Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційних радіотехнологій і технічного захисту інформації
(повна назва)
Кафедра Радіотехнологій інформаційно-комунікаційних систем
(повна назва)

# КВАЛІФІКАЦІЙНА РОБОТА
## Пояснювальна записка

рівень вищої освіти другий (магістерський)

ГЮІК.ХХХХХХ.000ПЗ
(позначення документа)

Методи штучного інтелекту для дослідження мутацій COVID-19

(тема)

Виконав:
студент II курсу, групи ІКТм-20-1
Музика Р.В.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Освітня програма Інформаційно-комунікаційні технології
(повна назва освітньої програми)

Керівник проф. Кузьомін О. Я.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ О.І. Цопа
(підпис) (прізвище, ініціали)

2021 р.

Не містить відомостей заборонених для відкритого публікування.

Керівник _____ Кузьомін О.Я.

Студент _____ Музика Р.В.

Харківський національний університет радіоелектроніки

Факультет Інформаційних радіотехнологій і технічного захисту інформації

Кафедра Радіотехнологій інформаційно-комунікаційних систем

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Освітня програма Інформаційно-комунікаційні технології
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

"_____" _____ 2021 р.

# ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Музиці Руслану Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи Методи штучного інтелекту для дослідження мутації COVID-19
затверджена наказом по університету від 05 листопада 2021 р. № 1648Ст
2. Термін подання студентом роботи до екзаменаційної комісії 9 грудня 2021 р.
3. Вихідні дані до роботи розробити штучний інтелект, що буде проводити дослідження різноманітних мутацій коронавірусної хвороби
4. Перелік питань, що потрібно опрацювати в роботі аналіз предметної області, постановка задачі, аналіз існуючих алгоритмів розв'язання задачі, вибір алгоритму розв'язання задачі, розробка структури додатку, розробка алгоритму функціонування, розробка алгоритму достлідження мутацій.
5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) еволюція віруса, структура SARS-CoV-2, структура геному SARS-CoV-2, запропонований підхід SPM, послідовні правила, алгоритм роботи, мутація геному COVID-19

6. Консультанти розділів роботи

| Найменування розділу | Консультант (посада, прізвище, ім’я, по батькові) | Позначка консультанта про виконання розділу | |
|---|---|---|---|
| | | підпис | дата |
| Основна частина | проф. Кузьомін О.Я. | | |
| | | | |

## КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів роботи | Терміни виконання етапів роботи | Примітка |
|---|---|---|---|
| 1 | *Ознайомлення із завданням. Уточнення ТЗ* | 01.09.21 | вик. |
| 2 | *Підбір літератури за темою роботи* | 13.09-22.09.21 | вик. |
| 3 | *Теоретичний розділ* | 23.09-10.10.21 | вик. |
| 4 | *Проектний розділ* | 11.10-25.10.21 | вик. |
| 5 | *Оформлення презентаційного матеріалу,* | 26.10-30.11.21 | вик. |
| | *підготовка до захисту у ЕК* | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Дата видачі завдання __1__ ___вересня___ 2021 р.

Студент _____ Музика Р.В.
(підпис) (прізвище та ініціали)

Керівник роботи _____ проф. Кузьомін О.Я.
(підпис) (посада, прізвище та ініціали)

# РЕФЕРАТ

Пояснювальна записка до магістерської кваліфікаційної роботи: 68 с., 10 табл., 8 рис., 2 додатка, 76 джерел інформації.

БАЗА ДАНИХ, AI, COVID-19, TDAG, DG, SPM, RNA, SARS-CoV-2

Об’єктом досліджень є технології штучного інтелекту для дослідження мутацій COVID-19.

Предметом досліджень є використання технології штучного інтелекту для дослідження мутацій COVID-19.

Мета досліджень – розробка штучного інтелекту.

Методи дослідження – створення, тестування та впровадження нових технологій для дослідження мутацій COVID-19.

В результаті проведених досліджень вирішено задачу створення штучний інтелект за допомогою мови Python. Отримані результати використовуються, як нові можливості для дослідження мутацій COVID-19.

Мова програмування – Python. Пропонована розробка є корисною для вирішення різноманітних проблем, пов’язаних з дослудженнями мутацій COVID-19.

Галузь застосування розробки – ця система буде корисна у медичній сфері.

## ABSTRACT

Explanatory note to the master's qualification work: 68 pages, 10 tables, 8 figures, 2 appendices, 76 sources of information.

DATABASE, AI, COVID-19, TDAG, DG, SPM, RNA, SARS-CoV-2

The object of research is artificial intelligence technologies for the study of COVID-19 mutations.

The subject of research is the use of artificial intelligence technologies to study COVID-19 mutations.

The purpose of research is to develop artificial intelligence.

Research methods - creation, testing and implementation of new technologies for the study of COVID-19 mutations.

As a result, the problem of creating scientific intelligence using the Python language was solved. The results are obtained as new opportunities for the study of COVID-19 mutations.

The programming language is Python. The proposed development is useful for solving various problems related to the experience of COVID-19 mutations.

Scope of development - this system will be useful in the medical field.

ЗМІСТ

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

SPM – Sequential pattern mining – послідовний аналіз шаблонів;

COVID-19 – Coronavirus disease 2019 – коронавірусна хвороба;

AI – Artificial intelligence – штучний інтелект;

РНК – Рибонуклеїнова кислота;

CPT – Compact prediction tree – компактне дерево передбачення;

DG – Dependency graph – граф залежності;

TDAG – Transition directed acyclic graph – Синтаксичний аналізатор ациклічних графів на основі переходів;

SRA – Sequence read archive – архів послідовного читання.

# ВСТУП

Здатність вчених успішно адаптувати вакцини COVID-19 для використання проти варіантів коронавірусу, що викликають занепокоєння , частково вплине на здатність швидко виявляти інфекційні мутації в генетичному складі вірусу. У цьому може допомогти комп'ютер, який розуміє людську мову.

За словами дослідників з Массачусетського технологічного інституту, які використовують алгоритми машинного навчання, розроблені для природної мови, щоб оцінити, які мутації зберігаються, розташування амінокислот, які утворюють вірусні білки, можна аналогізувати з послідовністю слів, які наповнюють такі мови, як англійська. можливість ухилятися від імунного захисту організму.

Дослідники Массачусетського технологічного інституту навчили такі алгоритми для завдання, яке вони називають пошуком обмежених семантичних змін (CSCS), що дозволяє їм вивчати вірусні мутації, у тому числі ті, які переростають у високоінфекційні варіанти коронавірусу, такі як ті, що вперше з'явилися у Великобританії та Південній Африці. Ці ідеї особливо актуальні для таких регіонів, як Африка, де поширення нового коронавірусу серед переважно невакцинованого населення збільшує можливість виникнення відповідних мутацій.

«Гарна аналогія може бути дуже важливою», — нещодавно пояснив Брайан Брайсон, дослідник Бостонського Інституту, Массачусетського технологічного інституту та Гарварду, та один із науковців, які очолювали ініціативу. «Вірус може мутувати, щоб зберегти функції, необхідні для виживання, або зберегти граматику, при цьому вдається виглядати інакше для імунної системи та зазнавати значних семантичних змін».

Брайсон порівнює процес еволюції вірусу зі структурою речення, яке спирається на граматичні правила та послідовність, або семантику, для передачі значення. Така еволюцію зображена на рис. 1.1:

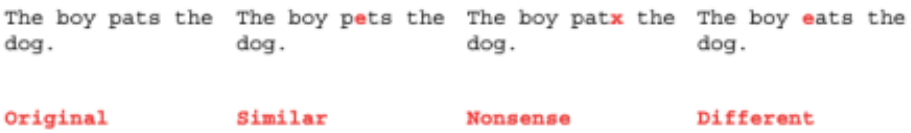


Рисунок 1.1 – Еволюція віруса

Дотримуючись аналогії, вірусна мутація повинна бути граматично правильною і зберігати значення, щоб вона могла успішно відтворюватися. Як і у випадку зі зміною у другому реченні (зліва) вище, так званий білок-шип на поверхні коронавірусу, який дозволяє йому причепитися до рецепторних клітин людини, може трохи мутувати, але все ще нагадувати оригінал достатньо, щоб імунна система могла розпізнати і атакувати його.

Навпаки, білок може відхилятися, як свідчить третє речення зліва, так що, за аналогією, він не є ані граматично правильним, ані має сенс, і більше не може бути «читаним» рецепторами; тобто прив'язати до них. Або, як у випадку з «їсть», у реченні в крайньому правому куті мутація може спостерігати «граматику білка», але змінитися настільки, що антитіла, вироблені імунною системою, більше не можуть зв'язуватися з нею, наче вірус з'являється замаскованим. Це може призвести до більш інфекційного варіанту.

«Ми можемо думати про цей ландшафт, який досліджує вірус, коли він мутує, як предмет обмежень, де ми хочемо зберегти граматику, але змінити семантику, щоб вижити», — говорить Брайсон. «Наша мовна модель вивчає ймовірність певної амінокислоти з урахуванням контексту послідовності».

Брайсон і його колеги навчили алгоритми оцінювати мутації в трьох білках: один на поверхні вірусу грипу, інший на поверхні ВІЛ, а третій на спайку коронавірусу. Для всіх трьох вірусів CSCS виявив мутації, які показали найвищий потенціал для втечі на основі варіацій їх послідовностей. Серед 891 окремої послідовності спайкового білка коронавірусу, які досліджували дослідники, одна була від штаму, який повторно інфікував когось, хто одужав минулого року від

Covid-19. Лише три інші послідовності в наборі показали як вищу семантичну зміну, так і так звану граматику.

Окрім можливості кількісної оцінки можливого втечі мутацій, дослідження може прокласти шлях до вакцин, які розширюють захист організму від варіантів або захищають реципієнтів від більш ніж одного вірусу, наприклад грипу та нового коронавірусу, за один укол.

# АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## Аналіз виникнення вірусу

Перше секвенування геному нової коронавірусної хвороби (COVID-19) було зроблене у січні 2020 року, приблизно через місяць після того, як він був виявлений в Ухані, столиці провінції Хубей, Китай. Секвенування геному COVID-19 має важливе значення для того, щоб зрозуміти поведінку вірусу, його походження, як швидко він мутує, а також для розробки ліків/вакцин та ефективних стратегій профілактики. У цій роботі досліджується використання методів штучного інтелекту завдяки яким є можливість отримати цікаву інформацію з послідовностей геному COVID-19. Послідовний аналіз шаблонів (SPM), зазвичай, спочатку застосовується на зрозумілому комп'ютеру корпусі послідовностей геному COVID-19, щоб побачити, чи можна знайти цікаві приховані закономірності, які виявляють часті моделі нуклеотидних основ та їх взаємозв'язки один з одним. По-друге, моделі передбачення послідовності застосовуються до корпусу, щоб оцінити, чи можна передбачити нуклеотидну основу (основи) з попередніх. По-третє, для аналізу мутацій у послідовностях геному розроблено алгоритм, щоб знайти місця в послідовностях геному, де змінені нуклеотидні основи, та обчислити швидкість мутації. Отримані результати свідчать про те, що методи аналізу SPM та мутацій можуть виявити цікаву інформацію та закономірності в послідовностях геному COVID-19 для вивчення еволюції та варіацій штамів COVID-19 відповідно.

Вірус важкого гострого респіраторного синдрому коронавірусу (SARS-CoV-2), також відомий як COVID-19, вперше був ідентифікований у пацієнта з пневмонією в Ухані, столиці провінції Хубей, Китай, у грудні 2019 року [1]. Всесвітня організація охорони здоров'я (ВООЗ) оголосила, що COVID-19 є надзвичайною ситуацією у світі 30 січня 2020 року [2], а пізніше — пандемією 11

березня 2020 року [3]. Згідно з останньою доповіддю ВООЗ [4], понад 65 мільйонів людей були інфіковані COVID-19, приблизно 1,5 мільйона смертей у всьому світі, і ця хвороба поширилася в більш ніж 200 країнах. Ефективного терапевтичного засобу чи вакцини ще не з'явилося через новизну вірусу та його поведінки. Країни та органи охорони здоров'я вживають та рекомендують профілактичні та ізоляційні заходи для зниження рівня передачі та розмноження. Щоб розробити ефективні терапевтичні засоби або вакцини, які створюють довготривалий імунітет, необхідно зрозуміти геном SARS-CoV-2 та його функціональні можливості.

Геном організму – це загальна сума всього його генетичного потенціалу, що зберігається у вигляді кодованої послідовності, що складається з чотирьох нуклеотидних основ (аденін-A, гуанін-G, цитозин-C і тимін-T), які складають його нуклеїнові кислоти. Послідовність геному COVID-19 складається з одноланцюгової послідовності нуклеотидів, званої РНК, і має довжину приблизно 30 Кб [5]. Визначення послідовності нуклеотидів у геномі називається секвенуванням геному. Геном SARS-CoV-2 був секвенований різними групами по всьому світу, які виявили кілька штамів вірусу і показали, що його геном на 79% схожий на SARS-CoV-1 і на 50% на MERS-CoV (Близький Схід). Респіраторний синдром коронавірус) відповідно [6]. Ідентифікація характеристик геному допомагає біомедичним експертам висунути гіпотези про вплив цих характеристик на прояви захворювання у популяції. Однак це часто повільний і ресурсомісткий процес, який значною мірою залежить від досвіду в області. Наприклад, під час пандемії COVID-19 раннє секвенування геному різних штамів SARS-CoV-2 не перетворилося на своєчасну практичну інформацію, і багато аспектів поведінки хвороби досі невідомі. Використання методів штучного інтелекту, включаючи послідовний аналіз шаблонів (SPM), може прискорити процес пошуку корисних ідей і в кінцевому підсумку сприяти кращому глобальному реагуванні.

