаючи рутинг, форми та HTTP-клієнт.
- Строга типізація з TypeScript: Використання TypeScript покращує якість коду.
- Висока продуктивність: Angular оптимізований для побудови великих і складних додатків

Недоліки:

- Складність навчання: Angular має круту криву навчання через складну архітектуру.
- Продуктивність: Для деяких додатків Angular може бути важчим і повільнішим у виконанні.
- Часті оновлення: Angular оновлюється досить часто, що може вимагати регулярних змін у коді додатку.

React — це декларативна, ефективна та гнучка JavaScript бібліотека для створення інтерфейсів користувача [36]. React дозволяє розробникам складати складні інтерфейси з невеликих частин коду, званих "компонентами".

Переваги:

- Гнучкість та модульність: React дозволяє створювати адаптивні та повторно використовувані компоненти.
- Велика спільнота та екосистема: React має активну спільноту, що надає різноманітні бібліотеки та ресурси.
- Використання віртуального DOM: Це підвищує продуктивність та ефективність оновлень UI.

Недоліки:

- Інтеграція: React потребує додаткової інтеграції з іншими бібліотеками, що може ускладнити архітектуру.
- Документація: Через часті оновлення, документація може бути неповною або застарілою.
- Оптимізація продуктивності: Оптимізація великих додатків може бути складною задачею.

Враховуючи всі переваги, React виглядає як найкращий вибір для нашого проекту. Його гнучкість, можливість повторного використання компонентів і велика кількість навчальних ресурсів роблять його ідеальним для розробки інноваційного та ефективного інтерфейсу користувача.

4. Вибір фреймворків для backend частини вебзастосунку

При виборі фреймворків для backend частини вебзастосунку основними варіантами є Express.js, Koa.js, Hapi.js та Sails.js. Розглянемо ключові переваги та недоліки кожного з цих фреймворків.

Express.js — це популярний і гнучкий фреймворк для Node.js, призначений для створення вебдодатків і API [37]. Він пропонує набір потужних функцій для веб- та мобільних додатків, таких як маршрутизація та підтримка мідлварів (middleware).

Переваги:

- Велика популярність: Найбільш використовуваний Node.js фреймворк, що забезпечує широку спільноту та підтримку.
- Легкість у використанні: Простота налаштування та швидкість розробки завдяки мінімалістичному підходу.
- Гнучкість: Можливість інтеграції з багатьма плагінами та іншими бібліотеками Node.js.

Недоліки:

- Обмежені функції: Не має деяких вбудованих функцій у порівнянні з більш повнофункціональними фреймворками.
- Потреба в додаткових модулях: Часто потрібні додаткові модулі для розширення функціоналу.

Koa.js — це сучасний, легкий фреймворк для Node.js, створений тими ж людьми, що й Express [38]. Кoa спрощує створення вебдодатків і API, використовуючи новітні можливості JavaScript, такі як async/await.

Переваги:

- Модернізація: Використання новітніх можливостей ES6, таких як async/await, що полегшує написання асинхронного коду.
- Легкість та гнучкість: Мінімалістичний і гнучкий фреймворк, що дозволяє розробникам використовувати лише необхідні функції.

Недоліки:

- Менша спільнота: Не так широко використовується, як Express, що може обмежувати доступ до ресурсів та підтримки.

- Вимагає більше часу на навчання: Може потребувати більше часу для освоєння через нові підходи та концепції.

Нарі.js — це платформа для розробки вебдодатків і API, що використовує Node.js. Нарі відрізняється гнучкою конфігурацією та орієнтоване на створення API з операціями CRUD (створення, читання, оновлення, видалення) [39].

Переваги:

- Безпека: Стандартна конфігурація зорієнтована на безпеку.
- Широкий функціонал: Включає багато функцій "з коробки", що знижує потребу у додаткових плагінах.

Недоліки:

- Складність: Може бути перевантаженим для простих проектів.
- Менше ресурсів для навчання: Має менше навчальних матеріалів і гідів у порівнянні з іншими фреймворками.

Вибір **Express.js** для вашого проекту є раціональним рішенням, враховуючи його популярність, простоту використання та гнучкість. Це надає можливість швидко розвивати проект, маючи доступ до широкої спільноти та ресурсів.

5. Вибір СУБД

При виборі СУБД для вебзастосунків основними варіантами є PostgreSQL, MongoDB та MySQL. Розглянемо ключові переваги та недоліки кожної з цих СУБД.

PostgreSQL — це відкрита, об'єктно-реляційна система управління базами даних, відома своєю надійністю, розширеними функціями та високою продуктивністю[40].

Переваги:

- ACID-сумісність: Гарантує надійність транзакцій.
- Підтримка складних запитів: Відмінно підходить для аналітичних обчислень і складних запитів.

- Розширюваність: Підтримує безліч доповнень, включаючи геопросторові дані через PostGIS.
- Безпека: Міцні можливості управління доступом та безпеки.

Недоліки:

- Складність налаштування: Може вимагати більше часу та зусиль для налаштування в порівнянні з більш простими СУБД.
- Використання ресурсів: Може бути вимогливим до ресурсів при обробці великих обсягів даних або складних операцій.

MongoDB – це документо-орієнтована NoSQL база даних, що використовує формат JSON-подібних документів з динамічними схемами. Вона розроблена для забезпечення високої продуктивності, легкої масштабованості та гнучкості в управлінні даними[41].

Переваги:

- Гнучкість схем: Дозволяє зберігати документи без строгих вимог до структури даних.
- Масштабованість: Легко масштабується горизонтально, що робить її ідеальною для великих обсягів даних.
- Швидкість та продуктивність: Оптимізована для швидких запитів на великих обсягах даних.
- Інтуїтивно зрозумілий запит: Запити в MongoDB часто прості та інтуїтивно зрозумілі.

Недоліки:

- Транзакції: Тільки останні версії MongoDB підтримують транзакції, які можуть бути складними у використанні в порівнянні з SQL-системами.
- Споживання пам'яті: MongoDB може бути вимогливою до пам'яті, особливо при роботі з великими наборами даних.

MySQL – це популярна система управління реляційними базами даних, яка використовує SQL для зберігання, витягу та управління даними. MySQL є відкритим джерелом і широко використовується у вебзастосунках [42].

Переваги:

- Широка підтримка: Одна з найбільш популярних реляційних СУБД з великою спільнотою та підтримкою.
- Легкість використання: Проста у встановленні та налаштуванні, з інтуїтивно зрозумілим запитом.
- Висока продуктивність: Оптимізована для швидких читань та простих транзакцій.
- Висока сумісність: Легко інтегрується з багатьма мовами програмування та платформами.

Недоліки:

- Масштабованість: Може мати проблеми з горизонтальною масштабованістю порівняно з NoSQL базами даних.
- Обмеження функціоналу: Для дуже складних або великих баз даних може не надавати такого ж рівня функціональності, як деякі інші реляційні СУБД (наприклад, PostgreSQL).

Враховуючи вимоги до проекту програмної розробки модифікованого способу прогнозування індексу споживчих цін (ІСЦ), MongoDB може бути найкращим вибором через її адаптивність, легкість масштабування та ефективність у роботі з великими обсягами даних. Її гнучка схема даних дозволить швидко адаптуватися до змін у вимогах проекту, а велика спільнота та багатий набір ресурсів забезпечать необхідну підтримку та документацію.

Отже, обрані технології та засоби розробки забезпечують баланс між ефективністю розробки, легкістю вивчення та підтримки, формуючи гнучку і масштабовану архітектуру для вебзастосунку. JavaScript і React, у поєднанні з Express та MongoDB, створюють потужну основу для розробки,

тоді як Python забезпечує надійний інструментарій для розробки модулів аналізу даних.

3.2 Програмна реалізація індексу споживчих цін в умовах не стабільної економіки

Багатошарова архітектура (або многослойна) є поширеним підходом до розробки вебзастосунків, що дозволяє чітко розподілити відповідальності між різними компонентами системи. Це забезпечує легкість у підтримці, масштабуванні та управлінні проектом. У нашому випадку архітектура включає чотири основні шари: frontend, backend, модуль прогнозування та базу даних.

Frontend (React)

Frontend частина вебзастосунку виконує важливі функції, які забезпечують зручний і ефективний взаємодію користувача з системою. Основні відповідальності включають:

1. Графічний Інтерфейс Користувача (GUI)

- Реєстрація та Авторизація Користувачів: Frontend реалізує форми для введення даних, які дозволяють користувачам створювати облікові записи та входити в систему. Це включає в себе: Поля для введення логіна та пароля, валідацію даних, що вводяться, для перевірки їх коректності перед відправкою на сервер.
- Інтерфейс для Завантаження Даних: Забезпечення можливості для користувачів вибирати спосіб завантаження даних. Це може включати: Вибір файлів для завантаження, параметри для налаштування формату та типу даних.
- Візуалізація Результатів Прогнозування: Після обробки даних сервером, frontend відповідає за: Відображення отриманих результатів у зрозумілому та наочному форматі (графіки, таблиці, діаграми тощо),

динамічне оновлення інтерфейсу відповідно до результатів, що надходять.

2. Взаємодія з Сервером

- Відправка Запитів до Серверу: Frontend відповідає за формування і відправку HTTP-запитів до Express.js сервера для: Передачі даних користувача (реєстрація, авторизація, завантаження даних), вибору методу завантаження даних та інших параметрів, необхідних для обробки.
- Отримання Результатів Прогнозування: Після обробки запитів сервером, frontend обробляє відповіді: Отримує результати прогнозування, відображає їх у графічному інтерфейсі, забезпечуючи інтерактивність і швидку реакцію на дії користувача.

Ці відповідальності дозволяють забезпечити інтерактивний, зручний і функціональний інтерфейс для користувачів, сприяючи покращенню їх досвіду взаємодії з вебзастосунком.

Backend (Express.js)

Backend частина вебзастосунку виконує критично важливі функції для забезпечення логіки, обробки даних і комунікації між різними компонентами системи. Основні відповідальності включають:

1. API та Маршрутизація

- Обробка Запитів від Frontend: Backend відповідає за отримання запитів від frontend частини, аналізує їх і виконує відповідні дії, включаючи: Обробку запитів на реєстрацію та авторизацію користувачів, обробку запитів на завантаження даних та запитів на прогнозування.
- Автентифікація та Авторизація Користувачів: Реалізація механізмів безпеки для перевірки особи користувача. Це включає: Зберігання та перевірку облікових даних користувачів, видачу токенів доступу для підтвердження авторизації.

- Передача Даних для Прогнозування до Python Модуля: Збирання та обробка даних, необхідних для прогнозування, і передача їх до Python модуля. Це може включати: Форматування даних відповідно до вимог модулю, визначення необхідних параметрів для прогнозування.

2. Система Користувачів і Підписок

- Управління Користувацькими Обліковими Записами: Реалізація механізмів для створення, редагування та видалення облікових записів. Це включає: Валідацію даних, введених під час реєстрації. Зберігання та захист даних користувачів у базі даних.
- Управління Підписками: Забезпечення механізму для управління підписками користувачів на послуги або функції програми. Це включає: Створення, оновлення та скасування підписок, зберігання інформації про активність користувачів у системі.

3. Комунікація з Python Модулем

- Відправка Даних до Python Модуля: Передача зібраних даних до Python модуля для обробки прогнозування. Це включає: Виклик необхідних функцій у Python модулі, форматування запитів відповідно до вимог Python модуля.
- Отримання Результатів Прогнозування: Обробка результатів, отриманих від Python модуля, і передача їх до frontend частини. Це може включати: Форматування даних у зручний для користувача вигляд, відправка результатів назад на frontend для відображення користувачеві.

Ці відповідальності забезпечують ефективну обробку даних і взаємодію між різними компонентами вебзастосунку, що є критично важливим для його функціонування та задоволення потреб користувачів.

Модуль прогнозування (Python)

Модуль прогнозування виконує важливу роль у забезпеченні точності та ефективності прогнозів, що генеруються вебзастосунком. Основні відповідальності цього модуля включають:

1. Валідація та Обробка Даних

- Перевірка Даних: Валідація вхідних даних, щоб переконатися, що вони відповідають необхідним форматам і критеріям. Це може включати: Перевірку наявності всіх обов'язкових полів, перевірку на відповідність типу даних (числа, дати тощо).
- Підготовка Даних: Форматування та трансформація вхідних даних для забезпечення їх придатності до використання в алгоритмах прогнозування. Це може включати: Нормалізацію або стандартизацію даних, обробку відсутніх значень та аномалій.

2. Логіка Прогнозування

- Виконання Алгоритмів Прогнозування: Реалізація алгоритмів, які використовують історичні дані для створення прогнозів. Це включає: Використання статистичних методів і машинного навчання для аналізу даних, вибір оптимальних моделей на основі специфіки завдання та доступних даних.
- Аналіз Результатів: Оцінка точності прогнозів і внесення корективів у алгоритми за потреби. Це може включати: Порівняння прогнозів із фактичними результатами для визначення точності, налаштування параметрів моделі для покращення результатів.

3. Запис Даних до Баз Даних

- Запис Результатів Прогнозування: Збереження результатів прогнозування у базі даних для подальшого використання. Це включає: Форматування даних для відповідності схемі бази даних, визначення методу збереження (наприклад, вставка, оновлення).
- Запис Оброблених Історичних Даних: Зберігання даних, які були використані для прогнозування, щоб забезпечити доступ до них у майбутньому. Це може включати: Запис метаданих про проведене прогнозування, збереження історії даних для аналізу трендів і шаблонів.

- Отримання Результатів Прогнозування: Обробка результатів, отриманих від Python модуля, і передача їх до frontend частини. Це може включати: Форматування даних у зручний для користувача вигляд, відправка результатів назад на frontend для відображення користувачеві.
- Запис Оброблених Історичних Даних: Зберігання даних, які були використані для прогнозування, щоб забезпечити доступ до них у майбутньому. Це може включати: Запис метаданих про проведене прогнозування, збереження історії даних для аналізу трендів і шаблонів.

Ці відповідальності забезпечують ефективну роботу модуля прогнозування, дозволяючи здійснювати точні прогнози, базуючись на якісних даних та алгоритмах. Правильна реалізація цих функцій сприяє загальному успіху вебзастосунку та задоволенню потреб користувачів.

База Даних (MongoDB)

Зберігання історичних даних, результатів прогнозування, даних користувачів та інформації про підписки є ключовою відповідальністю бази даних. Така організація забезпечує чітке розділення між різними компонентами системи, що сприяє ефективному управлінню даними та забезпечує гнучкість і масштабованість архітектури.

Frontend орієнтується на взаємодію з користувачем, надаючи інтуїтивно зрозумілий інтерфейс для роботи з даними. Серверна частина на Express.js виконує роль посередника, обробляючи запити та передаючи дані між frontend і модулем прогнозування. Таким чином, усі компоненти системи працюють у тандемі, забезпечуючи узгоджену та продуктивну роботу веб-застосунку. Схема розробленої клієнт-серверної архітектури представлена на рисунку 3.1.

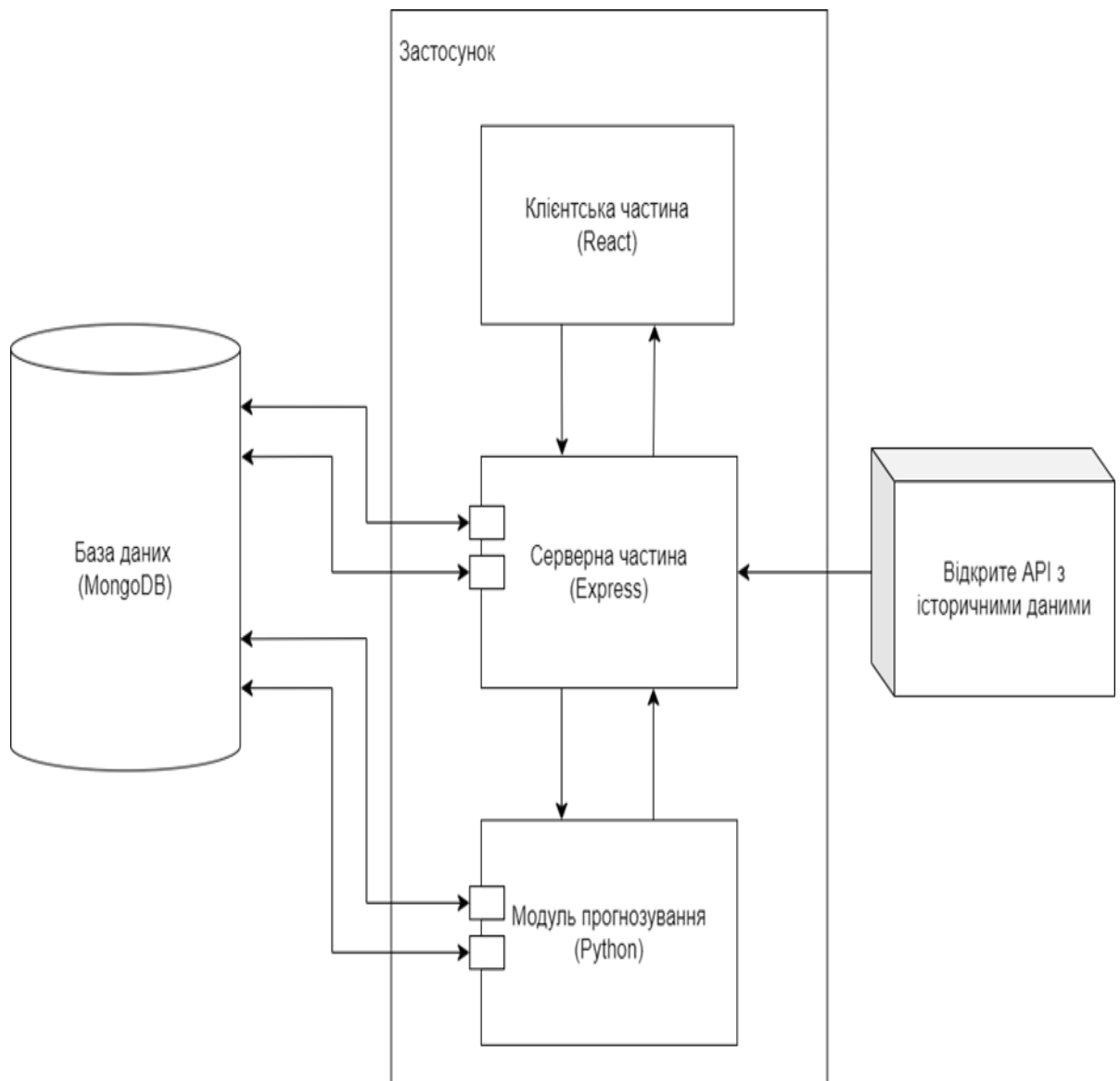


Рисунок 3.1 — Архітектура розробленого веб-застосунку

Використання шаблону “Model-View-Controller”

У вебдодатку для прогнозування індексу споживчих цін реалізована клієнт-серверна архітектура, основана на патерні "Model-View-Controller" (MVC). Ця архітектурна структура забезпечує гнучкість у проектуванні, спрощує процес оновлення та додавання нових функцій до системи, а також дозволяє ефективно повторно використовувати існуючі програмні компоненти [43].

На рис 3.2. представлено спосіб взаємодії користувача з системою, розробленою з використанням MVC-патерну.

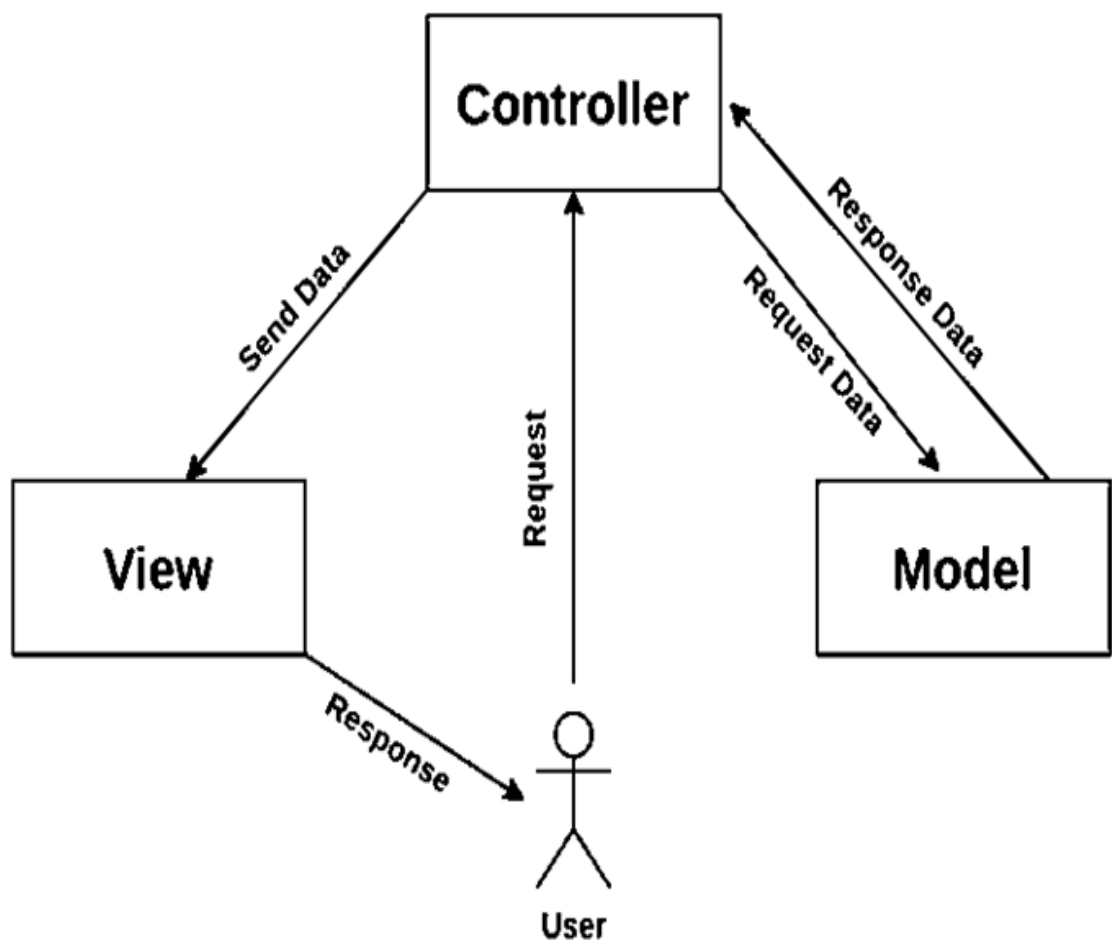


Рисунок 3.2 — Взаємодія користувача з MVC системою

Архітектурний шаблон MVC (Model-View-Controller) розділяє вебзастосунок на три ключові компоненти, кожен з яких виконує свої функції та відповідає за певну частину процесу обробки даних:

1. Модель (Model) – цей компонент відповідає за обробку даних, їх зберігання та управління. Він повністю незалежний від користувацького інтерфейсу і містить основну бізнес-логіку вебзастосунку. У нашій архітектурі модуль прогнозування та база даних виконують роль "Моделі", оскільки вони займаються обробкою даних, валідацією, логікою прогнозування та зберіганням результатів.

2. Вигляд (View) – цей компонент відповідає за представлення даних користувачам. Він відображає інформацію, отриману від контролера, у

зрозумілому та зручному для користувача форматі. У нашому випадку frontend, реалізований на базі React, виконує роль "Виду", забезпечуючи інтерфейс для взаємодії з користувачем.

3. Контролер (Controller) – цей компонент забезпечує зв'язок між моделлю та виглядом, управлінню взаємодією користувача з програмою. Контролер приймає вхідні дані від користувача, обробляє їх, використовуючи модель, і передає результати назад до вигляду для відображення. У нашій архітектурі backend на базі Express.js виконує роль "Контролера", управляючи логікою обробки запитів, маршрутизацією, автентифікацією та взаємодією між frontend і модулем прогнозування.

Використання шаблону MVC спрощує розуміння та управління структурою вебзастосунку. Це не лише полегшує розширення функціональності, але й робить підтримку системи більш ефективною. Кожен компонент може бути модифікований або оновлений незалежно, що забезпечує більшу гнучкість у розробці та підтримці проекту.

Опис бази даних

Для реалізації розроблюваного програмного забезпечення було створено 7 сутностей в базі даних MongoDB (рис.3.3):

- *USERS*;
- *SUBSCRIPTION_USER*;
- *CPI*;
- *EXTERNAL_VALUES*;
- *FORECAST*;
- *GRADIENTBOOSTING_CONFIGURATION*;
- *SARIMAX_CONFIGURATION*.

На рис. 3.3. можна побачити розроблену схему бази даних з відповідними таблицями.

USERS

USERS зберігає інформацію про користувачів системи, включаючи їх персональні дані та зв'язок з їхніми підписками. Назви всіх полів таблиці та їх відповідні типи даних наведено у таблиці 3.1.

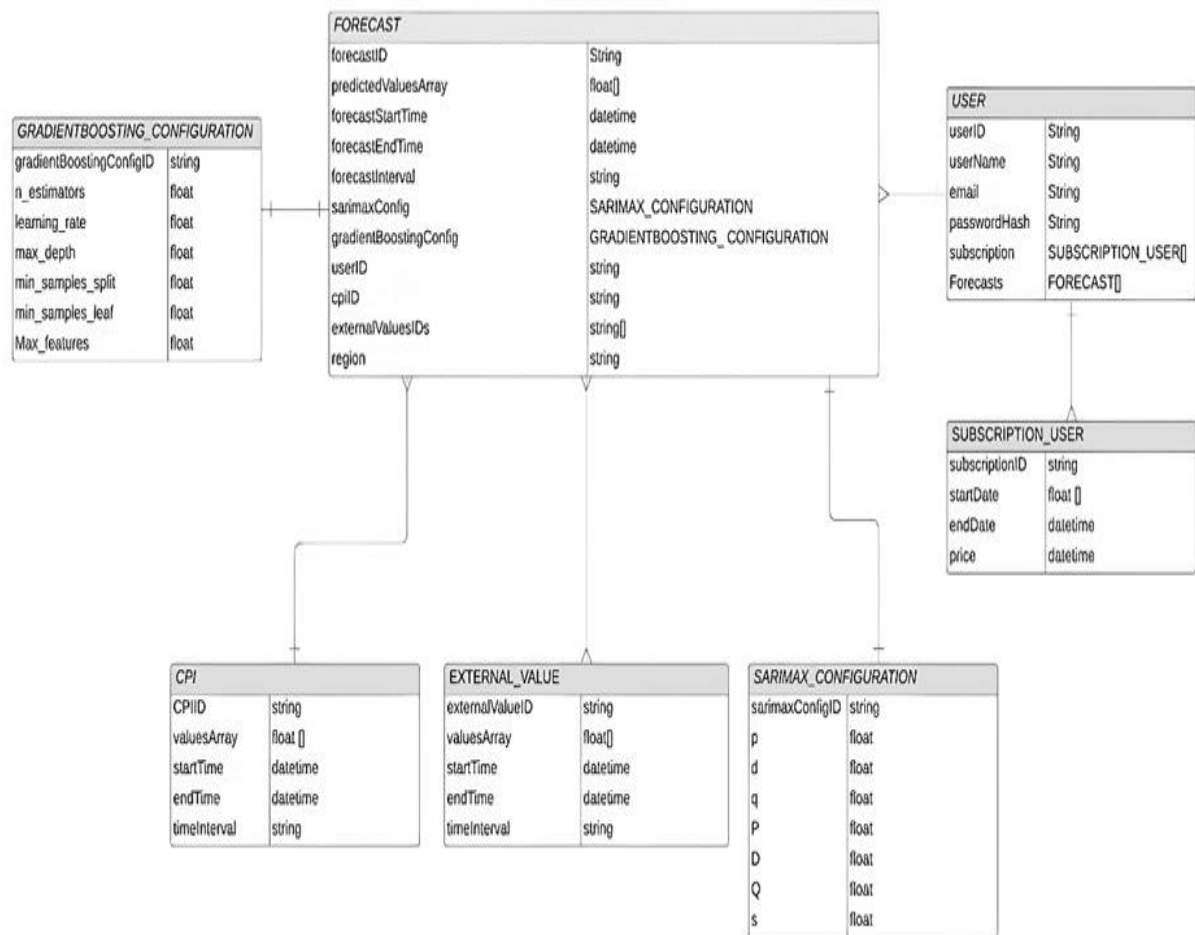


Рисунок 3.3 — Опис таблиць та зв’язків бази даних системи

Таблиця 3.1. Поля таблиці USERS

Назва поля	Тип даних
userID	String
userName	String
email	String
passwordHash	String
subscription	SUBSCRIPTION_USER[]
Forecasts	FORECAST[]

SUBSCRIPTION_USER

Роль. Керує підписками користувачів, забезпечуючи інформацію про тип та терміни підписки. Назви всіх полів таблиці та їх відповідні типи даних наведено у таблиці 3.2.