Поле аналізу шаблонів забезпечує ефективні комп'ютерні методики, які дозволяють людям, зокрема біоінформатикам, аналізувати складні та великі генетичні та геномні дані [7]. SPM [8], окремий випадок структурованого аналізу

даних, застосовувався в геноміці для пошуку моделей специфічних елементів у генах [9], для аналізу експресії генів [10], для отримання максимальних суміжних частих моделей із наборів даних послідовності ДНК [11], щоб виявити мотиви в послідовностях ДНК [12], передбачити функцію білка [13] та захворювання [14], виявити взаємодії генів та їх характеристики [15], інтерпретувати шаблони, витягнуті з мікрочіпів ДНК [16], добувати k-мерів [17] та побудувати філогенетичне дерево [18]. Використання SPM на послідовних даних геному може дати нове уявлення про мутації вірусу, вірулентність та різні прояви хвороби. Крім того, виявлення важливої прихованої інформації в геномах за допомогою SPM може допомогти прискорити процес біологічних досліджень і має велике значення для біологічного світу.

Загальна мета цієї роботи — дослідити використання методів штучного інтелекту для аналізу геному COVID-19. Точніше, три внески робляться для досягнення цих трьох підцілей.

- Щоб оцінити, чи можна знайти цікаві закономірності в послідовностях геному COVID-19, буде застосований до цих послідовностей SPM. Для цього послідовності геному спочатку перетворюються в корпус, придатний для навчання. Потім на корпусі застосовуються методи SPM, щоб знайти часті нуклеотидні основи (нуклеотиди) та їх моделі в послідовностях геному. Більше того, взаємозв’язки нуклеотидів/патернів один з одним виявляють шляхом послідовного аналізу правил.

- По-друге, щоб оцінити, чи можна передбачити наступні нуклеотидні основи в послідовностях геному COVID-19, тренуються і застосовуються на корпусі сучасні моделі прогнозування.

- По-третє, для аналізу мутацій у послідовності геному пропонується алгоритм пошуку мутацій, які займають місце в послідовностях геному, а також швидкості розповсюдження мутації. Алгоритм застосовується до послідовностей геному COVID-19.

## Довідкова інформація про SARS-CoV-2 та пов’язані з цим роботи

SARS-CoV-2 є бетакоронавірусом з оболонкою, одноланцюговими (позитивною) РНК-геномами зоонозного походження. Їх форма варіюється від сферичної до плеоморфної, а їх довжина становить 80-160нм [19]. SARS-CoV-2 містить чотири структурні білки: (1) шип (S), (2) оболонка (E), (3) мембрана (M) і (4) нуклеокапсид (N) (показано на рис. 1.2). Білки S, M і E утворюють оболонку цього вірусу. Білок E, який є найменшим структурним білком, також відіграє роль у виробництві та дозріванні SARS-CoV-2 [20]. Білки S і M також беруть участь у процесі прикріплення вірусу під час реплікації. N-білки залишаються пов’язаними з РНК, утворюючи нуклеокапсид всередині оболонки. N також бере участь в інших аспектах циклу реплікації вірусу (наприклад, збирання та брунькування) і відповіді клітини-господаря на вірусну інфекцію. Цей вірус назвали коронавірусом через схожий на корону вигляд білка S під мікроскопом.

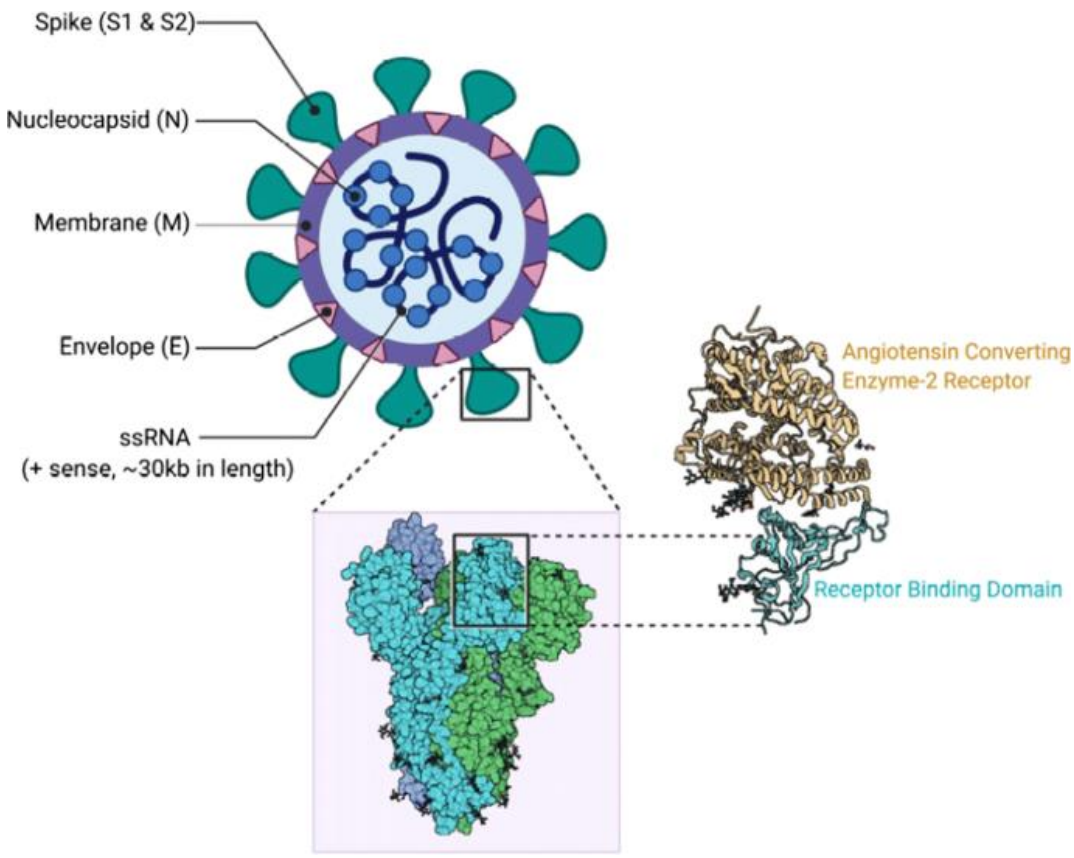


Рисунок 1.2 – Структура SARS-CoV-2

SARS-CoV-2 можна заразитись від людей, та тварин, таких як кажани. Цей вірус може потрапити в організм людини через його рецептори, ACE2, які присутні в різних органах, таких як легені, серце, нирки та шлунково-кишковий тракт. Таким чином, ACE2 полегшує проникнення вірусу в клітини-мішені [22]. Процес потрапляння CoV в клітину-хазяїна починається, коли S-білок, який містить субодиниці S1 і S2, зв'язується з рецептором ACE2 в клітинах-хазяїнах [23]. Таким чином, інфіковані пацієнти відчувають не тільки проблеми з диханням, такі як пневмонія, що призводить до гострого респіраторного дистрес-синдрому (ОРДС), але також страждають розлади серця, нирок і травного тракту [22]. Компактний хребет білка S змушує вірус прикріплюватися сильніше, ніж інші віруси того ж походження, до клітин-господарів. Після того, як білок S зв'язується з рецептором у клітині-мішені, вірусна оболонка зливається з клітинною мембраною і вивільняє вірусний геном у клітину-мішень.

Геномним матеріалом, який виділяє цей вірус, є мРНК. У своєму геномному діапазоні цей вірус доповнюється приблизно від шести до дванадцяти відкритих рамок зчитування (ORFs). Розмір геному SARS-CoV-2 варіюється від 29,8 kb до приблизно 30 kb, а його структура геному відповідає специфічним характеристикам генів відомих CoV. У 5'UTR (термінальна область) більше двох третин геному містить ORF1ab, який кодує поліпротеїни ORF1ab. Тоді як у 3'UTR одна третина складається з генів, які кодують структурні білки (S, E, M і N) (рис. 1.3). SARS-CoV-2 також містить шість додаткових білків, які кодуються генами ORF3a, ORF6, ORF7a, ORF7b, ORF8 і ORF10 [24]. Варто відмітити, що нетрансльовані ділянки (5'UTR і 3'UTR) відповідають за між- і внутрішньомолекулярні взаємодії, взаємодії РНК-РНК і за зв'язування вірусних і клітинних білків [25].

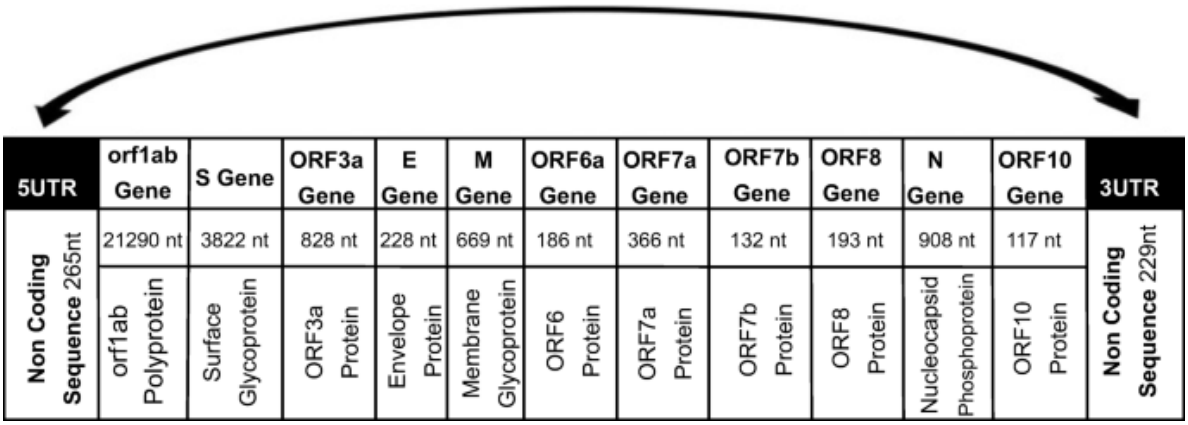


Рисунок 1.3 – Структура геному SARS-CoV-2

Супутні роботи

У цьому розділі обговорюється робота з використання методів на основі штучного інтелекту для діагностики, виявлення, прогнозування та прогнозування COVID-19. Огляд додаткової інформації [26] надав вичерпне розуміння використання математичних моделей та методів на основі штучного інтелекту в дослідженнях COVID-19. Методи штучного інтелекту (машинного навчання,

аналізу даних і глибокого навчання) використовуються в основному для сегментації та діагностики медичних зображень (наприклад, рентгенівська та комп'ютерна томографія (КТ)) [27]. Наприклад, діагностика та виявлення COVID-19 за допомогою комп'ютерної томографії та рентгенівських зображень проводилися з використанням методів глибокого навчання [28, 29, 30, 31, 32, 33] з використанням методів навчання під наглядом, таких як машина опорних векторів (SVM) [34, 35, 36], з використанням логістичної регресії (LR) [37, 38] та використанням дерев рішень (DT), випадкового лісу (RF) у моделях [39, 40] та ARIMA [41].

Для текстових даних, пов'язаних із COVID-19, у дослідженні [42] було проведено тематичний аналіз твітів, пов'язаних із COVID-19, за допомогою програмного забезпечення VOSviewer, щоб вивчити реакцію широкої громадськості, пов'язану зі спалахом COVID-19. Крім того, методи SPM були використані для пошуку частих слів/схем та їх взаємозв'язку в твітах. Швидкість мутацій досліджували [43] у геномних послідовностях, зібраних із даних пацієнтів із COVID-19 від GenBank. Швидкість міссенс-нуклеотидної мутації та швидкість мутації кодона вперше були виявлені в геномах. Після цього для прогнозування майбутньої швидкості мутації цього вірусу була використана модель довготривалої пам'яті (LSTM), заснована на рекуррентній нейронній мережі. У дослідженні автори зосередилися на швидкості мутації базової заміни і не враховували швидкість вставки та делеції. Деякі інструменти також були розроблені [44, 45, 46] для відстеження геномних варіацій SARS-CoV-2. Крім того, моделювання та прогнозування поширення COVID-19 у 5 найбільш постраждалих країнах (Бразилія, Індія, Перу, Росія та США) було зроблено [47], запропонувавши мережу WCGFVL, яка є мережею з випадковим векторним функціональним зв'язком (RVFL).

Більшість досліджень математичного моделювання для COVID-19 зосереджені в основному на динаміці COVID-19 і вивченні впливу методів профілактики, таких як обмеження поїздок, блокування, а також вивчення впливу

клімату на поширення COVID-19. Аналогічно, методи, засновані на штучному інтелекті, дуже добре працюють на тестових даних. Однак відомий факт, що хороша продуктивність алгоритму на тестових даних не гарантує того, що алгоритм буде працювати так само, коли буде розгорнуто на полі. Основна причина цього полягає в тому, що реальні дані більш схильні до шуму та інших артефактів, які зазвичай не присутні в даних навчання та тестування. З іншого боку, в аналізі на основі зображень бракує різноманітних анотованих зображень, які можна використовувати в експериментах [26]. Wynants та ін. [48] розглянли та критично оцінили дослідження, в яких описані моделі прогнозування для COVID-19. Вони стверджували, що про запропоновані моделі звітують погано, дуже упереджені, а продуктивність моделей, ймовірно, оптимістична. Автори припустили, що для суворих моделей прогнозування потрібні дані учасників із досліджень COVID-19, які добре задокументовані. Крім того, нові дослідження та дослідження повинні дотримуватися методологічних вказівок для розробки надійних моделей прогнозування, оскільки ненадійні моделі прогнозів можуть завдати більше шкоди, ніж користі при прийнятті клінічних рішень.