Таблиця 3.2. Поля таблиці SUBSCRIPTION_USER

<i>Назва поля</i>	<i>Тип даних</i>
subscriptionID	string
startDate	date
endDate	date
price	float

CPI

Роль. Зберігає історичні дані індексу споживчих цін, що є ключовими для аналізу та прогнозування. Назви всіх полів таблиці та їх відповідні типи даних наведено у таблиці 3.3

Таблиця 3.3. Поля таблиці CPI

<i>Назва поля</i>	<i>Тип даних</i>
CPIID	string
valuesArray	float []
startTime	datetime
endTime	datetime
timeInterval	string

EXTERNAL_VALUES

Роль. Записує екзогенні змінні, такі як економічні індикатори, які впливають на ІСЦ і використовуються в моделях прогнозування. Назви всіх полів таблиці та їх відповідні типи даних наведено у таблиці 3.4.

Таблиця 3.4. Поля таблиці EXTERNAL_VALUES

<i>Назва поля</i>	<i>Тип даних</i>
exogChangeID	string
valuesArray	float[]
startTime	datetime
endTime	datetime
timeInterval	string

FORECAST

Роль. Управляє прогнозами ІСЦ, зберігаючи результати прогнозів, їх часові метки та пов'язані конфігурації моделей. Назви всіх полів таблиці та їх відповідні типи даних наведено у таблиці 3.5.

Таблиця 3.5. Поля таблиці FORECAST

<i>Назва поля</i>	<i>Тип даних</i>
forecastID	String
predictedValuesArray	float[]
forecastStartTime	datetime
forecastEndTime	datetime
forecastInterval	string
sarimaxConfig	string
gradientBoostingConfig	string
userID	string
cpiID	string
externalValuesIDs	string[]
region	string

GRADIENTBOOSTING_CONFIGURATION

Роль: Ця таблиця забезпечує зберігання параметрів для моделей Gradient Boosting, використовуваних у прогнозах.

Ця схема бази даних організовує всі необхідні елементи для ефективної роботи системи прогнозування індексу споживчих цін (ІСЦ). Вона забезпечує взаємозв'язок між користувачами, їхніми підписками, історичними даними, прогнозами та налаштуваннями моделей, що дозволяє системі функціонувати безперешкодно і з максимальною ефективністю.

Назви всіх полів таблиці та їх відповідні типи даних наведено у таблиці 3.6.

Таблиця 3.6. Поля таблиці *GRADIENTBOOSTING_CONFIGURATION*

<i>Назва поля</i>	<i>Тип даних</i>
gradientBoostingConfigID	String
n_estimators	float
learning_rate	float
max_depth	float
min_samples_split	float
min_samples_leaf	float
max_features	float

SARIMAX_CONFIGURATION

Роль. Зберігає налаштування для моделей SARIMAX, використовуваних у прогнозах. Назви всіх полів таблиці та їх відповідні типи даних наведено у таблиці 3.7.

Таблиця 3.7. Поля таблиці *GRADIENTBOOSTING_CONFIGURATION*

<i>Назва поля</i>	<i>Тип даних</i>
sarimaxConfigID	string
p	float
d	float
q	float
P	float
D	float
Q	float
s	float

3.3 Розробка інтерфейсу користувача

Користувач може взаємодіяти з веб-застосунком через інтерфейс користувача, представлений у формі кількох веб-сторінок. Для авторизованого користувача в системі доступні наступні сторінки:

- Головна сторінка: Ця сторінка надає загальний опис проекту, його цілей та можливостей для користувача. Вона також містить інформацію про платну підписку, що допомагає користувачам зрозуміти переваги, які вони можуть отримати.
- Сторінка авторизації: Ця сторінка забезпечує форму для введення електронної пошти та паролю, що дозволяє користувачу здійснити авторизований вхід у систему.
- Сторінка реєстрації: Сторінка для введення необхідних даних, які потрібні для реєстрації нового користувача в системі, включаючи ім'я, електронну пошту, пароль тощо.
- Сторінка користувача: Ця сторінка містить основну інформацію про користувача, таку як ім'я та електронна пошта, а також відомості про

підписку: статус поточної підписки, дату закінчення, ціну, а також можливість оновлення або зміни підписки.

- Сторінка створення прогнозу: На цій сторінці користувач може завантажити необхідні дані для створення нового прогнозу. Тут представлено параметри прогнозу, такі як вибір методу моделі (ARIMAX/SARIMAX, Gradient Boosting), часовий період для прогнозу та інші налаштування. Користувач також має можливість використовувати попередні дані для створення прогнозу. Для підтвердження даних і відправлення запиту на створення прогнозу потрібно натиснути відповідну кнопку.
- Сторінка історії прогнозів: Ця сторінка відображає список усіх прогнозів, зроблених користувачем. Користувач може переглядати детальну інформацію про кожний прогноз, а також переходити на сторінку конкретного прогнозу з усіма деталями, включаючи історичні дані, параметри моделі та отримані прогнози.
- Сторінка прогнозу: Ця сторінка демонструє параметри моделі, що використовувалися для конкретного прогнозу, а також графіки та таблиці з прогнозованими значеннями. Користувач має можливість переглядати історичні дані, які були використані для створення цього прогнозу.

Ця структура вебзастосунку надає користувачам зручний та інтуїтивно зрозумілий інтерфейс для взаємодії з системою прогнозування індексу споживчих цін, забезпечуючи доступ до всіх необхідних функцій та даних.

3.4 Розробка алгоритм модуля прогнозування індексу споживчих цін в умовах не стабільної економіки.

Модуль прогнозування в розроблюваному вебзастосунку виконує основну функцію – прогнозування ІСЦ на основі історичних даних та екзогенних змінних. Однак він також виконує додаткові завдання, такі як

обробка та збереження інформації про історичні значення ІСЦ та екзогенних змінних у базі даних. При першому вході в систему користувач має два варіанти завантаження історичних даних для прогнозування: ручне завантаження з локального сховища або автоматизоване завантаження через API-інтеграцію з іншими сервісами.

Після завантаження дані проходять процес валідації. Якщо дані успішно валідовані, вони переходять до наступного етапу, де обробляються для приведення до формату, необхідного для прогнозування. Після обробки дані використовуються для прогнозування за модифікованим методом. Після завершення прогнозування всі дані, що стосуються прогнозу, записуються до бази даних. Ці дані можуть бути використані для повторних прогнозів без необхідності знову завантажувати та обробляти історичні дані, що є особливо корисним для різних параметрів моделей.

Неавторизовані користувачі мають доступ лише до головної сторінки, а також до сторінок реєстрації та авторизації. Зареєстровані, але не оплачувані користувачі мають доступ до всіх сторінок системи, проте функція проведення нових прогнозів недоступна. Повторне використання вже підготовлених даних для нових прогнозів зменшує навантаження на систему та час, витрачений на обробку даних.

Централізоване зберігання даних спрощує користувачеві процес взаємодії з системою та підвищує масштабованість і безпеку. Накопичення історичних даних дозволяє з часом оптимізувати процес прогнозування на основі аналітики, що може призвести до підвищення точності прогнозів. Таким чином, модуль прогнозування не лише забезпечує основну функцію прогнозування, але й створює зручні умови для роботи з даними, підвищуючи загальну ефективність системи.

Алгоритм роботи модулю прогнозування можна переглянути на рисунку 3.11.

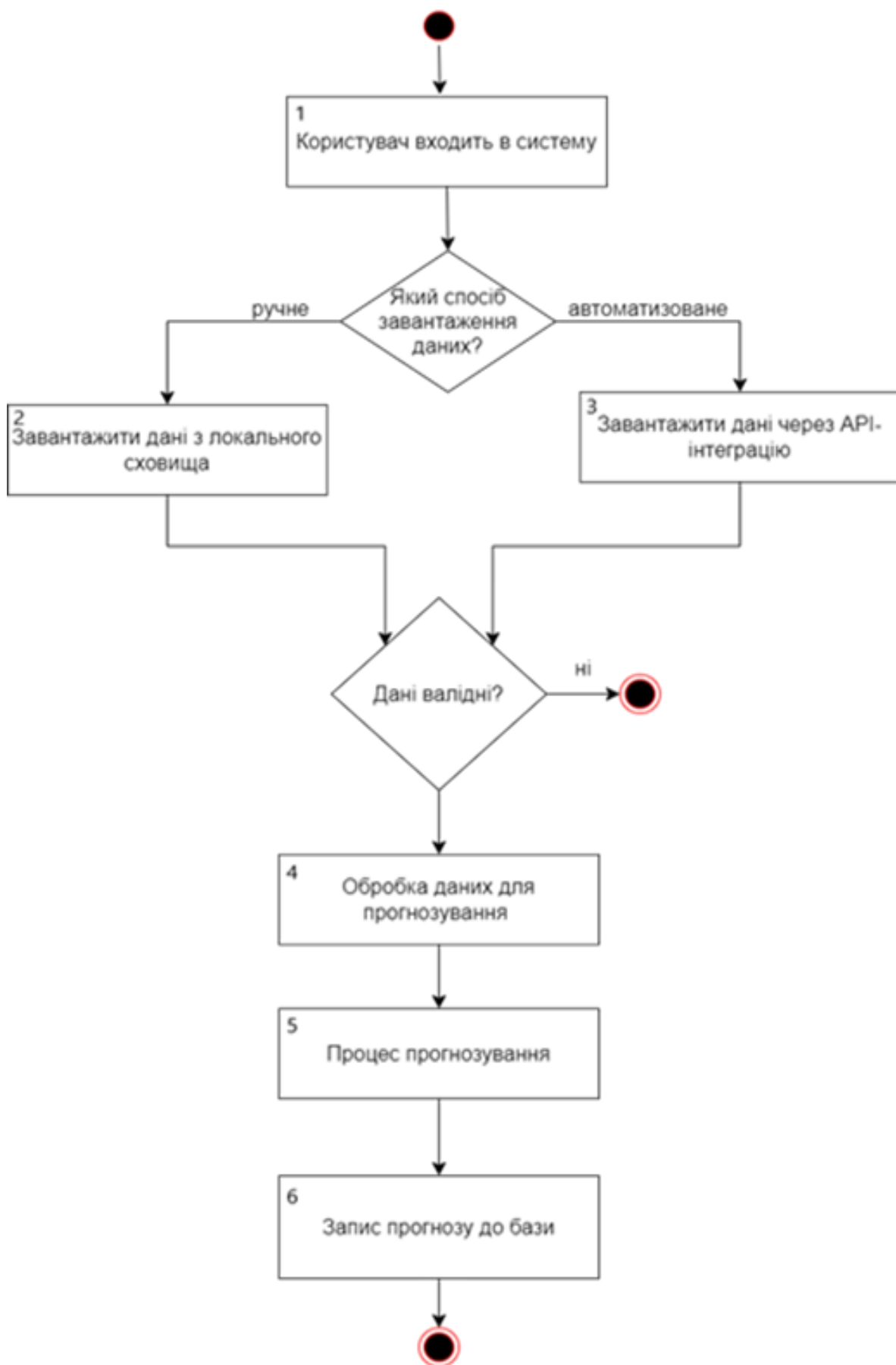


Рисунок 3.11 — Алгоритм роботи модулю прогнозування

3.5 Висновки до розділу 3

У цьому розділі детально проаналізовано оптимальний вибір технологій та інструментів для розробки веб-застосунку з модулем прогнозування індексу споживчих цін. Для більшої частини веб-застосунку обґрунтовано вибір JavaScript як основної мови програмування. Це зумовлено її універсальністю, оскільки вона підходить як для розробки клієнтської, так і серверної частини. JavaScript має величезну спільноту розробників та різноманітні фреймворки й бібліотеки, що спрощує та пришвидшує розробку. Серед JavaScript-фреймворків для клієнтської частини обрано React завдяки його гнучкості, модульності та можливостям багаторазового використання компонентів.

Для серверної частини рекомендовано Express – найпопулярніший Node.js фреймворк, що відрізняється простотою налаштування, швидкістю розробки та інтеграцією з екосистемою Node.js. Як систему управління базами даних обрано MongoDB, з огляду на її гнучкі схеми даних, легку масштабованість та ефективність обробки великих обсягів даних. Це дозволить швидко адаптувати структуру даних під потреби проекту.

Для статистичної обробки даних та побудови моделей прогнозування запропоновано використовувати мову програмування Python, яка має широкий набір спеціалізованих бібліотек для аналізу даних, машинного навчання та візуалізації. Запропоновано клієнт-серверну архітектуру, побудовану за шаблоном "Model-View-Controller", що забезпечує чіткий розподіл функцій між компонентами: фронтенд відповідає за взаємодію з користувачем, серверна частина виступає посередником, а окремий модуль прогнозування на Python виконує всі необхідні обчислення.

Детально описано базу даних, що складається з семи основних сутностей для зберігання інформації про користувачів, підписки, історичні дані, прогнози та конфігурації моделей машинного навчання. Запропонована модель бази даних оптимально організовує дані для

ефективної роботи системи прогнозування. Окремо розглянуто модуль прогнозування на Python, що виконує валідацію даних, підготовку даних, безпосередньо прогнозування за обраним методом та запис результатів до бази даних. Наведено алгоритм роботи цього модуля.

Отже, зазначений технологічний стек забезпечує масштабовану архітектуру веб-застосунку, при цьому враховано баланс між потужністю обраних технологій, продуктивністю розробки та наявністю експертної підтримки відповідних спільнот розробників. Вибрані технології та структурні рішення оптимально задовольняють вимоги до системи прогнозування індексу споживчих цін.

Також у розділі детально описано інтерфейс користувача у вигляді декількох основних веб-сторінок, через які відбувається взаємодія з системою прогнозування ІСЦ. Для авторизованих користувачів доступні такі сторінки: головна, авторизації, реєстрації, особистого кабінету користувача, створення та перегляду прогнозів. Окремо виділено три рівні доступу до застосунку: для неавторизованих користувачів, зареєстрованих без підписки та повністю авторизованих користувачів з активною підпискою. Останні мають повний доступ до всіх можливостей веб-застосунку.

4 ТЕСТУВАННЯ ТА АНАЛІЗ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Тестування результатів роботи програми індексу споживчих цін в умовах не стабільної економіки

У процесі розробки модифікованого програмного способу прогнозування індексу споживчих цін (ІСЦ) ключовим аспектом є встановлення об'єктивних критеріїв для оцінки його ефективності. Основним критерієм, що враховується, є точність прогнозування. Для кількісного аналізу точності використовується середня абсолютна помилка (Mean Absolute Percentage Error, MAPE), яка дозволяє оцінити відхилення прогнозованих даних від фактичних. Це забезпечує зрозуміле та точне вимірювання ефективності прогнозу, даючи можливість оцінити, наскільки модель наближається до реальних значень ІСЦ. Середня абсолютна помилка визначається за формулою:

$$q = \frac{|x_r - x_f|}{x_r},$$

де x_r – реальне значення курсу валюти, x_f – прогнозоване значення курсу валюти, q – середня абсолютна похибка.

На заключному етапі проводиться порівняльний аналіз модифікованого способу прогнозування з традиційними методами. Оцінюються не лише точність прогнозів, але й інші важливі характеристики, такі як гнучкість використання та можливість масштабування системи. Такий підхід забезпечує всебічну оцінку переваг модифікованого способу прогнозування, дозволяє визначити його ефективність у порівнянні з існуючими рішеннями та виявити потенційні напрямки для подальшого вдосконалення алгоритмів і технологій.

Результати порівняння ефективності роботи модифікованого та класичного способів наведено у таблиці 4.1.

Таблиця 4.1. Порівняння ефективності модифікованого та класичного способів прогнозу

Місяць прогнозу	Модифікований спосіб прогнозу		Класичний спосіб прогнозу		Реальне значення
	Похибка прогнозу	Значення	Похибка прогнозу	Значення	
01.06.2023	1.10	100,2	0.14	101.16	101.30
01.07.2023	0.35	101,25	0.47	101.13	101.60
01.08.2023	1.60	102,9	3.12	101.38	104.50
01.09.2023	0.60	102,5	2.03	101.07	103.10
01.10.2023	0.30	102,4	1.46	101.24	102.70
01.11.2023	0.20	101,7	1.37	100.13	101.50
01.12.2023	0.20	100,4	0.47	99.73	100.20
01.01.2024	0.30	100,2	0.08	100.58	100.50
01.02.2024	0.10	100,9	0.76	101.56	100.80
01.03.2024	0.20	99,6	2.54	101.94	99.40

Порівняння ефективності модифікованого та класичного способів прогнозування індексу споживчих цін (ІСЦ) в Україні за період з червня 2023 до вересня 2024 року свідчить про перевагу модифікованого методу над класичним. Для цього аналізу використовувалися історичні дані ІСЦ та екзогенні змінні, такі як середній рівень заробітної плати, курс долара і рівень безробіття.

Модифікований метод демонструє нижчу похибку прогнозу протягом усього аналізованого періоду. Зокрема, середня похибка прогнозування після використання моделі Sarimax склала 1.324 пункти, тоді як модифікований спосіб забезпечив середню похибку на рівні 0.475 пунктів. Така значна різниця свідчить про вищу точність модифікованого методу, що дозволяє йому краще пристосовуватися до змін економічної ситуації.

Загалом, результати порівняльного аналізу демонструють, що модифікований метод прогнозування забезпечує значно точніші та надійніші прогнози ІСЦ порівняно з класичним підходом, що робить його більш адаптованим до економічної невизначеності та швидких змін.

Наглядно порівняти точності модифікованого та класичного способів можна на рисунку 4.2.

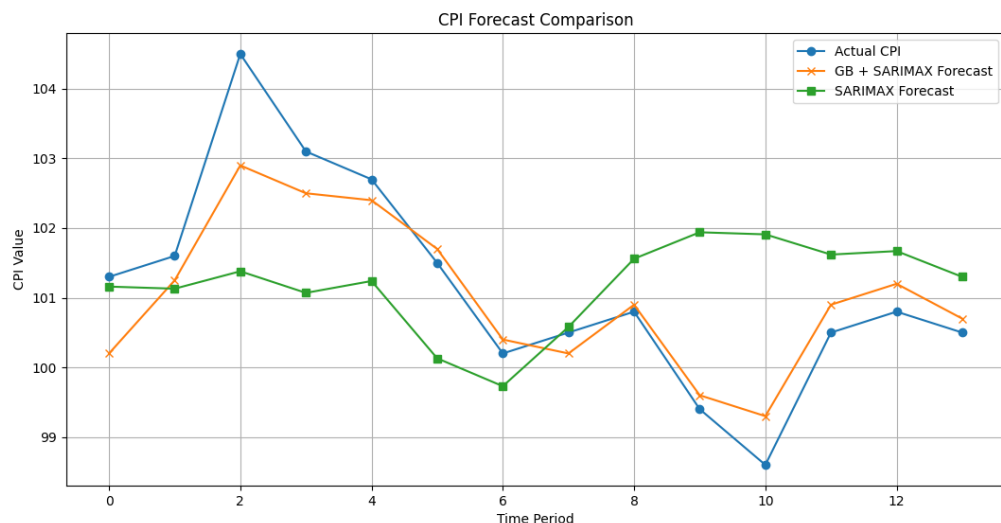


Рисунок. 4.2. Прогнозування модифікованого та класичного способів

4.2 Аналіз розробленого програмного забезпечення індексу споживчих цін в умовах не стабільної економіки

Процес тестування та аналізу розробленого вебдодатку для прогнозування індексу споживчих цін (ІСЦ) охопив кілька критичних аспектів, що мають значення для забезпечення його ефективності та надійності.

На етапі тестування фронтенду, реалізованого на React, головна увага була зосереджена на функціональності різних вебсторінок. Тестування охоплювало перевірку інтуїтивності навігації, коректності роботи форм реєстрації та авторизації, а також зручності інтерфейсу на сторінці створення прогнозів. Особлива увага приділялася тестуванню відображення результатів прогнозування та історії прогнозів, з метою перевірки точності й доступності інформації для користувачів.

Для бекенду на Express.js основний акцент було зроблено на тестуванні API та маршрутизації, перевірці системи аутентифікації користувачів, а також на надійності обробки запитів і передачі даних до модуля прогнозування.

Модуль прогнозування на Python, як ключовий елемент системи, також пройшов ретельне тестування. Перевірялися алгоритми прогнозування ІСЦ, процеси валідації й обробки історичних даних. Особлива увага приділялася механізмам завантаження даних, як у ручному режимі, так і через автоматизовану API-інтеграцію. Тестування цього модуля зосереджувалося на перевірці точності прогнозів і коректності їхнього запису до бази даних.

База даних, яка зберігає історичні дані ІСЦ, результати прогнозів і інформацію про користувачів, була протестована на стабільність, безпеку та швидкість виконання операцій. Велике значення надавали ефективності

обробки великих обсягів інформації та надійності зберігання даних, оскільки ці фактори безпосередньо впливають на якість прогнозування.

Важливою частиною тестування стало застосування unit-тестів, що дали змогу перевірити кожен компонент системи окремо. Це дозволило виявити й усунути конкретні помилки та неефективності на ранніх етапах розробки. Unit-тести використовувалися для тестування логіки фронтенду, обробки запитів на бекенді, правильності виконання алгоритмів прогнозування, а також для перевірки надійності зберігання та обробки даних у базі даних.

Під час тестування програмне забезпечення показало високу надійність, ефективність і коректність роботи, що є важливим для забезпечення якості послуг, які надаються користувачам. Аналіз результатів тестування виявив корисні висновки для подальшого вдосконалення та розвитку системи.

4.3 Висновки до розділу 4

Для об'єктивної оцінки ефективності модифікованого способу прогнозування індексу споживчих цін (ІСЦ) було обрано критерій точності прогнозу, що вимірювався за допомогою середньої абсолютної похибки. Порівняння з класичним методом прогнозування у період з червня 2024 року по вересень 2024 року продемонструвало значну перевагу модифікованого підходу, який показав нижчу середню похибку на рівні 0.475 проти 1.324 у класичного методу.

Результати аналізу підтвердили вищу точність, адаптивність та надійність модифікованого способу прогнозування ІСЦ, що робить його більш ефективним інструментом для економічного моделювання. Процес тестування був комплексним і охоплював перевірку функціональності та інтерфейсу фронтенд частини, роботи бекенду та API, алгоритмів та

механізмів прогнозування, а також взаємодії з базою даних. Особлива увага приділялася точності та надійності прогнозів індексу споживчих цін.

Застосування unit-тестів дало змогу ретельно протестувати окремі компоненти системи та виявити потенційні дефекти на ранніх стадіях розробки. Це допомогло забезпечити високу якість та надійність програмного продукту. Комплексне тестування та аналіз його результатів підтвердили ефективність розробленого рішення, а також виявили напрямки для подальшого вдосконалення системи прогнозування ІСЦ.

5 ЕКОНОМІЧНА ЧАСТИНА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПРОГНОЗУВАННЯ ІНДЕКСУ СПОЖИВЧИХ ЦІН В УМОВАХ НЕСТАБІЛЬНОЇ ЕКОНОМІКИ

Науково-технічна розробка може бути реалізована і впроваджена, якщо вона відповідає сучасним вимогам як у сфері науково-технічного прогресу, так і з точки зору економічної доцільності. Тому для науково-дослідної діяльності важливо оцінювати економічну ефективність отриманих результатів.

Магістерська кваліфікаційна робота «Інформаційна технологія прогнозування індексу споживчих цін в умовах нестабільної економік» належить до науково-технічних розробок, які мають потенціал для виведення на ринок, тобто спрямовані на комерціалізацію. Такий підхід є місцем, оскільки розробкою можуть зменшити інші споживачі, отримуючи економічні вигоди. Для цього необхідно залучити інвестора та переконатися в його економічній доцільності проекту.

Для наведеного випадку нами мають бути виконані такі етапи робіт:

- 1) Оцінювання комерційного потенціалу розробки;
- 2) Прогнозування витрат на виконання науково-дослідної роботи;
- 3) Розрахунок економічної ефективності науково-технічної розробки.

5.1 Оцінювання комерційного потенціалу розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Інформаційна технологія прогнозування індексу споживчих цін в умовах нестабільної економік» є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

В умовах сучасної України, що включають військовий конфлікт та його економічні наслідки, точне прогнозування індексу споживчих цін ІСЦ

стає надзвичайно важливим інструментом для моніторингу інфляційних процесів та прийняття ефективних економічних рішень. ІСЦ відіграє ключову роль у формуванні економічної політики та впливає на інвестиційну активність. Під час війни економіка зазнає значних змін, що робить прогнозування ІСЦ ще складнішим та критичним завданням.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в табл. 5.1.

Для наведеного випадку нами мають бути виконані такі етапи робіт:

- 1) Оцінювання комерційного потенціалу розробки;
- 2) Прогнозування витрат на виконання науково-дослідної роботи;
- 3) Розрахунок економічної ефективності науково-технічної розробки.

Таблиця 5.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів

Продовження таблиці 5.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років

Продовження таблиці 5.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри- терій	0	1	2	3	4
				3-х до 5-ти років	
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки пові- домлення відповідни м органам про виробницт во та реалізацію продукту	Відсутні будь-які регламент ні обмеженн я на виробниц тво та реалізаці ю продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до наступної таблиці.

Таблиця 5.2 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0-10	Низький
11-20	Нижче середнього
21-30	Середній
31-40	Вище середнього
41-48	Високий

Таблиця 5.3 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	Експерт 1	Експерт 2	Експерт 3
	Бали, виставлені експертами:		
1	1	2	4
2	1	1	3
3	5	4	4
4	4	2	2
5	4	2	2
6	2	3	3
7	2	3	4
8	2	4	4

Критерії	Прізвище, ініціали, посада експерта		
	Експерт 1	Експерт 2	Експерт 3
	Бали, виставлені експертами:		
9	2	2	2
10	2	2	3
11	4	4	4
12	2	4	2
Сума балів	СБ ₁ =31	СБ ₂ =33	СБ ₃ =37
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_1^3 СБ_i}{3} = \frac{31 + 33 + 37}{3} = 34$		

Середня оцінка, отримана на основі експертних висновків, становить 34 бали, що відповідно до таблиці 5.2 про комерційний потенціал проведених досліджень на рівні вище середнього. Інформаційна веб-система для аналізу динаміки географічної структури зовнішньої торгівлі України, яка є програмою для моніторингу, аналізу та порівняння даних, буде корисною для різних організацій та компаній, які орієнтуються.

5.2 Прогнозування витрат на виконання науково-дослідної роботи

Витрати на проведення науково-дослідної роботи виділяються на такі категорії: оплата праці, соціальні витрати, витрати на матеріали, паливо й енергію для науково-виробничих потреб, службові відрядження, програмне забезпечення наукових цілей, інші витрати та накладні витрати.

Основна заробітна плата кожного із дослідників Z_o , якщо вони працюють в наукових установах бюджетної сфери визначається за формулою:

$$Z_o = \frac{M}{T_p} * t \text{ (грн)}, \quad (5.1)$$

де M – оклад розробника (інженера, дослідника, науковця тощо), грн.;

T_p – число робочих днів в місяці; приблизно $T_p \approx 21...23$ дні;

t – число робочих днів роботи дослідника.