# ВИБІР ІНСТРУМЕНТІВ ДЛЯ РОЗРОБКИ

Загалом, щоб знайти цікаві закономірності в даних, було розроблено і застосовано кілька методів аналізу шаблонів для різних типів наборів даних, починаючи від транзакцій і закінчуючи графіками, рядками та послідовностями [49]. Ці методи були використані в багатьох різних програмах. Однак традиційні методи аналізу шаблонів погано працюють на даних, які залежать від часу або послідовно впорядковані, наприклад послідовності геному. Для таких даних їм не вдається знайти закономірності, що описують послідовні відносини між подіями або елементами. Щоб усунути це обмеження, були розроблені методи для SPM, які можуть міняти шаблони в структурованих послідовних даних [8]. SPM складається з ідентифікації важливих підпослідовностей (шаблів) у наборі дискретних послідовностей, де важливість підпослідовності можна виміряти за допомогою різних показників, таких як частота появи підпослідовності, її прибуток та довжина. Оскільки послідовності геному є формою дискретних послідовностей, варто вибрати методи SPM для їх аналізу.

Для другої підцілі цієї роботи застосовуються сучасні моделі передбачення послідовності, щоб побачити, чи можна передбачити наступні нуклеотидні основи з попередніх у послідовності геному. Розглянуті моделі – компактне дерево передбачення (CPT) [50], CPT+ [51], графік залежності (DG) [52], All-K-Order-Markov (AKOM) [53], Transition Directed Acyclic Graph (TDAG) [54] і LZ78 [55].

Загальний запропонований підхід для аналізу послідовностей геному COVID-19 з використанням моделей SPM та передбачення послідовності зображений на рис. 2.1. Він складається з двох основних частин:

1. **Розвиток корпусу**: послідовності геному COVID-19 перетворюються на корпус дискретних послідовностей, де кожна ціла послідовність геному перетворюється на послідовність нуклеотидів.

2. **Навчання за допомогою методів SPM та передбачення послідовності**: алгоритми SPM застосовуються до корпусу, щоб виявити часто зустрічаються нуклеотиди, послідовні відносини між нуклеотидами та передбачити наступну(і) основу(и) нуклеотидів послідовності.

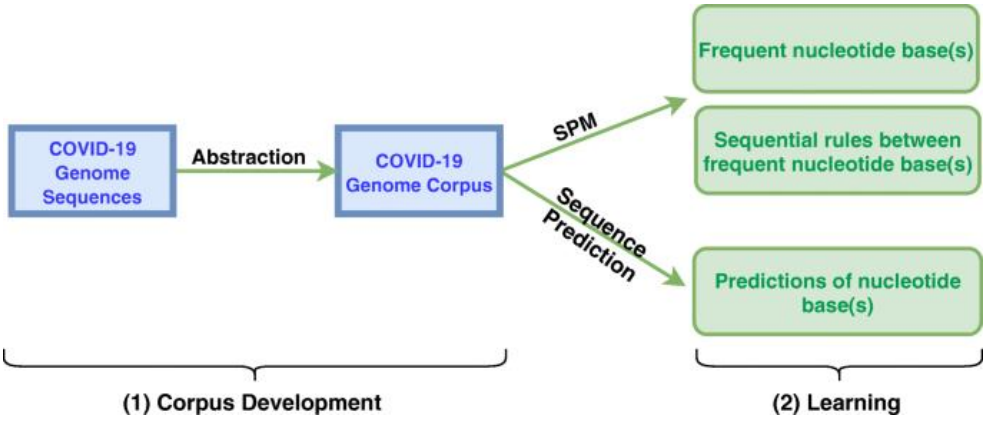


Рисунок 2.1 – Запропонований підхід SPM та прогнозування послідовності для аналізу послідовностей геному COVID-19

Розробка корпусу

Базу даних послідовності геному GenBank [56] використовували для отримання даних секвенування для штамів SARS-CoV-2. GenBank — популярна публічна база даних нуклеотидних послідовностей, яка також підтримує бібліографічні та біологічні анотації. Він підтримується Національним центром біотехнологічної інформації (NCBI) і будується в основному на основі матеріалів від окремих лабораторій і великомасштабних центрів секвенування. За останні два десятиліття GenBank виріс в геометричній прогресії: кількість записів послідовності подвоюється приблизно кожні 18 місяців [56]. Такі онлайн-бази даних дозволяють науковцям-дослідникам у всьому світі швидко аналізувати будь-яку конкретну вірусну структуру, її функції та молекулярну основу. Дані про секвенування генома вірусу, отримані з онлайн-бази даних, також мають вирішальне значення в глобальних зусиллях з розробки вакцин, противірусних препаратів і особливо в точних, чутливих діагностичних тестах. На момент

написання цієї роботі база даних NCBI щодо SARS-CoV-2 містить 43 779 записів нуклеотидів COVID-19 і 112 477 запусків SRA (архів послідовного читання). Для комп'ютерників та біоінформатів кількість даних, пов'язаних із COVID-19, що зберігається в GenBank, є величезною та є у вільному доступі в Інтернеті. Послідовності геному COVID-19 можна розглядати як комп'ютерно зрозумілий корпус.

Щоб застосувати моделі SPM або передбачення послідовності до даних послідовності геному, їх потрібно спочатку трансформувати у відповідний електронний формат, який задовольняє двом основним вимогам, які роблять його придатним для навчання:

- Дані повинні бути перетворені в довгі послідовності елементів (символів), щоб отримати дискретні послідовності, які дозволяють виявити цікаві закономірності в корпусі та виконати точне прогнозування.

- Набір символів, що використовуються для представлення даних у вигляді дискретних послідовностей, повинен бути ретельно відібраний, щоб забезпечити відповідну абстракцію, щоб нерелевантна інформація могла бути пропущена, зберігаючи всю значущу інформацію.

Для виконання цього перетворення використовується абстракція «нуклеотиди до цілих чисел». Вона складається з перетворення кожного нуклеотиду в окремий елемент (символ), представлений у вигляді цілого додатного числа. Ця абстракція є досить загальною і дозволяє застосовувати різні алгоритми SPM, а також моделі передбачення послідовності.

Корпус послідовностей геному COVID-19, отриманий від GenBank [56], представляє кожну послідовність геному у вигляді файлу у форматі FASTA, що містить назви генів, за якими слідує послідовність нуклеотидів (A, C, G і T). Це означає, що після видалення поля генів повна послідовність геному являє собою послідовність нуклеотидів (позначається як $N\ s$ ). Об'єднання всіх цих нуклеотидних послідовностей може створити корпус дискретних

послідовностей. Формально цей корпус визначається так.

**Визначення 1** (набір нуклеотидних основ)

Нехай $NB = \{A, C, G, T\}$ — множина всіх різних нуклеотидних основ. Позначення $|NB|$ позначає потужність множини. Отже, $|NB| = 4$, оскільки існує 4 різних нуклеотиду.

На основі визначення набору нуклеотидних основ послідовність геному COVID-19 представлена таким чином.

**Визначення 2** (послідовність геному COVID-19)

Послідовність геному COVID-19 являє собою впорядкований список нуклеотидних основ, $CGS = \langle NB_1, NB_2, ..., NB_n \rangle$, такий що $NB_i \subseteq NB \ (1 \le i \le n)$.

**Визначення 3** (корпус послідовності геному COVID-19)

*COVID-19* послідовності генома корпус *CGSC* є список послідовностей генома $CGSC = < CGC_1, CGC_2, ..., CGC_p >$, де кожна послідовність генома має унікальний ідентифікатор (ID). Наприклад, у табл. 2.1 показано *CGSC*, що містить чотири рядки (послідовності геному) з ідентифікаторами 1, 2, 3 і 4.

Таблиця 2.1 – Зразок CGSC

| ID | Sequence |
|---|---|
| 1 | ⟨....AATAACTCTATTGCCATACCCACAAATT.....⟩ |
| 2 | ⟨....TGCAGCAATCTTTTGTTGCAATATGGC.....⟩ |
| 3 | ⟨....CAGGTGCTGCATTACAAATACCATTTG.....⟩ |
| 4 | ⟨....CCCTAATGTGTAAAATTAATTTTAGTA.....⟩ |

Варто звернути увагу, що кодон в послідовності геному являє собою послідовність з трьох нуклеотидних основ. Існує $4^3 = 64$ різних кодонів, в яких 61 представляє різні амінокислоти, що входять до складу білків. Решта три кодони представляють сигнали зупинки. Оскільки існує лише 20 різних амінокислот і 61

можливий кодон, більшість амінокислот (крім триптофану та метіоніну) кодуються більш ніж одним кодоном. Наприклад, кодони *GGC* , *GGA* і *GGG* кодують амінокислоту, відому як гліцин. Генетичний код визначає відображення між кодонами та амінокислотами; таким, що кожні три нуклеотидні основи (кодон) кодують одну амінокислоту [57].

Останнім кроком є перетворення послідовностей геному в послідовність цілих чисел, щоб загальні алгоритми SPM можна було застосувати до корпусу. Перед цим кроком кожен рядок містить послідовність нуклеотидів, знайдених у геномі. Кожен нуклеотид у послідовності замінений на додатне ціле число. Наприклад, нуклеотид *A* замінено на 1. Аналогічно, *C, G* і *T* кодуються як 2, 3 і 4 відповідно. Крім того, щоб застосувати деякі алгоритми SPM, між нуклеотидами, такими як від'ємне ціле число -1, і від'ємне ціле число -2, в кінці кожного рядка (рядка) необхідно додати символи-розділювачі [58].

### Навчання з використанням методів SPM та передбачення послідовності SPM

Після підготовки корпусу можна застосувати різні методи SPM для пошуку моделей (підпослідовностей нуклеотидів), які з’являються в послідовностях геному. Але щоб вибрати цікаві візерунки, необхідно використовувати відповідну міру. Найпоширенішим заходом для оцінки закономірностей у аналізі шаблонів є підтримуюча міра (наскільки часто зустрічається) [8, 49]. Цей захід є актуальним для даного дослідження, оскільки дозволяє знайти підпослідовності нуклеотидних основ, які з’являються в численних послідовностях геному, і таким чином виявити їх схожість. SPM з використанням міри підтримки відомий як завдання частого SPM. Як правило, він складається з перерахування всіх частих підпослідовностей у набір дискретних послідовностей [8]. Частий SPM застосовувався для аналізу різних типів даних, таких як текстові документи та послідовності кліків на веб-сторінках. Для контексту аналізу послідовностей геному COVID-19, частий SPM

визначається наступним чином.

**Визначення 4** (Стримування послідовності геному)

Послідовність геному $S_\alpha = \langle \alpha_1, \alpha_2, ..., \alpha_n \rangle$ присутня (або міститься) в іншій послідовності геному $S_\beta = \langle \beta_1, \beta_1, ..., \beta_m \rangle$, якщо існує цілі числа $1 \leq i_1 < i_2 < ... < i_n \leq m$, такі, що $\alpha_1 \subseteq \beta_{i1}, \alpha_2 \subseteq \beta_{i2}, ..., \alpha_n \subseteq \beta_{im} \subseteq \beta_{im}$ (позначається як $S\alpha \sqsubseteq S\beta$). Якщо $S_\alpha$ міститься в $S_p$, то $S_\alpha$ стає підпослідовністю з $S_p$.

**Визначення 5** (Підтримка)

Підтримка генома (суб) послідовність $S_\alpha$ в корпусі *CGSC* є загальною кількістю послідовностей, які містять $S_a$. Він позначається як $sup(S_\alpha)$ і визначається як: $sup(S\alpha) = |\{S | S\alpha \sqsubseteq S \wedge S \in CGSC\}|$.

**Визначення 6** (Часті SPM в корпусі послідовності геному)

Нехай є послідовність генома корпус *CGSC* і певного користувачем мінімального порогу підтримка *minsup*, таким чином, що $minsup > 0$. Завдання частою SPM в *CGSC*, щоб перерахувати всі часті підпослідовності генома . Підпослідовність геному *S* є частою, якщо $sup(S) \geq minsup$.

Наприклад, розглянемо зразок корпусу табл. 2.1. Підпослідовність $\langle A\ A\ T \rangle$має підтримку 4, оскільки вона міститься в чотирьох рядках (послідовності геному). Видобуток частих послідовних моделей у корпусі послідовностей геному COVID-19 не є легким завданням, оскільки послідовності можуть бути дуже довгими і схожими. Послідовність, що містить *n* елементів (нуклеотидів), може мати до $2^n - 1$ окремих підпослідовностей. Це робить наївний підхід до обчислення опори всіх підпослідовностей нездійсненним. За останні роки було розроблено кілька ефективних алгоритмів, які застосовують різні оптимізації для пошуку точного рішення проблеми SPM, не досліджуючи весь простір пошуку.

Алгоритми SPM досліджують простір пошуку шаблонів, спочатку ідентифікуючи всі часті підпослідовності, кожна з яких містить 1 елемент (нуклеотид), що називається 1-послідовністю. Потім алгоритм рекурсивно додає

елементи до цих підпослідовностей, щоб знайти більші підпослідовності. Це робиться за допомогою двох основних операцій, а саме *s*-розширень та *i*-розширень. Ці операції використовуються для створення $(k+1)$-послідовності з однієї або кількох $k$- послідовностей. Важливо зазначити, що SPM можна застосувати до більш загального випадку, ніж описано в цій роботі, коли одночасно дозволено використовувати елементи в послідовності. Однак у цій роботі цей випадок не обговорюється, оскільки нуклеотиди в послідовностях геному завжди повністю впорядковані.