Для розробки програмних засобів необхідно залучити програміста з посадовим окладом 25000 грн. Кількість робочих днів у місяці складає 30, а кількість робочих днів програміста складає 22. Зведемо сумарні розрахунки до таблиця 5.4.

Таблиця 5.4 – Заробітна плата дослідника в науковій установі бюджетної сфери

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник	30000	1363	5	6815
Програмний інженер	25000	1136	40	45440
Всього				52255

Розрахунок додаткової заробітної плати робітників

Додаткова заробітна плата Z_d всіх розробників та робітників, які приймали участь в розробці нового технічного рішення розраховується як 10 - 12 % від основної заробітної плати робітників.

На даному підприємстві додаткова заробітна плата начисляється в розмірі 10% від основної заробітної плати.

$$Z_d = (Z_o + Z_p) * \quad (5.2)$$

$$Z_d = 0,1 * 52255 = 5226(\text{грн}).$$

. Нарахування на заробітну плату $H_{ЗП}$ дослідників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою:

$$H_{ЗП} = (Z_o + Z_d) * \frac{\beta}{100}, \quad (5.3)$$

де Z_o – основна заробітна плата розробників, грн.;

Z_d – додаткова заробітна плата всіх розробників та робітників, грн.;

β – ставка єдиного внеску на загальнообов’язкове державне соціальне страхування, % .

Дана діяльність відноситься до бюджетної сфери, тому ставка єдиного внеску на загальнообов’язкове державне соціальне страхування буде складати 22%, тоді:

$$H_{ЗП} = (52255 + 5226) * \frac{22}{100} = 12646 (\text{грн}).$$

Витрати на комплектуючі вироби, які використовують при виготовленні одиниці продукції, розраховуються, згідно їх номенклатури, за формулою:

$$K = \sum_{i=1}^n H_i * C_i * K_i \quad (5.4)$$

де H_i – кількість комплектуючих i -го виду, шт.;

C_i – покупна ціна комплектуючих i -го найменування, грн.;

K_i – коефіцієнт транспортних витрат (1,1...1,15).

Таблиця 4.5 – Комплектуючі, що використані на розробку

Найменування матеріалу	Ціна за одиницю, грн.	Витрачено	Вартість витраченого матеріалу, грн.
Папір	100	1	100
Ручка	30	1	30
CD-диск	50	1	50
Флешка	250	1	250
Всього			430
З врахуванням коефіцієнта транспортування			500

Програмне забезпечення для наукової роботи включає витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення необхідного для проведення дослідження.

Для написання магістерської роботи використовувалися редактор VSCode та онлайн база даних MongoDB.

Амортизація обладнання, комп'ютерів та приміщень, які використовувались під час виконання даного етапу роботи

Дані відрахування розраховують по кожному виду обладнання, приміщенням тощо.

$$A = \frac{C * T}{T_{\text{кор}} * 12}, \quad (5.5)$$

де C – балансова вартість даного виду обладнання (приміщень), грн.;

$T_{\text{кор}}$ – час користування;

T – термін використання обладнання (приміщень), цілі місяці.

Згідно пункта 137.3.3 Податкового кодекса амортизація нараховується на основні засоби вартістю понад 2500 грн. В нашому випадку для написання магістерської роботи використовувався персональний комп'ютер вартістю 40000 грн.

$$A = \frac{40000 \cdot 1}{2 \cdot 12} = 1667 \text{ (грн)}.$$

До статті «Паливо та енергія для науково-виробничих цілей» відносяться витрати на всі види палива й енергії, що безпосередньо використовуються з технологічною метою на проведення досліджень.

$$B_e = \sum_{i=1}^n \frac{W_{yt} \cdot t_i \cdot C_e \cdot K_{впi}}{\eta_i}, \quad (5.6)$$

де W_{yt} – встановлена потужність обладнання на певному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

C_e – вартість 1 кВт-години електроенергії, грн;

$K_{впi}$ – коефіцієнт, що враховує використання потужності, $K_{впi} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

Для написання магістерської роботи використовується персональний комп'ютер для якого розрахуємо витрати на електроенергію.

$$B_e = \frac{0,5 \cdot 300 \cdot 1,68 \cdot 0,5}{0,8} = 126 \text{ (грн)}.$$

Витрати на службові відрядження, витрати на роботи, які виконують сторонні підприємства, установи, організації та інші витрати в нашому дослідженні не враховуються оскільки їх не було.

Накладні (загальновиробничі) витрати $B_{нзв}$ охоплюють: витрати на управління організацією, оплата службових відряджень, витрати на

утримання, ремонт та експлуатацію основних засобів, витрати на опалення, освітлення, водопостачання, охорону праці тощо. Накладні (загальновиробничі) витрати $V_{нзв}$ можна прийняти як $(100...150)\%$ від суми основної заробітної плати розробників та робітників, які виконували дану МКНР, тобто:

$$V_{нзв} = (Z_o + Z_p) \cdot \frac{H_{нзв}}{100\%}, \quad (5.7)$$

де $H_{нзв}$ – норма нарахування за статтею «Інші витрати».

$$V_{нзв} = 52255 \cdot \frac{100}{100\%} = 52255(\text{грн}) \quad (5.8)$$

Сума всіх попередніх статей витрат дає витрати, які безпосередньо стосуються даного розділу МКНР

$$B = 52255 + 5226 + 12646 + 500 + 1667 + 126 + 52255 = 124675 (\text{грн}).$$

Прогнозування загальних втрат $ЗВ$ на виконання та впровадження результатів виконаної МКНР здійснюється за формулою:

$$ЗВ = \frac{B}{\eta}, \quad (5.9)$$

де η – коефіцієнт, який характеризує стадію виконання даної НДР.

Оскільки, робота знаходиться на стадії науково-дослідних робіт, то коефіцієнт $\beta = 0,9$.

Звідси:

$$ЗВ = \frac{124675}{0,9} = 138528(\text{грн}).$$

5.3 Розрахунок економічної ефективності науково-технічної розробки

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження передбачають комерціалізацію протягом 4-х років реалізації на ринку.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

ΔN – збільшення кількості споживачів продукту, у періоди часу, що аналізуються, від покращення його певних характеристик;

Показник	1-й рік	2-й рік	3-й рік	4-й рік
Збільшення кількості споживачів, осіб	250	550	770	300

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки, прийmemo 5000 осіб;

C_o – вартість програмного продукту у році до впровадження результатів розробки, прийmemo 9400,00 грн;

$\pm \Delta C_o$ – зміна вартості програмного продукту від впровадження результатів науково-технічної розробки, прийmemo 658 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta \Pi_i$ для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [42]:

$$\Delta\Pi_i = (\pm\Delta\Pi_o \cdot N + \Pi_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \quad (5.10)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2024 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту). Прийmemo $\rho = 30\%$;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2024 році $\vartheta = 18\%$;

Збільшення чистого прибутку 1-го року:

$$\Delta\Pi_1 = (658 \cdot 5000,00 + 9958,70 \cdot 250) \cdot 0,83 \cdot 0,3 \cdot (1 - 0,18/100\%) = 1180093,99 \text{ грн.}$$

Збільшення чистого прибутку 2-го року:

$$\Delta\Pi_2 = (658 \cdot 5000,00 + 9958,70 \cdot 800) \cdot 0,83 \cdot 0,3 \cdot (1 - 0,18/100\%) = 2298446,09 \text{ грн.}$$

Збільшення чистого прибутку 3-го року:

$$\Delta\Pi_3 = (658 \cdot 5000,00 + 9958,70 \cdot 1570) \cdot 0,83 \cdot 0,3 \cdot (1 - 0,18/100\%) = 3911262,61 \text{ грн.}$$

Збільшення чистого прибутку 4-го року:

$$\Delta\Pi_4 = (658 \cdot 5000,00 + 9958,70 \cdot 1870) \cdot 0,83 \cdot 0,3 \cdot (1 - 0,18/100\%) = 4474149,17 \text{ грн.}$$

5.4 Розрахункова ефективність вкладених інвестицій та період їх окупності

Розрахуємо основні показники, які визначають доцільність фінансування наукової розробки певним інвестором, є абсолютна і відносна ефективність вкладених інвестицій та термін їх окупності.

Розрахуємо величину початкових інвестицій PV, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки.

$$PV = k_{\text{інв}} \cdot 3B, \quad (5.11)$$

де $k_{\text{інв}}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо ($k_{\text{інв}} = 2 \dots 5$).

$$PV = 2 \cdot 138528 = 277056(\text{грн}).$$

Розрахуємо абсолютну ефективність вкладених інвестицій $E_{\text{абс}}$ згідно наступної формули:

$$E_{\text{абс}} = (\text{ПП} - PV), \quad (5.12)$$

де ПП – приведена вартість всіх чистих прибутків, що їх отримає підприємство від реалізації результатів наукової розробки, грн.;

$$\text{ПП} = \sum_1^T \frac{\Delta\Pi_i}{(1+\tau)^t}, \quad (5.13)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДЦКР, грн.;

T – період часу, протягом якою виявляються результати впровадженої НДЦКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,2;

t – період часу (в роках).

$$\begin{aligned} \text{ПП} &= \frac{1180093,99}{(1 + 0,2)^1} + \frac{2298446,09}{(1 + 0,2)^2} + \frac{3911262,61}{(1 + 0,2)^3} + \frac{4474149,17}{(1 + 0,2)^4} \\ &= 700689,19 \text{ (грн)}. \end{aligned}$$

$$E_{abc} = (700689,19 - 277056) = 6723433,18 \text{ (грн)}.$$

Оскільки $E_{abc} > 0$, то вкладання коштів на виконання та впровадження результатів НДДКР може бути доцільним.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_B . Для цього користуються формулою:

$$E_B = \sqrt[T_{\text{ж}}]{1 + \frac{E_{abc}}{PV}} - 1, \quad (5.14)$$

де $T_{\text{ж}}$ – життєвий цикл наукової розробки, роки.

$$E_B = \sqrt[3]{1 + \frac{6723433,18}{277056}} - 1 = 1,96.$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \quad (5.15)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2024 році в Україні $d = (0,14 \dots 0,2)$;

f – показник, що характеризує ризикованість вкладень; зазвичай, величина $f = (0,05 \dots 0,1)$.

$$\tau_{min} = 0,19 + 0,7 = 0,89.$$

Так як $E_e > \tau_{min}$ то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_B}. \quad (5.16)$$

$$T_{ок} = \frac{1}{0,89} = 1,12 \text{ (роки)}.$$

Так як $T_{ок} \leq 3...5$ -ти років, то фінансування даної наукової розробки в принципі є доцільним.

5.5 Висновок економічної частини науково-технічної розробки

Було проведено оцінку комерційного потенціалу інформаційної веб-системи для аналізу динаміки географічної структури зовнішньої торгівлі товарами України, яка підсумовує аналіз даних, моніторинг та створення порівняльних графіків статистики.

Прогнозовані витрати на виконання науково-дослідної роботи за шкідливою статтею витрат становлять 124675 грн. Загальна сума витрат на реалізацію та впровадження результатів НДР складе 138528 грн

Інвестиції в цей проект окупляться за 1,12 роки. Чиста приведена вартість усіх прибутків, які створені підприємством від реалізації результатів наукових розробок, складе 700689,19 грн.

ВИСНОВКИ

Індекс споживчих цін є важливим показником, що відображає загальні тенденції цінової динаміки в країні. Для України характерна сезонність ІСЦ через коливання цін на продукти харчування та енергоносії. Традиційно для прогнозування ІСЦ використовуються статистичні методи. Водночас методи машинного навчання можуть підвищити точність завдяки здатності адаптуватися до динамічних даних. Існуючі статистичні пакети, такі як Stata, EViews, SAS, SPSS та Bloomberg Terminal, мають потужний аналітичний інструментарій, але є досить дорогими, складними у використанні та потребують багато неавтоматизованих операцій. Крім того, вони обмежені в застосуванні сучасних методів машинного навчання, які дозволяють покращити точність прогнозування. Тому розробка спеціалізованого вебзастосунку дозволяє поєднати традиційні та сучасні підходи для підвищення точності прогнозування ІСЦ.

Основними джерелами історичних даних ІСЦ та супутніх економічних показників в Україні є НБУ, Держстат, Світовий банк та Trading Economics. Доступ до даних можливий через API, вебскрапінг або завантаження. Найкращим є використання API для отримання структурованих даних. Ці дані можуть бути використані в нашому модифікованому підході до програмного прогнозування ІСЦ, який поєднує статистичні методи та алгоритми машинного навчання. Основні етапи включають збір і попередню обробку даних, аналіз стаціонарності, побудову оптимізованих за Акаїке моделей ARIMAX/SARIMAX, перевірку мультиколінеарності, тестування моделей та уточнення прогнозу за допомогою Gradient Boosting. Переваги підходу полягають у врахуванні особливостей даних ІСЦ, використанні сучасних статистичних і машинних методів, а також в оптимізації параметрів моделей. Це дозволяє підвищити точність прогнозів. Впровадження запропонованого підходу може покращити аналіз та

прогнозування макроекономічних показників, сприяючи економічній стабільності.

Обґрунтовано та обрано засоби та технології для розробки вебзастосунку з модулем прогнозування ІСЦ. JavaScript є універсальною мовою для розробки як клієнтської, так і серверної частини. Серед JavaScript-фреймворків виділено React для клієнтської частини завдяки гнучкості та можливостям повторного використання компонентів. Для серверної частини рекомендовано Express – популярний і простий у використанні Node.js-фреймворк. Як СУБД обрано MongoDB для забезпечення гнучкості та масштабованості. Для статистичної обробки даних та побудови моделей прогнозування пропонується використовувати Python зі спеціалізованими бібліотеками для data science. Запропоновано клієнт-серверну MVC-архітектуру, де окремий модуль прогнозування на Python виконує обчислення. Детально описано структуру бази даних з 7 сутностями та алгоритм роботи модуля прогнозування.

Вибрані засоби та технології задовольняють вимоги до програмного прогнозування ІСЦ, забезпечуючи масштабовану архітектуру, продуктивну розробку та підтримку. Виконано оцінку ефективності модифікованого способу прогнозування ІСЦ, тестування вебзастосунку та способи вдосконалення розробленого програмного забезпечення. Для оцінки ефективності модифікованого способу прогнозування ІСЦ було обрано середню абсолютну похибку. Порівняння з класичним методом прогнозування продемонструвало перевагу модифікованого підходу, який показав меншу похибку на рівні 0,475 проти 1,324. Це підтвердило вищу точність, адаптивність та надійність модифікованого способу, що робить його ефективнішим інструментом для економічного моделювання.

Комплексне тестування охоплювало перевірку frontend, backend, API, алгоритмів прогнозування та взаємодії з БД. Особлива увага приділялася точності генерованих прогнозів ІСЦ. Застосування unit-тестів дозволило виявити недоліки на ранніх стадіях розробки. Тестування підтвердило

ефективність розробленого рішення та визначило напрямки вдосконалення: оптимізацію інтерфейсу, покращення алгоритмів, впровадження нових методів аналізу даних та оптимізацію БД. Розроблено вебзастосунок з інтуїтивно зрозумілим інтерфейсом, модулем прогнозування ІСЦ та відкритим API. Конкурентні переваги – висока точність, адаптивність, інтеграція та автоматизація.

Регулярне оновлення та вдосконалення застосунку має відбуватися безперервно для задоволення потреб користувачів та адаптації до змін ринку. Це забезпечить постійний розвиток системи прогнозування ІСЦ. Побудовано бізнес-модель. Аналіз проблеми високого і нестабільного ІСЦ в Україні показує його негативний вплив на економіку та населення. Запропонована бізнес-модель ставить за мету створення програмного забезпечення для точного і оперативного прогнозування ІСЦ, що допоможе вирішити ці проблеми. Основні споживачі – уряди, інвестиційні компанії та великий бізнес. Унікальна ціннісна пропозиція – спеціально розроблений метод прогнозування ІСЦ.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кравчук А. І., Арсенюк І. Р. Інформаційна технологія прогнозування індексу споживчих цін в умовах не стабільної [Електронний ресурс].
Режим доступу: <https://www.economy-confer.com.ua/full-article/5856/>
2. *Olena Harmash*: Ukraine suffers biggest economic fall in independent era due to war: [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.reuters.com/markets/europe/ukraines-economy-falls-304-2022-minister-2023-01-05/#:~:text=%2A%20Ukraine%27s%20GDP%20fell%2030.4,uncertainty%20and%20risks%20remain%20high>
3. Методологічні пояснення щодо організації статистичного спостереження за змінами цін (тарифів) на споживчі товари (послуги) і розрахунку індексу споживчих цін: [Електронний ресурс]. – Режим доступу до ресурсу: https://ukrstat.gov.ua/operativ/operativ2005/ct/cn_rik/isc/isc_u/isc_met.html
4. *Laura E. Jackson*: The Effects of Uncertainty on Economic Outcomes: [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.stlouisfed.org/on-the-economy/2019/july/effects-uncertainty-economic-outcomes>
5. Комплекс моделей для прогнозування індексу споживчих цін: [Електронний ресурс]. – Режим доступу до ресурсу: <https://journal.bank.gov.ua/ua/article/2021/252/01>
6. Коментар Національного банку щодо рівня інфляції в січні 2023 року: [Електронний ресурс]. – Режим доступу до ресурсу: <https://bank.gov.ua/ua/news/all/komentar-natsionalnogo-banku-schodo-rivnya-inflyatsiyi-v-sichni-2023-roku>
7. Forecasting Swiss inflation using VAR models: [Електронний ресурс]. – Режим доступу до ресурсу:

https://www.snb.ch/en/publications/research/economic-studies/2006/12/economic_studies_2006_02

8. Forecasting Inflation in India Using Arima Modelling: [Электронный ресурс]. – Режим доступа до ресурсу:
<https://ijrpr.com/uploads/V4ISSUE9/IJRPR17369.pdf>
9. Forecasting euro area inflation with machine learning models: [Электронный ресурс]. – Режим доступа до ресурсу:
<https://www.ecb.europa.eu/pub/economic-research/resbull/2023/html/ecb.rb231017~b910853393.en.html>
10. *Andrii Babii, Eric Ghysels, Jonas Striaukas*: Econometrics of Machine Learning Methods in Economic Forecasting: [Электронный ресурс]. – Режим доступа до ресурсу:
<https://ar5iv.labs.arxiv.org/html/2308.10993>
11. Stata: [Электронный ресурс]. – Режим доступа до ресурсу:
<https://en.wikipedia.org/wiki/Stata>
12. An Introduction to EViews: [Электронный ресурс]. – Режим доступа до ресурсу: <https://www.eviews.com/Discovering/whatisviews.html>
13. SAS (software): [Электронный ресурс]. – Режим доступа до ресурсу:
[https://en.wikipedia.org/wiki/SAS_\(software\)](https://en.wikipedia.org/wiki/SAS_(software))
14. SPSS: [Электронный ресурс]. – Режим доступа до ресурсу:
<https://en.wikipedia.org/wiki/SPSS>
15. *Will Kenton*: What is a Bloomberg Terminal? Functions, Costs and Alternatives: [Электронный ресурс]. – Режим доступа до ресурсу:
https://www.investopedia.com/terms/b/bloomberg_terminal.asp
16. Autoregressive integrated moving average: [Электронный ресурс]. – Режим доступа до ресурсу:
https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average
17. *Jason Brownlee* : A Gentle introduction to SARIMA for Time Series Forecasting in Python: [Электронный ресурс]. – Режим доступа до

ресурсу: <https://machinelearningmastery.com/sarima-for-time-series-forecasting-in-python/>

18. *Ritu Santra*: Introduction to SARIMA Model: [Електронний ресурс]. –

Режим доступу до ресурсу:

[https://medium.com/@ritusantra/introduction-to-sarima-model-cbb885ceabe8#:~:text=SARIMA%20\(Seasonal%20Auto%2DRegressive%20Integrated,to%20the%20non%2Dseasonal%20components.](https://medium.com/@ritusantra/introduction-to-sarima-model-cbb885ceabe8#:~:text=SARIMA%20(Seasonal%20Auto%2DRegressive%20Integrated,to%20the%20non%2Dseasonal%20components.)

19. *Viktor Mehandzhiyski*: What is an Arimax model : [Електронний

ресурс]. – Режим доступу до ресурсу:

<https://365datascience.com/tutorials/python-tutorials/arimax/>

20. *Brendan Artley*, Time Series Forecasting with ARIMA , SARIMA and

SARIMAX: [Електронний ресурс]. – Режим доступу до ресурсу:

<https://towardsdatascience.com/time-series-forecasting-with-arima-sarima-and-sarimax-ee61099e78f6>

21. Gradient boosting: [Електронний ресурс]. – Режим доступу до

ресурсу: https://en.wikipedia.org/wiki/Gradient_boosting

22. Тест Дікі-Фуллера: [Електронний ресурс]. – Режим доступу до

ресурсу:

https://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D1%81%D1%82_%D0%94%D1%96%D0%BA%D1%96-%D0%A4%D1%83%D0%BB%D0%BB%D0%B5%D1%80%D0%B0

23. Decomposition of time series: [Електронний ресурс]. – Режим

доступу до ресурсу:

https://en.wikipedia.org/wiki/Decomposition_of_time_series

24. Autocorrelation Function: [Електронний ресурс]. – Режим доступу до

ресурсу: <https://www.sciencedirect.com/topics/chemistry/autocorrelation-function>

25. Варіаційний коефіцієнт інфляції: [Електронний ресурс]. – Режим

доступу до ресурсу: https://uk.biennngoccruise.com/variance-inflation-factor#google_vignette

26. Variance inflation factor: [Електронний ресурс]. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Variance_inflation_factor
27. Akaike information criterion: [Електронний ресурс]. – Режим доступу до ресурсу:
https://en.wikipedia.org/wiki/Akaike_information_criterion
28. C Sharp: [Електронний ресурс]. – Режим доступу до ресурсу:
https://uk.wikipedia.org/wiki/C_Sharp
29. Java (programming language): [Електронний ресурс]. – Режим доступу до ресурсу:
[https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))
30. JavaScript language overview: [Електронний ресурс]. – Режим доступу до ресурсу: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Language_overview
31. Python (programming language): [Електронний ресурс]. – Режим доступу до ресурсу:
[https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))
32. R Programming Language – Introduction: [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/r-programming-language-introduction/>
33. Getting Started with Scala Programming: Features and Applications: [Електронний ресурс]. – Режим доступу до ресурсу:
<https://www.simplilearn.com/scala-programming-article>
34. Vue.js: [Електронний ресурс]. – Режим доступу до ресурсу:
<https://en.wikipedia.org/wiki/Vue.js>
35. Angular (web framework): [Електронний ресурс]. – Режим доступу до ресурсу: [https://en.wikipedia.org/wiki/Angular_\(web_framework\)](https://en.wikipedia.org/wiki/Angular_(web_framework))
36. Що таке React JS і для чого він потрібен?: [Електронний ресурс]. – Режим доступу до ресурсу: <https://dan-it.com.ua/uk/blog/chto-takoe-react-js-i-dlja-chego-on-nuzhen/>

- 37.What Is Express JS In Node JS?: [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-express-js>
- 38.Introduction to Koa Javascript (Koa js): [Електронний ресурс]. – Режим доступу до ресурсу: <https://cubettech.com/resources/blog/introduction-to-koa-javascript-koa-js-pros-and-cons/>
- 39.Intro to Hapi: The Node.js framework: [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.infoworld.com/article/3705991/intro-to-hapi-the-nodejs-framework.html>
- 40.What is PostgreSQL?: [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.postgresqltutorial.com/postgresql-getting-started/what-is-postgresql/>
- 41.MongoDB: [Електронний ресурс]. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/MongoDB>
- 42.What is MySQL and How Does It Work: [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.hostinger.com/tutorials/what-is-mysql>
- 43.MVC Framework Introduction: [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/mvc-framework-introduction/>
- 44.Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.
- 45.Методичні вказівки до виконання магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 152 с.

ДОДАТКИ

Додаток А (обов'язковий)

Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬНазва роботи: Інформаційна технологія прогнозування індексу споживчих цін в умовах нестабільної економікиТип роботи: магістерська кваліфікаційна робота
(БДР, МКР)Підрозділ кафедра комп'ютерних наук, ФІТА
(кафедра, факультет)

Показники звіту подібності Turnitin

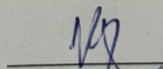
Оригінальність 96,00% Схожість 4,00%

Аналіз звіту подібності (відмітити потрібне):

- ☒ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

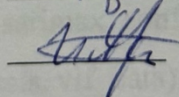
Ознайомлені з повним звітом подібності, який був згенерований системою Turnitin щодо роботи.

Автор роботи



Кравчук А.І.

Керівник роботи

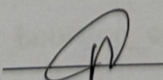


Арсенюк І.Р.

Опис прийнятого рішення

Магістерську кваліфікаційну роботу допущено до захисту

Особа, відповідальна за перевірку



Озеранський В.С.

Додаток Б (обов'язковий)

Лістинг програми

Лістинг 1. forecastingCpi.py

```
from flask import Flask, request, jsonify
from pymongo import MongoClient
import pandas as pd
from statsmodels.tsa.statespace.sarimax import SARIMAX
from sklearn.ensemble import GradientBoostingRegressor
from statsmodels.tsa.stattools import adfuller
from statsmodels.stats.outliers_influence import variance_inflation_factor
from statsmodels.tools.tools import add_constant
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.impute import SimpleImputer
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
import json
import db_entities

app = Flask(__name__)
client = MongoClient('mongodb://localhost:27017/forecastingDB')
db = client['cpi_forecasting']

def adjust_model_based_on_tests(sarimax_config,
gradient_boosting_config, stationary_tests, vif_data):
    if not all(test['stationary'] for test in stationary_tests.values()):
        sarimax_config['d'] = 1
    high_vif_columns = vif_data[vif_data['VIF'] > 10]['feature']
    return sarimax_config, gradient_boosting_config
```

```

def process_data(data):
    df = pd.DataFrame(data)
    numeric_features = df.select_dtypes(include=['int64', 'float64']).columns
    categorical_features = df.select_dtypes(include=['object']).columns
    numeric_transformer = Pipeline(steps=[
        ('imputer', SimpleImputer(strategy='median')),
        ('scaler', StandardScaler())])
    categorical_transformer = Pipeline(steps=[
        ('imputer', SimpleImputer(strategy='constant', fill_value='missing')),
        ('onehot', OneHotEncoder(handle_unknown='ignore'))])
    preprocessor = ColumnTransformer(
        transformers=[
            ('num', numeric_transformer, numeric_features),
            ('cat', categorical_transformer, categorical_features)])
    processed_data = preprocessor.fit_transform(df)
    return processed_data

```

```

def perform_adfuller_test(series):
    result = adfuller(series, autolag='AIC')
    adf_statistic, p_value, _, _, critical_values, _ = result
    return p_value < 0.05, adf_statistic, p_value, critical_values

```

```

def calculate_vif(dataframe):
    vif_data = pd.DataFrame()
    vif_data['feature'] = dataframe.columns
    vif_data['VIF'] = [variance_inflation_factor(dataframe.values, i) for i in
range(dataframe.shape[1])]
    return vif_data

```

```

def configure_sarimax(data, config):

```

```

        return SARIMAX(data, order=(config['p'], config['d'], config['q']),
                        seasonal_order=(config['P'], config['D'], config['Q'],
config['s']))
    def configure_gradient_boosting(config):
        return
GradientBoostingRegressor(n_estimators=int(config['n_estimators']),
                           learning_rate=config['learning_rate'],
                           max_depth=int(config['max_depth']),
                           min_samples_split=int(config['min_samples_split']),
                           min_samples_leaf=int(config['min_samples_leaf']),
                           max_features=config['max_features'])

@app.route('/forecast', methods=['POST'])
def forecast():
    request_data = request.json
    data = request_data['data']
    custom_sarimax_config = request_data.get('sarimax_config')
    custom_gradient_boosting_config =
request_data.get('gradient_boosting_config')

    processed_data = process_data(data)

    if custom_sarimax_config:
        sarimax_config = custom_sarimax_config
    else:
        sarimax_config =
db.sarimax_configuration.find_one({'sarimaxConfigID': 'defaultID'})

    if custom_gradient_boosting_config:
        gradient_boosting_config = custom_gradient_boosting_config