З метою прискорення виявлення послідовних шаблонів і для уникнення пошуку повторюваних послідовностей, алгоритми SPM вимагають визначити повне відношення порядку $\prec$ для елементів. Можна використовувати будь-який загальний порядок, і це не впливає на кінцевий результат, отриманий алгоритмами SPM. Таким чином, у контексті цієї роботи порядок $\prec$ просто визначається на нуклеотидних основах з *NB* як лексикографічний порядок, тобто $A \prec C \prec T \prec G$ .

Алгоритми SPM використовують пошук ушир або пошук у глибину. Пошук в ширину алгоритм спочатку сканує набір даних, щоб знайти часті послідовні шаблони , які містять один елемент (1-послідовностей). Потім алгоритм створює 2-послідовності, виконуючи *s*-розширення та *i*-розширення 1-послідовностей. Аналогічно, 3-послідовності виробляються за допомогою 2-послідовностей і так далі. Цей процес генерації шаблону продовжується до тих пір, поки не вдасться створити жодну послідовність. Тоді як алгоритми пошуку *в* глибину виявляють закономірності з іншим підходом. Алгоритм пошуку в глибину починається з послідовностей, що містять окремі елементи, а потім рекурсивно виконує *i*-розширення та *s*-розширення за допомогою однієї з цих послідовностей, щоб створити більші послідовності. Коли шаблон більше не розширюється, алгоритм повертається, щоб розширити інші шаблони.

Щоб уникнути дослідження всього простору пошуку, алгоритми SPM використовують властивість зменшення простору пошуку, яка називається властивістю Apriori або властивістю антимонотонності. Він стверджує, що для

будь-яких двох послідовностей $s_\alpha$ і $s_\beta$, якщо $s_\alpha$ є підпослідовністю $s_\beta$, то $s_\beta$ має мати опору, рівну або меншу, ніж опору $s_\alpha$. Наприклад, якщо послідовність $\langle A \rangle$ має опору 2, послідовність $\langle A\ C \rangle$ не може мати підтримку, більшу за 2. Властивість Apriori допомагає зменшити простір пошуку, оскільки якщо послідовність зустрічається нечасто, то всі розширення таких послідовностей також нечасті, а отже, не є послідовними шаблонами. Наприклад, якщо $minsup = 3$, немає потреби розглядати будь-які розширення $\langle A \rangle$, оскільки всі вони нечасті.

Основна відмінність між алгоритмами SPM полягає в наступних аспектах:

1. Чи використовується пошук у ширину чи в глибину;
2. Тип представлення бази даних (вертикальний або горизонтальний) і внутрішні структури даних;
3. Як підраховується підтримка шаблонів, щоб знайти ті, що задовольняють обмеження *minsup,* встановлене користувачем.

Деякі репрезентативні та ефективні алгоритми SPM - це SPAM [59], TKS [60] і CM-SPAM [61]. SPAM – це алгоритм пошуку в глибину, який покладається на вертикальне представлення бази даних для пошуку всіх послідовних шаблонів. Використання вертикального представлення дозволяє ефективно розраховувати підтримку шаблонів, не виконуючи багато дорогого сканування бази даних. Алгоритм CM-SPAM [61] є покращеною версією SPAM, яка використовує структуру даних під назвою CMAP (Co-comcurrence MAP) для зменшення простору пошуку та ефективного виявлення послідовних шаблонів. У цій структурі зберігається інформація про одночасне з’єднання елементів. Однак встановлення мінімального порога для застосування SPAM або CM-SPAM до нового набору даних не є інтуїтивно зрозумілим. НалаштуванняЗанадто високий *minsup* може призвести до того, що шаблони не буде знайдено, а занизьке значення може призвести до пошуку мільйонів шаблонів. Щоб усунути це обмеження, було запропоновано розширення CM-SPAM під назвою TKS (Top-k

Sequential), яке безпосередньо дозволяло користувачеві встановлювати кількість шаблонів *k,* які потрібно знайти. Потім TKC виводить TOP - *K* найбільш часто послідовні моделі у вхідному наборі даних. TKS застосовує різні стратегії для зменшення простору пошуку. Розробка алгоритмів SPM є активним напрямом досліджень.

Окрім SPM, іноді також цікаво знайти набори нуклеотидів, які часто з'являються в послідовностях геному без урахування послідовного впорядкування. Для цього розглянуто задачу частого видобутку елементів (FIM) [49], яку можна розглядати як окремий випадок SPM. У контексті цієї роботи FIM визначається наступним чином.

**Визначення 7** (Частий видобуток набору елементів)

Нехай існує корпус послідовності геному *CGSC* і визначений користувачем мінімальний поріг підтримки *minsup,* такий, що $m\ i\ n\ s\ u\ p > 0$. Нехай *NBS* представляє набір *нуклеотидних основ* , такий щоНВ S⊆ NB. *Підтримка* по *NBS* в корпусі *CGSC* є загальна кількість послідовностей , які містять нуклеотиди від NB*CA* . Він позначається як $s\ u\ p\ (\ N\ B\ S\ )$ і визначається як: $s\ u\ p\ (\ N\ B\ S\ ) = |\{\ S\ |\exists\ x \in S\ \forall\ x \in N\ B\ S\ \}|$. Завдання частого видобутку набір елементів у *CGSC* полягає в тому, щоб перерахувати всі часті набори нуклеотидних основ. *NBS* називається частим, якщо $s\ u\ p\ (\ N\ B\ S\ ) \geq m\ i\ n\ s\ u\ p$ .

Наприклад, в табл. 2.1 набір нуклеотидних основ {*A, C, G, T*} є частим, оскільки вони з'являються у всіх чотирьох послідовностях геному.

Першим і найвідомішим алгоритмом для FIM є Apriori [ 62 ]. Він призначений для пошуку частих наборів елементів у великих базах даних. Він продовжується шляхом виявлення загальних елементів, які можна розширити на більші набори елементів, які з'являються досить часто. Набори предметів (*NBS* у цій роботі), вилучені Apriori, також можна використовувати для визначення правил асоціації (відношень) між елементами. Протягом багатьох років було запропоновано кілька швидких і ефективних для пам'яті алгоритмів FIM [49].

Інший тип шаблонів, які розглядаються в цьому дослідженні для аналізу корпусу послідовностей геному, — це послідовні правила . Мотивація пошуку цих закономірностей полягає в наступному. Хоча часті послідовні моделі можуть виявляти часті підпослідовності нуклеотидних основ, деякі моделі можуть бути помилковими, оскільки послідовні моделі виявляються без оцінки чи ймовірності того, що одні нуклеотидні основи слідують за іншими. Таким чином, у деяких випадках послідовні моделі можуть вводити в оману. Алгоритми послідовного аналізу правил виявляють закономірності, беручи до уваги не лише їхню підтримку, а й достовірність [63]. Для послідовностей геному завдання послідовного визначення правил визначається наступним чином.

**Визначення 8** (Правило послідовності)

Послідовне правило X→ YX→Івзаємозв'язок між двома *NB* зX, I⊆ NБ, такий, що $X \cap Y = \emptyset$ і $X$ , $Y \neq \emptyset$ . Правило r : X→ Y означає, що якщо елементи *X* зустрічаються в послідовності, елементи *Y* будуть відбуватися пізніше в тій же послідовності.

**Визначення 9** (Підтримка послідовного правила)

*X* міститься в $S_a$ (записується у виглядіX⊑Sa) якщо $X \subseteq \bigcup_{ni=1}\{\alpha i\}$. Правило r : X→ Y міститься в $S_\alpha$ (r ⊑Sap) якщо існує ціле число *k* таке, що $1 \leq k < n$ , $X \subseteq \bigcup$ і $Y \subseteq \bigcup_{ni=k+1}\{\alpha i\}$. Підтримка правила *r* у корпусі *CGSC* визначається: як:

$$conf_{CGSC}(r) = \frac{|\{S|r \sqsubseteq S \wedge S \in CGSC\}|}{|\{S|X \sqsubseteq S \wedge S \in CGSC\}|}, \quad (2.1)$$

$$sup_{CGSC}(r) = \frac{|\{S|r \sqsubseteq S \wedge S \in CGSC\}|}{|CGSC|}, \quad (2.2)$$

**Визначення 10** (Послідовний аналіз правил)

Нехай існує корпус послідовності геному CGSC і визначена користувачем мінімальна підтримка та мінімальний поріг довіри $minsup > 0$ і $minconf \in [0,1]$. Правило r є частим послідовним правилом, якщо $sup_{CGSC}(r) \geq minsup$ і r є дійсним послідовним правилом тоді, коли $conf_{CGSC}(r) \geq minconf$. Видобуток послідовних правил у корпусі полягає в пошуку всіх дійсних послідовних правил.

Репрезентативним алгоритмом послідовного аналізу правил є ERMiner (послідовність правил на основі класу еквівалентності) [63]. Він спирається на концепцію класів еквівалентності правил, що мають однаковий антецедент і наслідок, а також на вертикальне представлення бази даних для дослідження простору пошуку правил. ERMiner використовує дві операції (ліве і праве злиття) для створення більших правил з менших правил і зменшує простір пошуку за допомогою методу Sparse Count Matrix (SCM). Було показано, що ERMiner є більш ефективним, ніж декілька попередніх алгоритмів послідовного аналізу правил [63].

Методи прогнозування послідовності

Інше навчальне завдання, яке виконується в цьому дослідженні, полягає в побудові моделей прогнозування послідовності з використанням послідовностей геному COVID-19, щоб побачити, чи є розташування нуклеотидних основ передбачуваним. Щоб визначити, яка з них найкраще працює, використовується кілька популярних моделей. Застосовані моделі включають CPT+ [51], CPT [50], DG [52], AKOM [53], Mark1 [64], TDAG [54] і LZ78 [55]. DG [52] — це легка модель на основі Маркова, яка бере в якості вхідних даних набір навчальних послідовностей і обчислює ймовірність того, що за кожним символом слідує кожен символ. Обмеженням DG є те, що тільки останній символ вважається передбачуваним наступного. AKOM [53 ] модель вирішує цю проблему, беручи до уваги останні *k* символів для передбачення (де *k* визначено користувачем). Mark1, модель передбачення Маркова першого порядку, передбачає наступний символ або елемент на основі поточного символу елемента. LZ78 [55] і TDAG [54] використовують підходи стиснення даних для передбачення послідовності.

Компактне дерево передбачення (CPT) та його покращена версія CPT+ є складними моделями. Вони не тільки враховують більше одного символу, але й розглядають різні упорядкування та застосовують стратегії видалення шуму. Однак недоліком є те, що моделі CPT і CPT+ зазвичай потребують великого

обсягу пам’яті. CPT+ приймає набір навчальних послідовностей як вхідні дані та генерує три структури даних: дерево передбачення, таблицю пошуку та інвертований індекс. Ці три структури будуються поступово, розглядаючи кожну послідовність одну за одною під час навчання. Для послідовності геному $S_\alpha$ з $n$ елементів, суфікс $S_\alpha$ розміру $y$, де $1 \leq y \leq n$ визначається як $P_y(S_\alpha) = <\alpha_{n-y+1}, \alpha_{n-y+2},..., \alpha_n>$. Прогнозування наступних нуклеотидних основ (основ) $S_\alpha$ здійснюється шляхом знаходження тих послідовностей, які подібні до $P_y(S_\alpha)$ у будь-якому порядку. Для передбачення CPT+ використовує *консеквент* кожної послідовності, подібний до $S_\alpha$. Нехай $S_\beta$ — інша послідовність геному, подібна до $S_\alpha$. Наслідок $C_\beta$ відносно $S_\alpha$ є найдовшою підпослідовністю $\langle \beta_v, \beta_{v+1},..., \beta_m \rangle$ $S_\beta$ така, що $U_{k=1}^{v-1}\{B_k\} \subseteq P_y(S_\alpha)$ $1 \leq v \leq m$. Кожний базовий нуклеотид (и) виявлено в слідстві аналогічної послідовності генома $S_\alpha$, який зберігається в структурі даних таблиці підрахунку (КТ). Нарешті, CPT+ повертає як передбачення найбільш підтримувані нуклеотидні основи в СТ.

# ЕКСПЕРИМЕНТИ ТА РЕЗУЛЬТАТИ

У цьому розділі представлені результати, отримані шляхом застосування методик, представлених у попередньому розділі, щодо послідовностей геному COVID-19, отриманих з NCBI GenBank. Статистичні дані про зібрані послідовності геному представлені в табл. 3.1, де ID – це номер приєднання послідовності геному. NCBI GenBank пропонує завантажити кожну послідовність у вигляді нуклеотиду, кодуючої області або білка. Послідовності геному були завантажені в нуклеотидній формі.

Таблиця 3.1 – Характеристики генома COVID-19, взятого з NCBI

| ID | Release Date | Length | Location | Collection Date |
|---|---|---|---|---|
| MT745584 | 2020-07-13 | 29860 | Bahrain | 2020-06-22 |
| MT750057 | 2020-07-13 | 29782 | USA:Illinois | 2020-06-17 |
| MT750058 | 2020-07-13 | 29782 | USA: Wisconsin | 2020-06-09 |
| MT291827 | 2020-04-06 | 29858 | China: Wuhan | 2019-12-30 |
| MT291828 | 2020-04-06 | 29858 | China: Wuhan | 2019-12-30 |

Бібліотека аналізу даних SPMF [58], розроблена на JAVA, використовується для аналізу послідовностей геному. SPMF — це кросплатформний фреймворк з відкритим кодом, який спеціалізується на задачах аналізу шаблонів. Він пропонує реалізацію понад 180 алгоритмів аналізу даних. Результати, отримані шляхом застосування алгоритмів на корпусі, представлені в наступних підрозділах.