```

```

else:
    gradient_boosting_config =
db.gradientboosting_configuration.find_one({'gradientBoostingConfigID':
'defaultID'})

    sarimax_config, gradient_boosting_config =
adjust_model_based_on_tests(
    sarimax_config, gradient_boosting_config, stationary_tests, vif_data
)

    sarimax_model = configure_sarimax(processed_data, sarimax_config)
    gradient_boosting_model =
configure_gradient_boosting(gradient_boosting_config)
    sarimax_model.fit()
    gradient_boosting_model.fit(X, y) # Замініть X, y на відповідні дані

df = pd.DataFrame(processed_data)
stationary_tests = {}
for column in df.columns:
    stationary, adf_statistic, p_value, critical_values =
perform_adfuller_test(df[column])
    stationary_tests[column] = {
        'stationary': stationary,
        'adf_statistic': adf_statistic,
        'p_value': p_value,
        'critical_values': critical_values
    }

vif_data = calculate_vif(df)
sarimax_model.fit()

```



```

        gradient_boosting_model.fit(X, y)
        sarimax_predictions =
sarimax_model.get_forecast(steps=10).predicted_mean
        gradient_boosting_predictions =
gradient_boosting_model.predict(X_test)

        db_entities.insert_forecast(forecast_data)
        db_entities.insert_sarimax_config(sarimax_config_data)
        db_entities.insert_gradient_boosting_config(gradient_boosting_config_d
ata)
        return jsonify({'sarimax_predictions': list(sarimax_predictions),
                        'gradient_boosting_predictions':
list(gradient_boosting_predictions)})

if __name__ == '__main__':
    app.run(debug=True, port=5000)

```

Лістинг 2. controller.js

```
const express = require('express');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');
const cors = require('cors');
const jwt = require('jsonwebtoken');
const bcrypt = require('bcryptjs');
const userRoutes = require('./routes/userRoutes');
const forecastRoutes = require('./routes/forecastRoutes');
const subscriptionRoutes = require('./routes/subscriptionRoutes');
const nbuService = require('../services/nbuService');
const app = express();
app.use(cors());
app.use(bodyParser.json());
mongoose.connect('mongodb://localhost:27017/forecastingDB', {
  useNewUrlParser: true,
  useUnifiedTopology: true
});
app.use('/user', userRoutes);
app.use('/forecast', forecastRoutes);
app.use('/subscription', subscriptionRoutes);
const port = 3000;
app.listen(port, () => {
  console.log(`The server is running on the port ${port}`);
});
const express = require('express');
const router = express.Router();
router.post('/register', async (req, res) => {
  try {
    const { email, password } = req.body;
```

```

    const existingUser = await UserModel.findOne({ email });
    if (existingUser) {
        return res.status(400).send('A user with this email already exists.');
```

```
    }
```

```
    const hashedPassword = bcrypt.hashSync(password, 8);
```

```
    const newUser = new UserModel({ email, password: hashedPassword
```

```
});
```

```
    await newUser.save();
```

```
    res.status(200).send('Registration is successful.');
```

```
  } catch (error) {
```

```
    res.status(500).send('Error during registration.');
```

```
  }
```

```
});
```

```
router.get('/profile', async (req, res) => {
```

```
  try {
```

```
    const userId = req.userId;
```

```
    const user = await UserModel.findById(userId);
```

```
    if (!user) {
```

```
        return res.status(404).send('User not found.');
```

```
    }
```

```
    res.status(200).send({ email: user.email, createdAt: user.createdAt });
```

```
  } catch (error) {
```

```
    res.status(500).send('Error when receiving a profile.');
```

```
  }
```

```
});
```

```
router.post('/create', async (req, res) => {
```

```
  try {
```

```
    const { userId, forecastInputData } = req.body;
```

```
    const userHasSubscription = await
```

```
SubscriptionService.checkUserSubscription(userId);
```

```

    if (!userHasSubscription) {
        return res.status(403).send('Access is limited. Active subscription
required.');
```

}

```

    const forecastResult = await
PythonForecastService.generateForecast(forecastInputData);

    const newForecast = new ForecastModel({
        userId,
        inputData: forecastInputData,
        result: forecastResult
    });

    await newForecast.save();

    res.status(200).send({ message: 'Forecast successfully created',
forecastId: newForecast._id });

    } catch (error) {
        res.status(500).send('Error when creating a forecast.');
```

}

```

    });

    router.get('/history', async (req, res) => {
        try {
            const userId = req.query.userId;
            const forecasts = await ForecastModel.find({ userId: userId });
            res.status(200).send(forecasts);
        } catch (error) {
            res.status(500).send('Error when retrieving the history of forecasts.');
```

}

```

    });

    router.get('/:forecastId', async (req, res) => {
        try {
            const forecastId = req.params.forecastId;
```

```

    const forecast = await ForecastModel.findById(forecastId);
    if (!forecast) {
        return res.status(404).send('forecast not found.');
```

```
    }
```

```
    res.status(200).send(forecast);
```

```
  } catch (error) {
```

```
    res.status(500).send('Error when receiving a forecast.');
```

```
  }
```

```
});
```

```
router.post('/update', async (req, res) => {
```

```
  try {
```

```
    const { userId, subscriptionDetails } = req.body;
```

```
    const updatedSubscription = await
```

```
SubscriptionModel.findOneAndUpdate(
```

```
    { userId: userId },
```

```
    { $set: subscriptionDetails },
```

```
    { new: true }

```

```
);
```

```
if (!updatedSubscription) {
```

```
    return res.status(404).send('User or subscription not found.');
```

```
}
```

```
    res.status(200).send({ message: 'Subscription has been updated',
```

```
updatedSubscription });
```

```
  } catch (error) {
```

```
    res.status(500).send('Error updating your subscription.');
```

```
  }
```

```
});
```

```
router.get('/status', async (req, res) => {
```

```
  try {
```

```
    const userId = req.query.userId;
```

```

const subscription = await SubscriptionModel.findOne({ userId: userId
});

if (!subscription) {
    return res.status(404).send('Інформація про підписку не знайдена.');
```

}

```

const currentDate = new Date();
const isSubscriptionActive = subscription.endDate > currentDate &&
subscription.status === 'active';

res.status(200).send({
    subscriptionStatus: isSubscriptionActive ? 'active' : 'inactive',
    subscriptionDetails: subscription
});
} catch (error) {
    res.status(500).send('Error checking subscription status.');
```

}

```

});

async function fetchCPI(start, end, period) {
    const url =
`https://bank.gov.ua/NBUStatService/v1/statdirectory/inflation?period=${period}
&start=${start}&end=${end}&json`;

    try {
        const response = await axios.get(url);
        return response.data;
    } catch (error) {
        console.error('Error when receiving CPI data from the NBU API:', error);
        throw error;
    }
}

```

```

router.post('/forecastWithNBUData', async (req, res) => {
  try {
    const exogenousVariables = req.body.exogenousVariables;
    const cpiData = await
nbuService.fetchCPI(req.body.start,req.body.end,req.body.period);
    const forecastInput = { cpiData, exogenousVariables };
    const forecastResult = await
pythonForecastService.generateForecast(forecastInput);

    res.status(200).send(forecastResult);
  } catch (error) {
    res.status(500).send('Error when creating a forecast.');
```

}

```

});

router.post('/forecastWithLocalData', async (req, res) => {
  try {
    const { cpiData, exogenousVariables } = req.body;
    const forecastInput = { cpiData, exogenousVariables };
    const forecastResult = await
pythonForecastService.generateForecast(forecastInput);

    res.status(200).send(forecastResult);
  } catch (error) {
    res.status(500).send('Error when creating a forecast.');
```

}

```

});

module.exports = router;
```

Лістинг 3. App.jsx

```
import React, { useState, useEffect } from 'react';
import { BrowserRouter as Router, Route, Switch } from 'react-router-dom';
import axios from 'axios';
import HomePage from './components/HomePage';
import LoginPage from './components/LoginPage';
import RegisterPage from './components/RegisterPage';
import UserProfilePage from './components/UserProfilePage';
import ForecastCreationPage from './components/ForecastCreationPage';
import ForecastHistoryPage from './components/ForecastHistoryPage';
import ForecastDetailPage from './components/ForecastDetailPage';
import NavigationBar from './components/NavigationBar';
export const AuthContext = React.createContext();
function App() {
  const [isAuthenticated, setIsAuthenticated] = useState(false);
  useEffect(() => {
    axios.get('/api/auth/check')
      .then(response => {
        if (response.data.authenticated) {
          setIsAuthenticated(true);
        }
      })
      .catch(error => console.error('Authorization error ', error));
  }, []);
  return (
    <AuthContext.Provider value={{ isAuthenticated, setIsAuthenticated }}>
      <Router>
        <NavigationBar />
        <Switch>
          <Route path="/" exact component={HomePage} />
```



```
    <Route path="/login" component={LoginPage} />
    <Route path="/register" component={RegisterPage} />
    <Route path="/profile" component={UserProfilePage} />
    <Route path="/create-forecast" component={ForecastCreationPage} />
    <Route path="/forecast-history" component={ForecastHistoryPage} />
    <Route path="/forecast/:id" component={ForecastDetailPage} />
  </Switch>
</Router>
</AuthContext.Provider>
);
}
export default App;
```


Додаток В (обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ДЛЯ ВИЯВЛЕННЯ НТТР
ЗАПИТІВ З АНОМАЛЬНОЮ ПОВЕДІНКОЮ

Рисунок В.1 – Базовий архітектурний набір для розробки інформаційної технології

Виконав: студент 2-го курсу,
групи 2КН-23м
спеціальності 122 «Комп'ютерні науки»
(шифр і назва напрямку підготовки, спеціальності)
 Кравчук А.І.
(прізвище та ініціали)

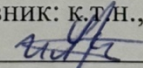
Керівник: к.т.н., доцент каф. КН
 Арсенюк І.Р.
(прізвище та ініціали)
« 05 » 12 2024 р.