Часті набори нуклеотидів

Алгоритм Apriori для FIM був вперше застосований до корпусу, щоб знайти

часто зустрічаються набори нуклеотидних основ. Apriori приймає корпус і мінімальний поріг як вхідні дані і виводить часті набори нуклеотидних основ. Потім проводиться етап постобробки, щоб зберегти лише часті набори елементів, що містять один нуклеотид або мають кратні трьом нуклеотидам (довжина кодону). Набори, виділені Apriori з послідовності геному MT745584 для різних значень *minsup* , наведені в табл. 3.2. Для значень *minsup* в діапазоні від 40% до 100% Apriori створив лише чотири часті моделі. При зменшенні *minsup* до 1%, Apriori генерується 15 моделей.

Таблиця 3.2 – Часті набори нуклеотидних основ, виявлені Apriori

| Pattern(s) | Support | Min. Support | Pattern(s) | Support | Min. Support |
|---|---|---|---|---|---|
| A | 8915 | 100% | AGT | 52 | 10% |
| C | 5487 | 100% | ACT | 48 | 5% |
| G | 5859 | 100% | CGT | 32 | 5% |
| T | 9599 | 100% | ACG | 12 | 1% |

Перші чотири моделі показують, що всі нуклеотиди з'явилися у всіх рядках послідовності, що і очікувалося. *A* і *T* становлять 62% послідовності геному (приблизно 30% для *A* і 32% для *T* ). Чотири частих набори нуклеотидів, відкриті Apriori, можна вважати нецікавими для біологів з двох причин. По-перше, часті набори нуклеотидів не впорядковані. Це означає, що вони не дотримуються якогось певного порядку. Наприклад, *AGT* може представляти вісім ($3^3$ - 1) різних кодонів, таких як *TGA* , *GAT* і *GTA*, які мають загальну підтримку 52. По-друге, Apriori не гарантує, що нуклеотиди з набору нуклеотидів з’являються безперервно в послідовності геному. Іншими словами, набір нуклеотидів можна вважати таким, що з’являється в послідовності, якщо в ньому з’являються всі його нуклеотиди, хоча нуклеотиди можуть бути відокремлені один від одного деякими іншими підшаблонами. Наприклад, вважається, що набір нуклеотидів *CGT* з’являється як

в *ACAAGT*, так і в *TAACCGGT* . У цих прикладах, Apriori ігнорує підшаблони нуклеотидів між *C*, *G*, і *T* . Отже, Apriori збільшує значення підтримки *CGT* по одному для кожної такої послідовності, де нуклеотиди не зустрічаються послідовно. Далі представлені результати застосування алгоритмів SPM, які долають два вищезазначені недоліки Apriori, і таким чином розкривають більш значущі закономірності.

Підтримка (частота появи) нуклеотидів у чотирьох інших послідовностях геному COVID-19 наведена в табл. 3.3. Кількість двох нуклеотидів (*A* і *G*) у двох штамів (MT291827 і MT291828) різна. MT291828 має на один *A* менше і один додатковий *G* порівняно з MT291827. Аналіз мутацій цих двох штамів також визначає, що *A* в MT291827 замінено на *G* у MT291828.

Таблиця 3.3 – Відсоток нуклеотидів у геномах COVID-19

| ID | A (%) | C (%) | G (%) | T (%) |
|---|---|---|---|---|
| MT750057 | 8891 (29.853) | 5470 (18.311) | 5849 (19.639) | 9572 (32.140) |
| MT750058 | 8891 (29.853) | 5470 (18.311) | 5849 (19.639) | 9572 (32.140) |
| MT291827 | 8932 (29.914) | 5482 (18.360) | 5859 (19.622) | 9585 (32.101) |
| MT291828 | 8931 (29.911) | 5482 (18.360) | 5860 (19.626) | 9585 (32.101) |

Часті послідовні візерунки

Потім були застосовані алгоритми SPM, щоб знайти приховані послідовні зв’язки між нуклеотидами. Виконано алгоритм CM-SPAM, для якого потрібно встановити мінімальний поріг. CM-SPAM був налаштований так, щоб знаходити лише безперервні послідовні моделі, оскільки шаблони, які пропускають

нуклеотиди, було б важко інтерпретувати, а шаблони, які не кратні трьом нуклеотидам (розміром з кодон), були відфільтровані. У табл. 3.4 наведено деякі з частих шаблонів, виявлених у MT745584 CM-SPAM. Десять візерунків на лівій стороні з'являються для невеликої кількості принаймні 33 % рядків у послідовності. Наприклад, частий шаблон *AATAAC*, з підтримкою 511 з'явився приблизно в 164 рядках послідовності, являє собою два кодони, які кодують амінокислоту аспарагін. Аналогічно, вісім візерунків з правого боку з'явилися щонайменше у 25% рядків, а решта два візерунки з'явилися принаймні в 15% рядків у послідовності. Наприклад, шаблон *ATTATCATA* показує три часті кодони, які кодують амінокислоту Ізолейцин, що має підтримку 416 і які з'явилися в 124 рядках послідовності. Аналогічно *GTTGTGGTAGTG* показує три кодони, де один кодон (*GTG*) з'являється двічі.

Таблиця 3.4 – Часті нуклеотиди, виділені за допомогою CM-SPAM

| Pattern | Support | Min. Sup | Pattern | Support | Min. Sup |
|---|---|---|---|---|---|
| AATAAC | 511 | 33% | AATAAC | 511 | 25% |
| AAAAAG | 530 | 33% | ATTATCATA | 416 | 25% |
| ACTATG | 510 | 33% | AGTAGCTAC | 369 | 25% |
| CAAAAG | 510 | 33% | CAAAAG | 510 | 25% |
| CTTTGT | 523 | 33% | CAATGTCTA | 392 | 25% |
| GTATTA | 508 | 33% | GACTATGTT | 392 | 25% |
| GTATGA | 503 | 33% | GTTGTGGTAGTG | 227 | 15% |
| CAACAA | 499 | 33% | TACTAGAAT | 403 | 25% |
| TTAACG | 499 | 33% | ACCTTAAACTAA | 243 | 15% |
| TCAGTG | 502 | 33% | TCAGTG | 502 | 25% |

З точки зору продуктивності, процес розробки шаблонів був досить

швидким. Таблиця 3.5 показує продуктивність CM-SPAM для різних мінімальних порогових значень. Помічено , що при зменшенні *minsup* CM-SPAM може виявляти більш часті шаблони, тоді як час виконання та використання пам’яті збільшуються.

Таблиця 3.5 – Продуктивність CM-SPAM із різним *minsup*

| Min. Sup % | Time (Sec) | Patterns | Memory (Mb) | Min. Sup. |
|---|---|---|---|---|
| 33% | 2 | 3549 | 45.839 | 493 |
| 25% | 42 | 194361 | 49.256 | 374 |
| 20% | 231 | 1372868 | 48 | 299 |

Також був застосований алгоритм TKS для послідовного аналізу шаблонів top-k. Потрібно корпус і вказаний користувачем параметр *K* в якості вхідних даних і повертає TOP- *K* найбільш часто послідовних патернів, як обсяг виробництва. Параметр *k* використовується замість *minsup* через наступні причини:

1. Вибір правильного значення *minsup* для виявлення бажаної кількості корисних шаблонів впливає на продуктивність алгоритмів SPM.

2. Процес мінімального тонкого налаштування підтримки є важким і тривалим.

Щоб подолати ці недоліки, параметр *k* встановлює межу на загальну кількість шаблонів, виявлених алгоритмом. Деякі найчастіші моделі нуклеотидів, виявлені в MT745584 за допомогою алгоритму TKS різної довжини, наведені в табл. 3.6. Також варто звернути увагу, що закономірності, виявлені алгоритмом CM-SPAM, майже подібні до результатів, отриманих за допомогою алгоритму TKS.

Таблиця 3.6 – Часті послідовні моделі нуклеотидів, виділені за допомогою TKS

| Pattern | Length | Support | Pattern | Length | Support |
|---|---|---|---|---|---|
| ACG | 3 | 594 | GTATTA | 6 | 508 |
| AGT | 3 | 611 | AAATTT | 6 | 537 |
| CGT | 3 | 594 | ATTATCATA | 9 | 416 |
| CTA | 3 | 597 | CTAGGTAAA | 9 | 396 |
| TAC | 3 | 604 | GATAAAGCT | 9 | 396 |
| GTC | 3 | 589 | TACTAGAAT | 9 | 403 |
| CAAAAG | 6 | 510 | CAATGTACG | 9 | 372 |
| TCAGTG | 6 | 502 | AATAAC | 6 | 510 |

Послідовні правила

Потім для пошуку послідовних правил був застосований алгоритм ERMiner. На рис. 3.1 показано деякі правила, знайдені ERMiner в MT745584, що вказують на міцні зв'язки між нуклеотидами. Поріг довіри (*minconf*) був встановлений на рівні 80 % , що означає , що були знайдені правила з довірою не менше 80% (правило X→ Y має 80% точності, якщо за набором нуклеотидів у *X* слідує набір нуклеотидів у *Y* принаймні в 80% випадків, коли *X* з'являється в послідовності геному). Мінімальний поріг підтримки також був встановлений на рівні 80%, і загалом ERMiner створив 43 послідовних правила. На цьому рисунку значення над стрілкою є підтримкою, а значення нижче вказує на достовірність (ймовірність). Наприклад, перше правило на рис. 3.1 вказує, що 93,5 % часу за нуклеотидом *A* слідує нуклеотид *C*нуклеотид. За допомогою ERminer було виявлено кілька цікавих зв'язків між нуклеотидами та кодонами. Наприклад, за нуклеотидами *CG* у 89,6 % часу слідує *A*, щоб утворити кодон *CGA*, який кодує амінокислоту аргінін. Аналогічно, за кодоном *CGT* слідує нуклеотид *A* 86,3 % і 89 % часу відповідно.

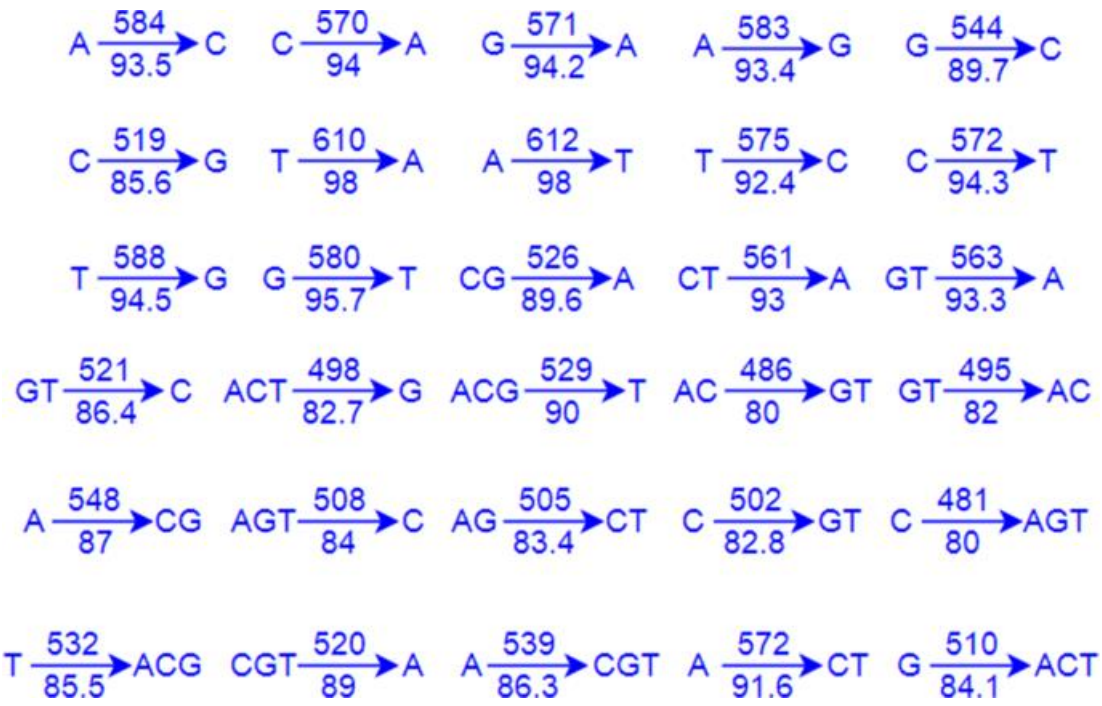


Рисунок 3.1 – Послідовні правила, виявлені в послідовності генома ERMiner

Деякі послідовні правила вказують на конкретний порядок появи між конкретними нуклеотидами і кодонами і навпаки. Наприклад, правила CGT→A і A→CGT правила ACT→G і G→ACT. Також спостерігалося, що загальна кількість нуклеотидів у послідовності (простота абстракції) безпосередньо впливає на ефективність алгоритмів SPM.

Прогноз послідовності

Прогнозування наступних нуклеотидів у послідовності геному було зроблено, щоб побачити, наскільки передбачувана послідовність. Для прогнозування наступних нуклеотидів та їх моделей було порівняно декілька моделей. Кожну модель спочатку навчають на нуклеотидах та їх моделях у послідовностях. Потім модель прогнозування використовується для прогнозування наступних нуклеотидів та їх моделей у послідовності. Прогноз наступних нуклеотидів та їх моделей базується на оцінках, розрахованих моделлю для

кожного нуклеотиду. Наприклад, CPT+ передбачив $\{T\}$ для послідовності $\{A, C\}$, а *ACT* є частим кодоном, який кодує амінокислоту треонін.

Була порівняна продуктивність CPT+ з іншими популярними моделями передбачення, такими як Dependency Graph (DG), Transition Directed Acyclic Graph (TDAG), CPT (попередник CPT+), Mark1, AKOM (All-K-Order-Markov) і LZ78. Кожна модель проходить навчання та тестування з 10-кратною перехресною перевіркою. Техніка перехресної перевірки характеризує продуктивність кожної моделі шляхом оцінки узагальнення незалежного набору щодо статистичних результатів, наданих моделлю. У k-кратній перехресній перевірці набір даних випадковим чином розбивається на *k* піднаборів даних. Потім один піднабір даних вибирається як набір перевірки для тестування моделі, а решта $k-1$ піднаборів використовуються для навчання моделі. Цей процес застосовується *k*разів, і кожен піднабір даних використовується рівно один раз як набір перевірки. Одиничну оцінку результату отримують шляхом взяття середнього з *k* результатів. Основною причиною використання 10-кратної перехресної перевірки є досягнення низької дисперсії під час кожного запуску. Деталі про набір даних, який містить лише MT745584, наведено в табл. 3.7.

Таблиця 3.7 – Статистика корпусу для передбачення послідовності

| parameter | Value |
|---|---|
| Number of Sequences | 1492 |
| Number of distinct items | 5 |
| Itemsets item ID | 4 |
| Distinct item per sequence | 20.42 |
| Occurrence for each item | 4.66 |
| Corpus size in MB | 5968 |

Для оцінки моделей прогнозування використовуються три міри. Результатом

прогнозу може бути:

- *успіх* , якщо модель пророкує точно,
- *провал* , якщо модель передбачає неточно і
- *відсутність відповідності,* якщо модель не може виконати передбачення.

Таблиця також містить інформацію про час навчання та час тестування для кожної моделі в секундах. Загалом, точність є найважливішим показником для порівняння моделей, оскільки вона відображає здатність робити хороші прогнози.

Результати наведені в табл. 3.8 для кожної порівнюваної моделі на MT745584. Щоб представити результати в перспективі, таблиця 3.8 також містить результати для базової лінії, яка є моделлю передбачення послідовності, яка випадковим чином передбачає наступний нуклеотид послідовності. Ця базова модель називається випадковою. За результатами робиться кілька спостережень. По-перше, виявлено, що NoMatch завжди дорівнює нулю, що означає, що передбачення завжди можна виконувати з використанням усіх моделей. По-друге, час тестування та навчання моделей був досить подібним для більшості моделей і залишався дуже низьким (менше 1 секунди у всіх випадках). По-третє, AKOM, де $k = 3$, забезпечив найвищу точність (20,71 %) порівняно з іншими моделями прогнозування, за яким слідує DG. CPT+ мав найбільшу кількість відмов, тоді як CPT+, CPT та TDAG мали подібну продуктивність. CPT, попередник CPT+, працював трохи краще, ніж CPT+. Причина, чому CPT і CPT+ не працюють особливо добре, полягає в тому, що вони вважають, що впорядкування попередніх нуклеотидів не є важливим для прогнозування наступного. Найточніші моделі (DG, Mark1 і AKOM) розглядають суворе впорядкування нуклеотидів (нуклеотидів) для виконання передбачень.

Таблиця 3.8 – Точність моделей прогнозування

| Models | DG | TDAG | CPT+ | CPT | Mark1 | AKOM | LZ78 | Random |
|---|---|---|---|---|---|---|---|---|
| Success | 20.643 | 18.901 | 18.035 | 18.298 | 20.174 | 20.71 | 19.722 | 16.1 |
| Failure | 79.357 | 81.099 | 81.965 | 81.702 | 79.826 | 79.29 | 80.228 | 83.9 |
| No Match | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Train Time | 0.011 | 0.081 | 0.039 | 0.002 | 0.005 | 0.057 | 0.021 | – |
| Test Time | 00.001 | 00.000 | 0.305 | 0.469 | 0.001 | 00.000 | 0.002 | – |

Загалом, точність моделей для передбачення нуклеотидних основ у послідовностях геному була низькою. Це може бути пов’язано з тим, що послідовності містять лише чотири окремих елементи (нуклеотиди), а їх розподіл у послідовності геному нерівномірний (*A* і *T* зазвичай становлять 60-64% від загальної послідовності геному). Незважаючи на низьку точність, усі моделі досягли кращої продуктивності, ніж модель випадкового прогнозування, яка мала точність 16,1%.

Обмеженням цих моделей передбачення є те, що вони можуть передбачити лише один елемент (нуклеотид у цій роботі) для послідовності елементів. Генетичний код, який відображає кодони на амінокислоти, відповідає властивості надмірності (або виродженості) [65]. Це означає, що два різних кодони можуть кодувати одну і ту ж амінокислоту. Надмірність зазвичай виявляється в третьому нуклеотиді в кодонах. Одним практичним наслідком надмірності є те, що заміна одного нуклеотиду (так звана синонімічна заміна) або помилка в третьому положенні зазвичай не призводить до зміни амінокислот у кодованому білку. Через надлишкову природу генетичного коду можна стверджувати, що передбачення лише одного нуклеотиду в кодоні не є таким важливим і корисним.

Проте кодони можуть бути організовані в 9 сімейств і 13 пар на основі частих

моделей надмірності. У 9 сімейств кодонів достатньо перших два нуклеотида для кодування унікальної амінокислоти. Додавання будь-якого третього нуклеотиду (скажімо, *X* ) утворить ту саму амінокислоту. Наприклад, два сімейства кодонів (шаблони) *CGX* і *GCX* кодують дві амінокислоти аргінін і аланін відповідно. Тоді як у 13 пар кодонів достатньо перших два нуклеотида для кодування двох різних амінокислот. Додавання третього пуринового нуклеотиду (який містить *A* або *G* ) (скажімо, *Y* ) утворює одну амінокислоту, а додавання третього піримідину (містить *C* або *T*) нуклеотид (скажімо, *Z* ) утворює іншу амінокислоту. Наприклад, амінокислота лейцин кодується сімейством кодонів ( *CTX* ) і парою кодонів ( *TTY* ). Таким чином, цікава можливість дослідження полягає в тому, щоб скористатися перевагами надлишкових частих шаблонів, виявлених алгоритмами SPM, для передбачення сімейств і пар кодонів. Іншим цікавим напрямком було б інтегрувати знання предметної області в моделі прогнозування для подальшого керування прогнозуванням.

Аналіз мутації геномів COVID-19

На даний момент залишається незрозумілим, яким чином COVID-19 викликає різноманітні захворювання, які можуть варіюватися від безсимптомних до смертельної дихальної недостатності. Як і багато інших організмів, що діляться і поширюються, вірус SARS-CoV-2 постійно розвивається, змінюючи кілька букв (нуклеотидів) за раз, щоб краще адаптуватися до нових умов. Процес еволюції не повністю відомий, оскільки він змінюється повільно [66] порівняно з іншими вірусами, таким чином, дає менше мутацій для вивчення. У середньому коронавірус накопичує близько двох змін на місяць у своєму геномі [67]. Більшість змін у структурі геному COVID-19 можуть не впливати на поведінку вірусу, але деякі можуть впливати на передачу або тяжкість захворювання. Наприклад, Korber et al. [46] стверджував, що мутація (D614G), схоже, більше передається між людьми, ніж попередня (D614). Однак це дослідження викликало критику, оскільки

вчені не довели, що сама мутація відповідальна за її домінування; вона могла отримати користь від інших факторів або від випадку. Тим не менш, дуже важливо зрозуміти закономірність мутацій вірусу, а також швидкість його мутації.

Швидкість мутації будь-якого вірусу є критичним параметром для розуміння вірусної еволюції [68]. Це також найважливіший фактор для оцінки ризику виникнення інфекційних захворювань, і його точна оцінка має велике значення [69]. Крім того, для розробки відповідних ліків/вакцин проти COVID-19 вирішальне значення мають геномна послідовність та аналіз мутацій [70] і точна інформація про частоту мутацій може відігравати життєво важливу роль в оцінці можливих ліків/стратегій вакцинації. У зв'язку з цим можна запропонувати алгоритм, який буде можливо використовувати для аналізу послідовностей геному на наявність варіацій, а також для вивчення частоти мутацій. У цій роботі увага зосереджена на заміщенні мутації, також відомої як точкова мутація. На рисунку 3.2 зображений алгоритм, який представляє псевдокод для аналізу точкових мутацій у послідовностях геному.

**Input**: Genome sequences ($GN_1$, $GN_2$)
**Output**: Locations in the sequences with changed nucleotides, mutation rate

1: Vec $\leftarrow \emptyset$;
2: $TL \leftarrow$ total lines in $GN_1, GN_2$;
▷ $len(GN_1) = len(GN_2)$
3: $x, y \leftarrow 0$
4: **for** $k \leftarrow 1$ to $TL$ **do**
5: **for** $i \leftarrow 1$ to length($TL$) **do**
6: **if** $GN_1(i) \neq GN_2(i)$ **then**
7: $Vec \leftarrow k, i, GN_1(i), GN_2(i)$;
8: $x \leftarrow x + 1$
9: **end if**
10: $y \leftarrow y + 1$
11: **end for**
12: $y \leftarrow y + 1$
13: **end for**
14: $MR \leftarrow \frac{x}{y} \times 100$
15: **return** $Vec, MR$

Рисунок 3.2 – Алгоритм роботи

Цей алгоритм бере дві послідовності геному COVID-19 ( $GN_1$ і $GN_2$ ) і порівнює нуклеотиди в двох послідовностях рядок за рядком. Місця та номери рядків, де нуклеотиди відрізняються, зберігаються в наборі, який називається *Vec* . Крім того, змінені значення нуклеотидів також зберігаються у *Vec* . Швидкість мутації ( *MR* ) розраховується за такою формулою:

$$MR = \frac{TM}{TNB} \times 100, \quad (3.1)$$

де *TM* – загальна мутація, що має місце в двох послідовностях, а *TNB* – загальна кількість нуклеотидів.

Цей алгоритм був розроблений на Python. Алгоритм був протестований для двох послідовностей геному (MT750057 і MT750058) з табл. 3.1. Алгоритм повертає номер рядків і місця, де змінилися нуклеотиди в послідовності геному (показано в табл. 3.9). Крім того, четвертий і восьмий стовпці табл. 3.9 надають інформацію про заміну нуклеотидних основ. Наприклад, перший запис у четвертому стовпці показує, що *G* (у MT750057) було замінено на *T* (у MT750058).

Таблиця 3.9 – Результати аналізу точкових мутацій

| Line | Location | Position | Change | Line | Location | Position | Change |
|---|---|---|---|---|---|---|---|
| 3 | 29 | 149 | G → T | 130 | 42 | 7,782 | C → A |
| 70 | 31 | 4,171 | T → C | 139 | 55 | 8,335 | A → G |
| 94 | 37 | 5,617 | T → C | 328 | 2 | 19,662 | T → G |
| 115 | 11 | 6,851 | C → T | 415 | 31 | 24,871 | G → T |

Цікаво помітити, що частота зустрічальності кожної нуклеотидної основи (A, C, G і T) однакова в двох штамах MT750057 і MT750058 (як зазначено в табл. 3.3), незважаючи на те, що відбувалися мутації. Розроблений алгоритм виявив, що штам MT750058 має на вісім змін більше, ніж штам MT750057. Причина, чому частота нуклеотидів залишається незмінною, незважаючи на ці мутації, полягає в тому, що

для кожної мутації, яка змінила один нуклеотид на інший, була інша мутація, яка змінила інший нуклеотид на перший. Точніше, мутації видалили один *A*, два *C*, два *G* і три *T*, але додали однакову кількість кожного нуклеотиду. Отже, загальна частота кожного нуклеотиду однакова в обох штамах.

Бібліотека matplotlib була використана для створення графіків для двох послідовностей геному та мутованої послідовності (рис. 3.3). Якщо є якась мутація в двох послідовностях геному (рис. 3.3 a, b), то мутована послідовність (рис. 3.3 c) матиме яскраві плями. Якщо мутації немає, то рис. 3.3 в буде темним без яскравих плям. Обидві послідовності мали довжину 497 рядків, і кожна лінія містить 60 нуклеотидів, за винятком останнього рядка, який містить 22 нуклеотиди. Вісь X представляє місце в рядку, де змінені нуклеотидні основи, а вісь Y представляє номер рядка, де змінено дві послідовності.