Рисунок В.2 – Взаємодія користувача з MVC як базовою

Вінниця ВНТУ - 2024 рік

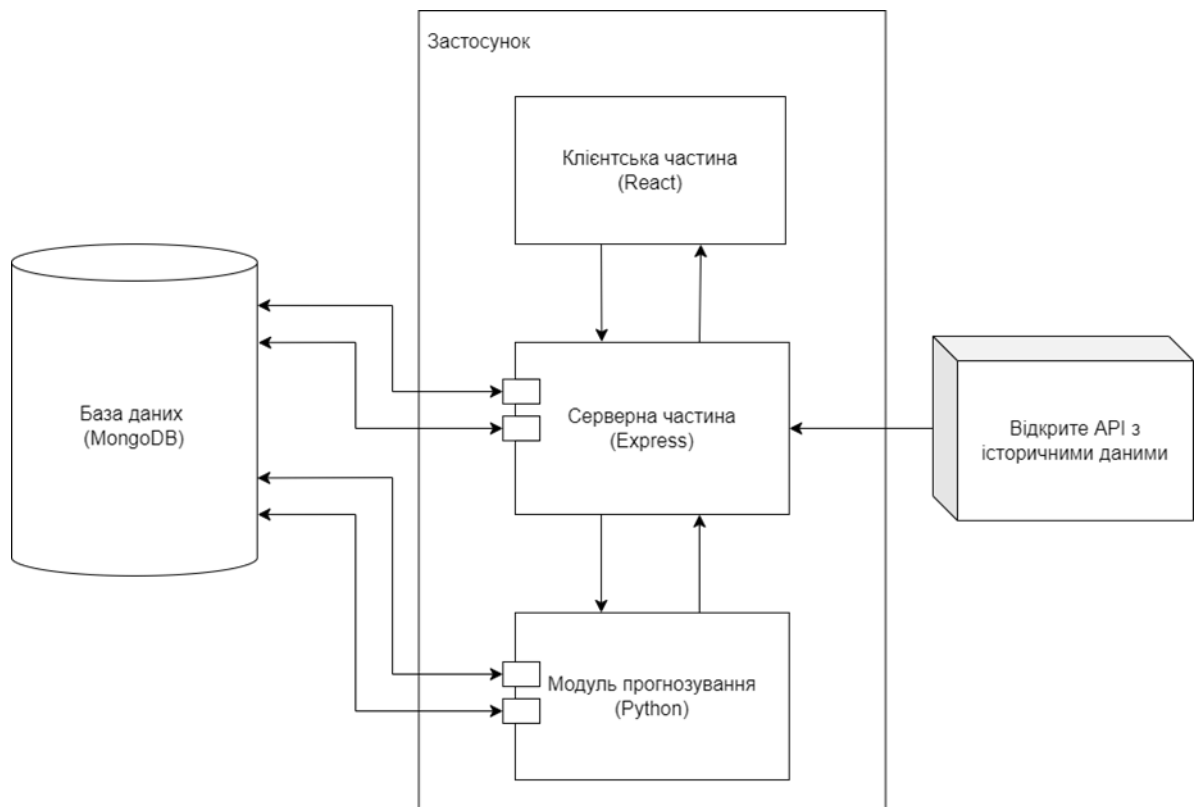


Рисунок В.1 – Багатошарова архітектура розробленого вебзастосунку

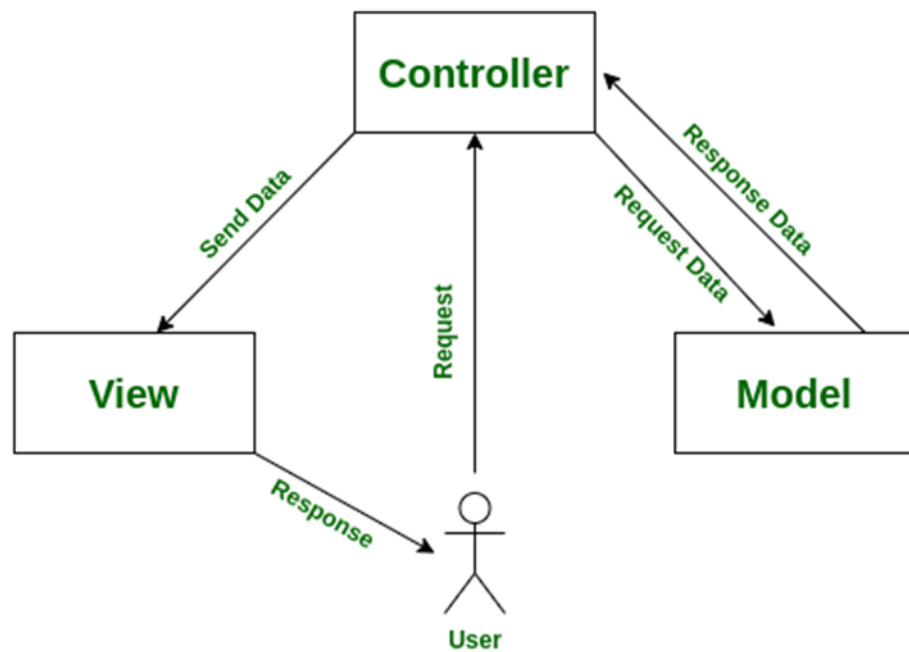


Рисунок В.2 – Взаємодія користувача з MVC системою

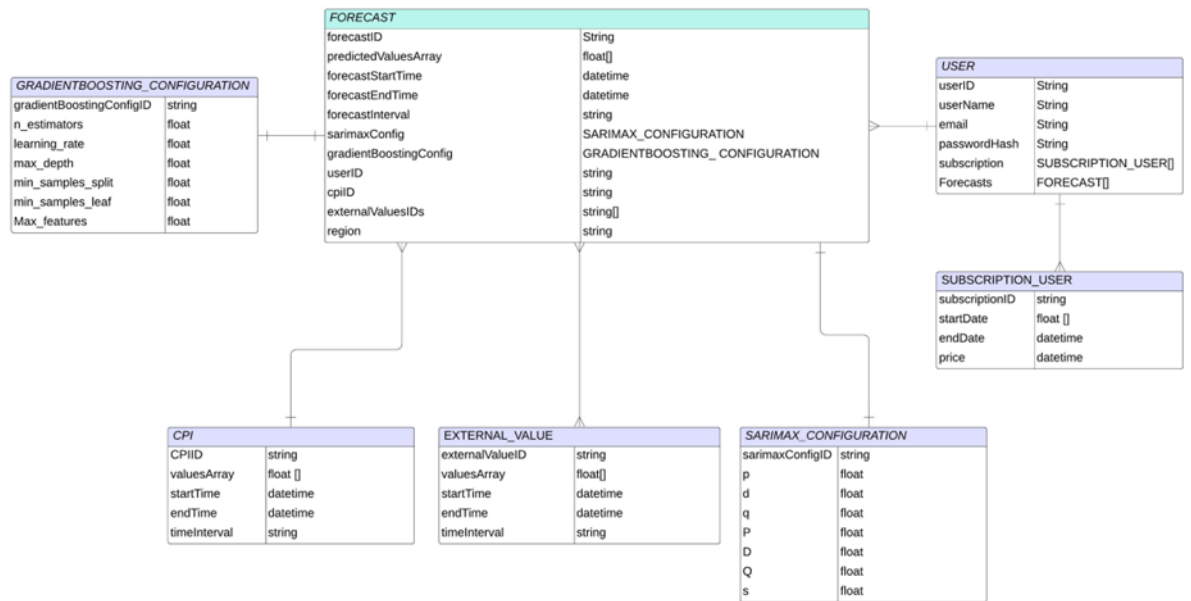


Рисунок В.3 – Загальна схема алгоритму роботи інформаційної технології для виявлення HTTP запитів з аномальною поведінкою

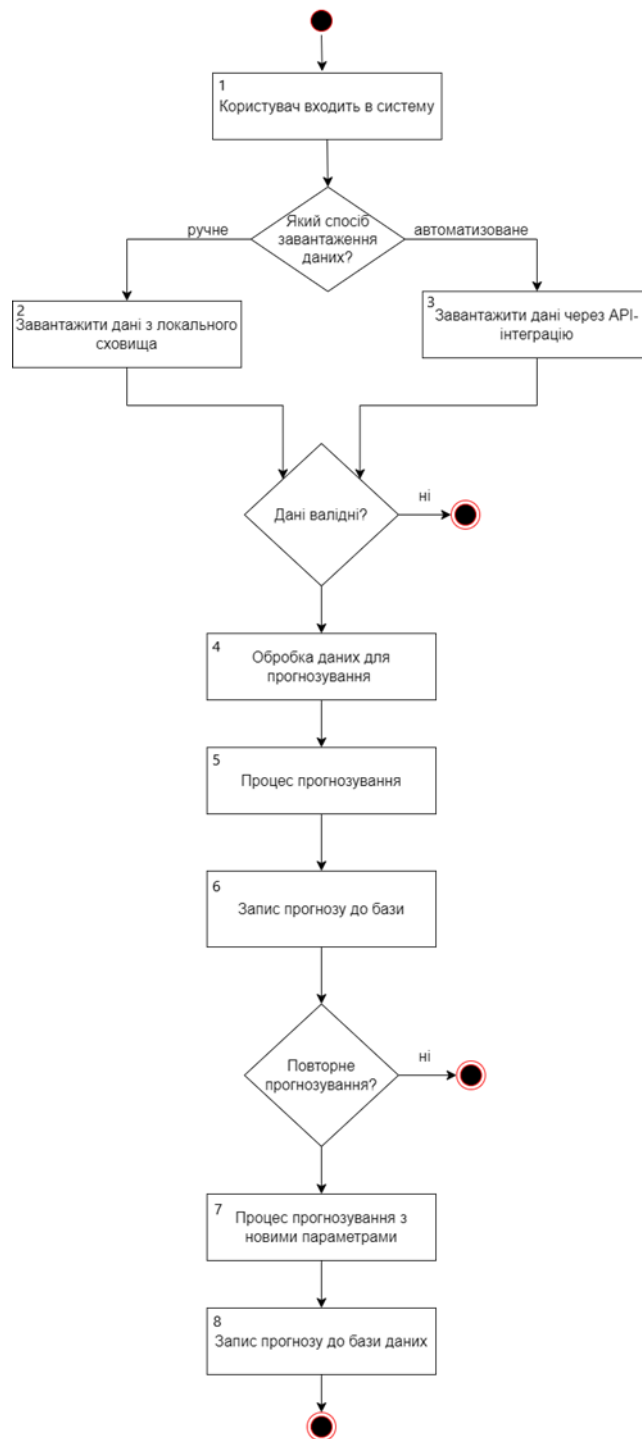


Рисунок В.4 – Алгоритм роботи модулю прогнозування

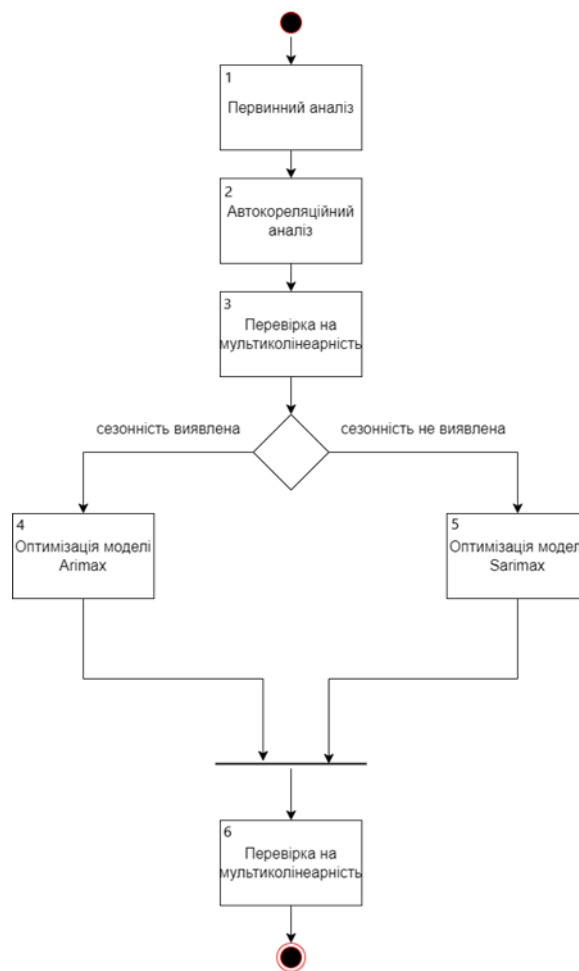


Рисунок В.5 – Алгоритм програмного прогнозування ІСЦ



Рисунок В.6 – Алгоритм первинного аналізу даних

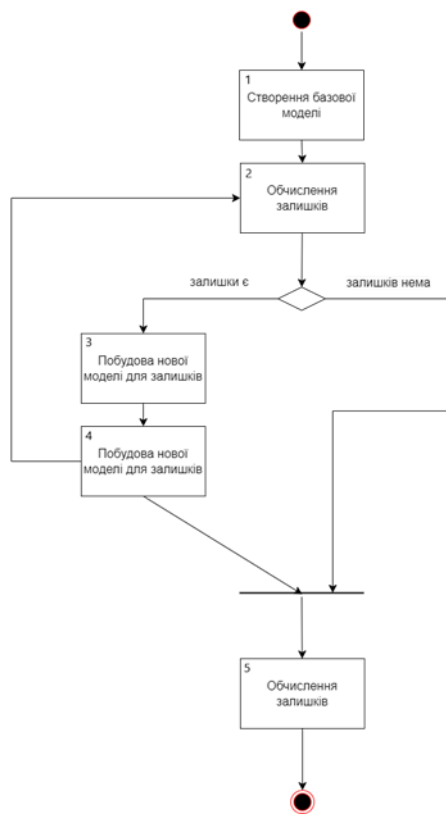


Рисунок В.7 – Алгоритм уточнення прогнозу з Gradient boosting

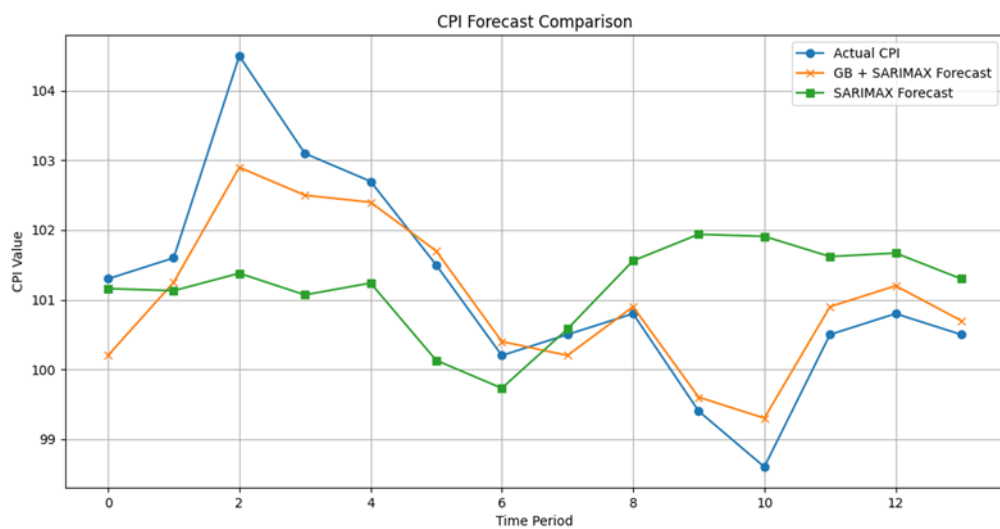


Рисунок В.8 – Графік порівняння точності прогнозування модифікованого та класичного способів

Додаток Г (довідниковий)

Інструкція користувача

Розглянемо два формати користування програмним продуктом, а саме зі сторони звичайного користувача, та з іншої – розробника, якому необхідно розвернути продукт та внести певні зміни.

Для типового користувача немає необхідності володіти технічними знаннями, щоб взаємодіяти з програмним продуктом. Користувачеві достатньо звернутись за налаштованим API ендпоінтом, надіславши файл, який буде проаналізовано.

The screenshot displays a REST client interface for a POST request to the `/analyze` endpoint. The request is configured with a `multipart/form-data` body, containing a required file named `NF-UNSW-NB15.csv`. The interface includes buttons for `Execute` and `Clear`. Below the request configuration, the `Responses` section shows the `Curl` command, the `Request URL` (`http://127.0.0.1:8000/analyze`), and the `Server response`. The response is a `200` status code with a JSON body: `{ "filename": "NF-UNSW-NB15.csv", "info": "Task for the NF-UNSW-NB15.csv analysis has been created." }`. A `Download` button is available for the response body.

```
POST /analyze Analyze-Data
```

Parameters Cancel Reset

No parameters

Request body ^{required} multipart/form-data

file ^{required} string(\$binary) Вибрати файл `NF-UNSW-NB15.csv`

Execute Clear

Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/analyze' \
  -H 'accept: application/json' \
  -H 'Content-Type: multipart/form-data' \
  -F 'file=@NF-UNSW-NB15.csv;type=csv'
```

Request URL

```
http://127.0.0.1:8000/analyze
```

Server response

Code	Details
200	Response body <pre>{ "filename": "NF-UNSW-NB15.csv", "info": "Task for the NF-UNSW-NB15.csv analysis has been created." }</pre> Download

Рисунок Г.1 – Приклад роботи API ендпоїнта

Надалі, користувач у заданому форматі, наприклад, відповідь поштою, отримає у якості відповіді дані про трафік, що було надано у файлі.

Процес аналізу не потребує від користувача дій, тому на цьому його взаємодія з програмним продуктом завершується.

Розроблений програмний продукт потребує розвертання та внесення змін зі сторони розробника, тому варто розглянути способи роботи з продуктом.

Інформаційну технологію імплементовано на базі мікросервісу, а точніше – легковісного та швидкісного серверу, що розроблено використовуючи фреймворк FastAPI. Першочергово, розробнику потрібно підготувати код проєкту, наприклад, скопувати з репозиторію чи використати наданий лістинг.