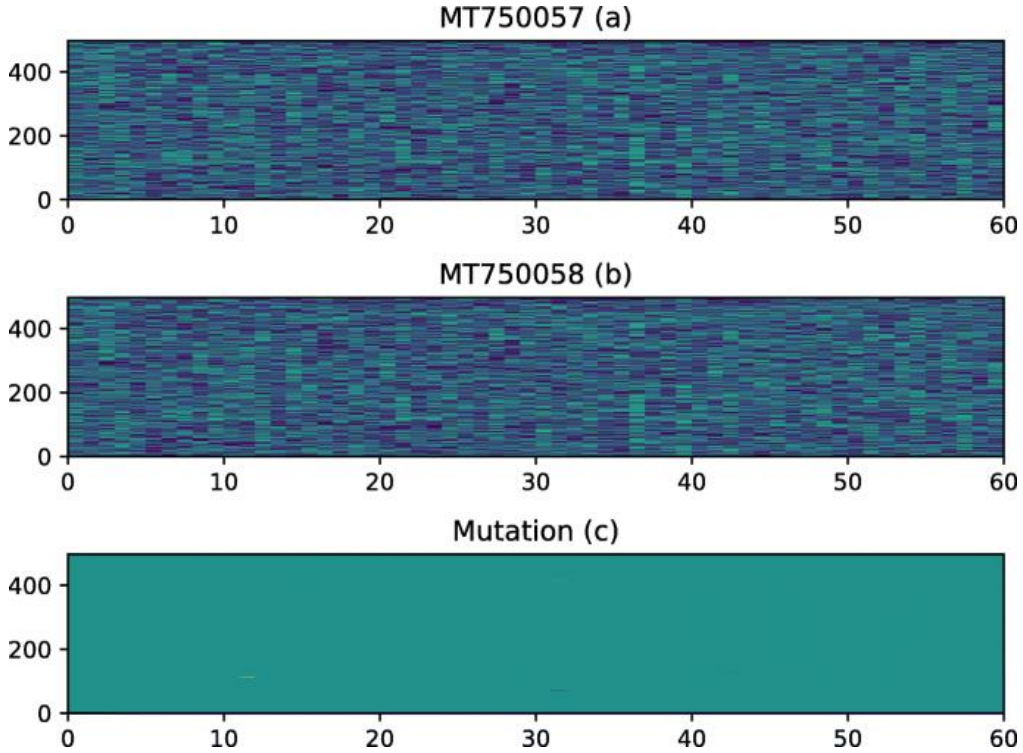


Рисунок 3.3 – Мутація геному COVID-19 у цілих послідовностях

Рисунок 3.3 не дає дуже чіткої картини мутації. Деякі плями можна побачити на рис. 3.3 (c), які показують місце мутацій. Щоб зробити результати більш

зрозумілими, нижче наведений графік аналізу мутації лише для тих рядків, де має місце мутація, замість того, щоб відображати графіки для всіх рядків у послідовностях. Отримані результати представлені на рис. 3.4, який містить лише 8 рядків з двох послідовностей, де мали місце мутації.

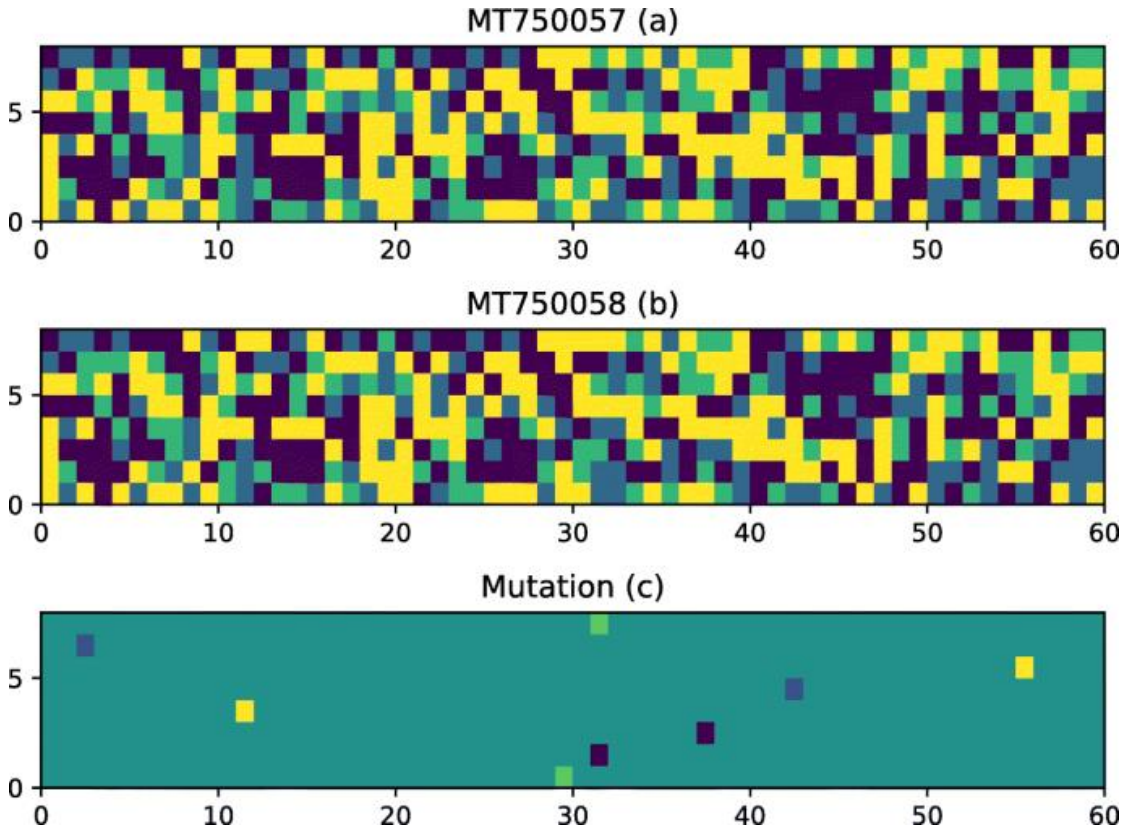


Рисунок 3.4 – Мутація геному COVID-19

Рівень мутацій становить 0,0268% для послідовностей геному MT750057 і MT750058. Аналогічно, швидкість мутації для інших двох послідовностей геному (MT-291827, MT291828) виявляється 0,0003% (одна нуклеотидна основа змінена (A (в MT291827) → G (у MT291828)) у рядку 405, місце 48, позиція 24,288). Таблиця 3.3, де перелічено загальну частоту чотирьох нуклеотидів у MT291827 і MT291828, також вказує, що MT291828 має на один *A* менше і на один *G* більше.ніж MT291827. Важливо зазначити, що послідовності геному Китаю були повідомлені раніше, ніж послідовності геному США. Більше того, послідовності геному з одного міста. Тоді як послідовності геному для США з

різних міст. Це показує, що частота мутації послідовностей геному COVID-19 різна і зростає з плином часу. Більше того, послідовності геному для різних місць (міст) в одній країні мають високий рівень мутації. За допомогою цієї розробленої процедури можна проаналізувати:

- Точкову мутація та швидкість мутації в різних послідовностях геному.
- Як розвиваються послідовності геному COVID-19, коли він поширюється в різних місцях.

Маючи таку інформацію, можна дослідити, як змінюються мутації від місця до місця та від країни до країни. Крім того, штами COVID-19 з різних місць можна проаналізувати, щоб дослідити, чи співіснують вони один з одним, чи ні. У зв'язку з цим, одне дослідження [72] виявило, що європейські, північноамериканські та азіатські штами COVID-19 можуть співіснувати.

Ця попередня техніка має деякі обмеження. По-перше, вимога полягає в тому, щоб обидві послідовності геному мали однакову довжину. Якщо довжина неоднакова, то деякі нуклеотиди в найдовшій послідовності геному не будуть враховуватися в аналізі. Наприклад, нехай довжина коротшої послідовності геному (скажімо X) дорівнює 85, а довжина довшої послідовності геному (скажімо, Y) дорівнює 95. Тоді нуклеотиди в Y від позиції 86 до 95 будуть ігноруватися як максимальна довжина X дорівнює 85. Одним з можливих рішень є додавання фіктивних значень в X. Однак це рішення страждає від проблеми порівняння фіктивних значень з нуклеотидами. Іншим рішенням є зробити довжину X рівною довжині Y, додавши ті нуклеотиди в хвості X, які взяті з Y. Однак, це зробить дві послідовності однаковими в доданих місцях, і аналіз мутації в цих точках буде марним.

Друге обмеження полягає в тому, що процедура аналізу мутацій враховує всю послідовність геному без інформації про ORF та білків. У цьому відношенні методику можна вдосконалити шляхом порівняння нуклеотидів у послідовностях геному ORF. Це допоможе знайти мутацію в окремих ORF і побудувати результати

аналізу мутацій для кожного ORF [73].

# ОБГОВОРЕННЯ МУТАЦІЙ

Оскільки SARS-CoV-2 є РНК-вірусом, він безперервно розвивається в людських популяціях з часом, що сприяє його масовій передачі по всьому світу. Через генетичну різноманітність вірусу, а також геномні варіації пацієнта, тяжкість COVID-19 сильно варіюється від пацієнта до пацієнта. Велика частина пацієнтів або залишаються безсимптомними, або мають легкі або помірні симптоми. За даними когортного дослідження, середній вік пацієнтів, які померли після госпіталізації, становив 70 років, з попередніми медичними проблемами, такими як цукровий діабет та ожиріння. Ця різниця у тяжкості захворювання від однієї людини до іншої пов’язана з багатьма факторами залежно від рівня вірусу, генетичних факторів людини та рівня здоров’я, таких як гіпертонія, діабет, ожиріння та дисфункція печінки. Дані геномної послідовності відкривають великі можливості для вивчення молекулярних змін у зростаючій вірусній популяції, надаючи нове уявлення про спосіб поширення, різноманітність під час пандемій та динаміку еволюції. Нинішнє дослідження було спрямоване на вивчення накопичення вірусних мутацій у всьому геномі в різні моменти часу, щоб ідентифікувати мутації, які відбуваються в усьому світі, і передбачити швидкість мутації вірусу в майбутньому.

Мутаційні профілі 259044 ізолятів SARS-CoV-2 з грудня 2019 року по грудень 2020 року розпізнали загалом 3334545 мутацій із середнім числом 14,01 мутації на зразок. Було виявлено, що кожен із 17221, 17084, 14983 і 14801 зразків містить 18, 17, 16 і 19 мутацій відповідно. Серед найбільш мутованих 20 зразків індійський зразок мав максимальну кількість мутацій 48, за яким слідували зразки (маючи до 36 мутацій) з Шотландії, США, Нідерландів, Норвегії та Франції. Виникнення такої великої кількості мутацій у великій кількості зразків вказує на більш швидку молекулярну еволюцію SARS-CoV-2, що, ймовірно, є причиною його більшої смертельної втрати з точки зору часу. Поряд з цим, було

помічено, що кількість мутацій експоненціально збільшується щомісяця від появи SARS-CoV-2 до грудня 2020 року (48 мутацій), що свідчить про те, що вірус зберігає свою мутаційну природу та постійно розвивається у випадковому порядку. З аналізу філогенетичного дерева помітно, що в більшості країн Європи, Азії, Африки та США переважають ізоляти, що мають приблизно 50% переходу C > T, 14% G > T. транс- версія і 11% A > G перехід. Аналіз природи мутацій SARS-CoV-2 підтверджує збережений молекулярний механізм мутаційної еволюції SARA-CoV-2, оскільки міссенс-мутації (52,35%) є найбільш поширеними мутаційними подіями з точки зору тривалого часу, за якими йдуть мовчазні SNP (37,02%) та екстрагенні SNP (10,12%). У одномм з широкомасштабних досліджень раніше повідомлялося про міссенс-мутації D614G та P314L також визначено як найбільш поширену мутацію у вірусному геномі. Мутація P314L в RdRp пов'язана з мутацією D614G і може сприяти SARS-CoV-2, посилюючи його здатність до передачі. Крім того, були виявлені 152 міссенс-мутації у всьому вірусному геномі, де 46 мутацій були передбачені як шкідливі, які можуть вплинути на структуру вірусного білка, отже, змінюючи стабільність взаємодій між білком і білком, що в кінцевому підсумку може вплинути на проникнення вірусу до хазяїна.

Виявлено, що мутація F106F є переважно тихою мутацією в NSP3, що свідчить про можливу роль у процесуванні мРНК, яка може змінити природу вірусного білка. Крім того, мутація 5′ UTR:C241T може бути пов’язана зі швидкістю транскрипції та реплікації SARS-CoV-2, оскільки виявлено, що вона зустрічається найбільш частішен. Поряд із раніше знайденою мутацією GGG > AAC, аналіз мутаційного типу визначає деякі інші багатонуклеотидні мутації CC > TT, TG > CA та AT > TA, які входять до 20 найбільших мутаційних типів і повинні контролюватися в майбутньому як GGG > AAC (R203K і G204R). Повідомляється, що це пов’язане з введенням лізину в домен SR білка N, що може впливати на фосфорилювання. Крім D416G, F106F, P314L та 5′ UTR:C241T, широкомасштабний аналіз також ідентифікує C22227T;L93L (мембранний білок), G29645T;A222V (шипковий білок), G21255C; A199A (NSP132), і мутації T445C;

A220V (білок нуклеокапсиду), які входять до 10 найпопулярніших мутацій, виявлених в одному з досліджень, і повинні мати значення для оцінки їх ролі в ефективності передачі SARS-CoV-2. D1118H, (S194L і R262H), (M809L, P314L, A8D і S220G), (A890D, G1433C і T1456I), R233C, F263S, L111K, (A54T, і A83T, і A74T, L. Було виявлено, що міссенс-мутації L46C, V48G, Q57H, W131R, G172V, Q185H і Y206S) значною мірою знижують структурну стабільність спайка, нуклеокапсиду, РНК-залежної РНК-полімерази, NSP3, NSP6, NSP15, NSP15, NSP18, NSP18 білки , NSP5 і ORF3a, відповідно і припускає, що ці міссенс-мутації можуть зменшити інфекційність вірусу. Навпаки, було виявлено, що міссенс-мутації D3L, L5F та S97I значною мірою підвищують структурну стабільність білків нуклеокапсиду, ORF7a та ORF8, відповідно, і припускають, що ці мутації можуть підвищити вірусну інфекційність.

Як би там не було, оскільки SARS-CoV-2 постійно мутує, завдяки природному відбору з’являться нові штами. Поява нових мутацій може вплинути на розробку нових методів лікування і навіть погіршити адаптацію поточних методів лікування для позбавлення від нових варіантів SARS-CoV-2. Поява нових мутацій може посилити передачу вірусу. Наприклад, можна помітити, що після появи коронавірусу в грудні 2019 року вірус із великою кількістю мутацій (до 30) був ідентифікований протягом 1 року в Індії, Шотландії, США, Нідерландах, Норвегії, Ізраїлі, Італії, Англії та Франції, що свідчить про найшвидше поширення субпопуляції SARS-CoV-2 у всьому світі.

# ВИСНОВКИ

У цій роботі запропоновано два підходи до дослідження та аналізу послідовностей геному COVID-19. У першому підході методи визначення шаблонів використовуються для пошуку частих нуклеотидних основ у послідовностях, їх частих шаблонів і послідовних відносин між такими шаблонами. Крім того, різні моделі передбачення послідовності були оцінені на послідовностях геному, де AKOM (All-K-Order-Markov) працював краще, ніж інші сучасні алгоритми. У другому підході був запропонований алгоритм аналізу мутацій у послідовностях геному COVID-19. Алгоритм знаходить місце(я) у штамах COVID-19, де нуклеотидні основи змінені, щоб обчислити швидкість мутації. Підходи, представлені в цій статті, не обмежуються вірусом SARS-CoV-2. Вони також можуть бути використані для аналізу інших вірусів людини.

- Використання методів видобутку нових шаблонів або методів аналізу набору контрастів [74] на послідовностях геному COVID-19, щоб виявити нові (або контрастні) тенденції в послідовностях геному, які показують чітку та корисну різницю (або контраст) між двома класами або непересічними ознаками.
- Дослідити застосовність методів аналізу зразків і глибокого навчання [75] для передбачення сімейства кодонів і пар кодонів у послідовностях геному.
- Щоб знайти конкретні кодони, за якими слідують конкретні кодони. Це дозволить нам знайти сигнатури кодонів, які вказують на липкість/перевагу між кодонами амінокислот.
- Щоб скористатися перевагами надлишкових частих моделей у геномах COVID-19, виявлених алгоритмами SPM, для прогнозування сімейства та пар кодонів.
- Удосконалити підхід до аналізу мутацій, щоб зробити його більш загальним. Наприклад, запропонувати деякі стратегії, які можна

використовувати для подолання обмеження довжини послідовності та врахування інформації про гени в послідовностях геному. Крім того, проведення аналізу мутації (вставка або видалення нуклеотидних основ) [76] разом із точковою (заміщення) мутацією.

- Техніку точкової мутації можна розширити, щоб порівняти нову послідовність геному з набором даних геномних послідовностей COVID-19. Кінцевою метою є розробка техніки, яка може добре працювати на послідовностях геному різної довжини, виконувати інделі та точкові мутації з інформацією про гени.

# ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Новий коронавірус пов'язаний з респіраторною хворобою людини в Китаї [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: https://www.nature.com/articles/s41586-020-2008-3.

2. Всесвітня організація охорони здоров’я оголосила надзвичайний стан у світі: огляд нового коронавірусу (COVID-19) 2019 року [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: https://www.sciencedirect.com/science/article/pii/S1743919120301977?via%3Dihub (дата звернення 13.09.2021).

3. Cucinotta D. WHO declares COVID-19 a pandemic / D. Cucinotta, M. Vanelli, 2020. – С. 157–160.

4. WHO (Accessed on December 6, 2020) WHO coronavirus disease (COVID-19) dashboard.

5. Mousavizadeha L. Genotype and phenotype of COVID-19: Their roles in pathogenesis [Електронний ресурс] / L. Mousavizadeha, S. Ghasemi. – 2020. – Режим доступу до ресурсу: https://www.sciencedirect.com/science/article/pii/S1684118220300827?via%3Dihub (дата звернення 14.09.2021).

6. Lu R. Genomic characterisation and epidemiology of 2019 novel coronavirus / Lu // Implications for virus origins and receptor binding / Lu.. – С. 565–574.

7. Chaki J. Pattern analysis of genetic and genomics: a survey of the state-of-art / J. Chaki, N. Dey. – С. 11163–11194.

8. Fournier-Viger P. A survey of sequential pattern mining / Fournier-Viger, 2017. – С. 54–77.

9. Abouelhoda M. String mining in bioinformatics / M. Abouelhoda, M. Ghanem // Scientific Data Mining and Knowledge Discovery-Principles and Foundations / M. Abouelhoda, M. Ghanem., 2010. – С. 207–247.

10. Zihayat M. Mining significant high utility gene regulation sequential patterns. [Електронний ресурс] / Zihayat M, Davoudi H, An A. – 2017. – Режим

доступу до ресурсу: https://bmcsystbiol.biomedcentral.com/articles/10.1186/s12918-017-0475-4 (дата звернення 15.09.2021).

11. Karim MR. An efficient approach to mining maximal contiguous frequent patterns from large DNA sequence databases [Електронний ресурс] / Karim MR. – 2013. – Режим доступу до ресурсу: https://genominfo.org/journal/view.php?doi=10.5808/GI.2012.10.1.5

12. Hsu C. Efficient discovery of structural motifs from protein sequences with combination of flexible intra- and inter-block gap constraints / Hsu C // Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining / Hsu C., 2006. – С. 530–539.

13. Wang M. Sequential pattern mining for protein function prediction. In: Proceedings of Advanced Data Mining and Applications / Wang M, Shang X, Li Z – 2008. – С. 652–658.

14. Kawade D. Exploration of DNA sequences using pattern mining / D. Kawade, K. Oza. – 2013. – №2. – С. 144–148.

15. Cellier P. Sequential pattern mining for discovering gene interactions and their contextual information from biomedical texts [Електронний ресурс] / Cellier. – 2015. – Режим доступу до ресурсу: https://jbiomedsem.biomedcentral.com/articles/10.1186/s13326-015-0023-3 (дата звернення 16.09.2021).

16. Sallaberry A. Sequential patterns mining and gene sequence visualization to discover novelty from microarray data [Електронний ресурс] / Sallaberry. – 2011. – Режим доступу до ресурсу: https://www.sciencedirect.com/science/article/pii/S1532046411000669?via%3Dihub (дата звернення 21.09.2021).

17. Zhang J. Efficient mining closed k-mers from DNA and protein sequences / Zhang // Proceedings of BigComp / Zhang., 2020. – С. 342–349.

18. Kang Y. PVTree: A sequential pattern mining method for alignment

independent phylogeny reconstruction [Електронний ресурс] / Kang. – 2019. – Режим доступу до ресурсу: https://www.mdpi.com/2073-4425/10/2/73 (дата звернення 20.09.2021).

19. Sapokta A. Structure and genome of SARS-CoV-2 (COVID-19) with diagram [Електронний ресурс] / Sapokta. – 2020. – Режим доступу до ресурсу: https://microbenotes.com/structure-and-genome-of-sars-cov-2/ (дата звернення 26.09.2021).

20. Schoeman D. Coronavirus envelope protein: Current knowledge [Електронний ресурс] / D. Schoeman, B. Fielding. – 2019. – Режим доступу до ресурсу: https://virologyj.biomedcentral.com/articles/10.1186/s12985-019-1182-0 (дата звернення 13.10.2021).

21. Cascella M. Features, evaluation and treatment coronavirus (COVID-19) [Електронний ресурс] / Cascella. – 2020. – Режим доступу до ресурсу: https://www.ncbi.nlm.nih.gov/books/NBK554776/ (дата звернення 18.10.2021).

22. Astuti I. Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2): An overview of viral structure and host response [Електронний ресурс] / Astuti. – 2019. – Режим доступу до ресурсу: https://www.sciencedirect.com/science/article/pii/S1871402120300849?via%3Dihub (дата звернення 09.09.2021).

23. Xu H. High expression of ACE2 receptor of 2019-nCoV on the epithelial cells of oral mucosa [Електронний ресурс] / Xu. – 2020. – Режим доступу до ресурсу: https://www.nature.com/articles/s41368-020-0074-x (дата звернення 13.11.2021).

24. Khailany R. Genomic characterization of a novel SARS-CoV-2 [Електронний ресурс] / R. Khailany, M. Safdar, M. Ozaslanc. – 2020. – Режим доступу до ресурсу: https://doi.org/10.1016%2Fj.genrep.2020.100682 (дата звернення 11.11.2021).

25. Yang D. The structure and functions of coronavirus genomic 3’ and 5’ ends [Електронний ресурс] / D. Yang, J. Leibowitz. – 2020. – Режим доступу до

ресурсу: https://doi.org/10.1016%2Fj.virusres.2015.02.025 (дата звернення 10.09.2021).

26. Mohamadou Y. A review of mathematical modeling, artificial intelligence and datasets used in the study, prediction and management of COVID-19 [Електронний ресурс] / Y. Mohamadou, A. Halidou, P. Kapen. – 2020. – Режим доступу до ресурсу: https://doi.org/10.1007/s10489-020-01770-9 (дата звернення 10.10.2021).

27. Shi F. Review of artificial intelligence techniques in imaging data acquisition, segmentation and diagnosis for COVID-19 [Електронний ресурс] / Shi. – 2020. – Режим доступу до ресурсу: https://doi.org/10.1109/RBME.2020.2987975 (дата звернення 13.10.2021).

28. Xu X. A deep learning system to screen novel coronavirus disease 2019 pneumonia [Електронний ресурс] / Xu. – 2020. – Режим доступу до ресурсу: https://doi.org/10.1016/j.eng.2020.04.010 (дата звернення 19.10.2021).

29. Apostolopoulos I. Automatic Detection from X-ray images utilizing transfer learning with convolutional neural networks [Електронний ресурс] / I. Apostolopoulos, T. Mpesiana. – 2020. – Режим доступу до ресурсу: https://doi.org/10.1007%2Fs13246-020-00865-4 (дата звернення 19.11.2021).

30. Mukherjee H. Deep neural network to detect COVID-19: One architecture for both CT scans and chest X-rays [Електронний ресурс] / Mukherjee. – 2020. – Режим доступу до ресурсу: Hhttps://doi.org/10.1007/s10489-020-01943-6.

31. Ozturk T. Automated detection of COVID-19 cases using deep neural networks with X-ray images / Ozturk, 2020. – С. 121.

32. Singh D. Classification of COVID-19 patients from chest CT images using multi-objective differential evolution-based convolutional neural networks [Електронний ресурс] / Singh. – 2020. – Режим доступу до ресурсу: https://doi.org/10.1007%2Fs10096-020-03901-z (дата звернення 15.10.2021).

33. Marques G. Automated medical diagnosis of COVID-19 through EfficientNet convolutional neural network. [Електронний ресурс] / Marques. –

2020. – Режим доступу до ресурсу

34. Barstugan M. Coronavirus (COVID-19) classification using CT images by machine learning methods [Електронний ресурс] / M. Barstugan, U. Ozkaya, S. Ozturk. – 2020. – Режим доступу до ресурсу: https://arxiv.org/abs/2003.09424 (дата звернення 25.10.2021).

35. Batista A. COVID-19 diagnosis prediction in emergency care patients: A machine learning approach [Електронний ресурс] / AFdM Batista. – 2020. – Режим доступу до ресурсу: https://www.medrxiv.org/content/10.1101/2020.04.04.20052092v2 (дата звернення 01.10.2021).

36. Hassanien A. Automatic X-ray COVID-19 lung image classification system based on multi-level thresholding and support vector machine [Електронний ресурс] / A. Hassanien. – 2020. – Режим доступу до ресурсу: https://www.medrxiv.org/content/10.1101/2020.03.30.20047787v1 (дата звернення 05.10.2021).

37. Kumar R. Accurate prediction of COVID-19 using chest X-Ray images through deep feature learning model with SMOTE and machine learning classifiers [Електронний ресурс] / Kumar. – 2020. – Режим доступу до ресурсу: https://www.medrxiv.org/content/10.1101/2020.04.13.20063461v1 (дата звернення 07.10.2021).

38. Li K. The clinical and chest CT features associated with severe and critical COVID-19 pneumonia [Електронний ресурс] / Li. – 2020. – Режим доступу до ресурсу: https://journals.lww.com/investigativeradiology/Fulltext/2020/06000/The_Clinical_and_Chest_CT_Features_Associated_With.1.aspx (дата звернення 08.10.2021).

39. Shi F. Large-scale screening of COVID-19 from community acquired pneumonia using infection size-aware classification [Електронний ресурс] / Shi. – 2020. – Режим доступу до ресурсу: https://arxiv.org/abs/2003.09860 (дата

звернення 29.10.2021).

40. Tang Z. Severity assessment of coronavirus disease 2019 (COVID-19) using quantitative features from chest CT images [Електронний ресурс] / Tang. – 2020. – Режим доступу до ресурсу: https://arxiv.org/abs/2003.11988 (дата звернення 22.10.2021).

41. Hernandez-Matamoros A. Forecasting of COVID19 per regions using ARIMA models and polynomial functions. [Електронний ресурс] / Hernandez-Matamoros. – 2020. – Режим доступу до ресурсу: https://doi.org/10.1016%2Fj.asoc.2020.106610 (дата звернення 23.10.2021).

42. Noor S. Analysis of public reactions to the novel coronavirus (COVID-19) outbreak on Twitter [Електронний ресурс] / Noor. – 2020. – Режим доступу до ресурсу: https://doi.org/10.1108/K-05-2020-0258 (дата звернення 30.10.2021).

43. Pathan R. Time series prediction of COVID19 by mutation rate analysis using recurrent neural network-based LSTM model [Електронний ресурс] / R. Pathan, M. Biswas, M. Khandaker. – 2020. – Режим доступу до ресурсу: https://doi.org/10.1016%2Fj.chaos.2020.110018 (дата звернення 18.10.2021).

44. Xing Y. MicroGMT: A mutation tracker for SARS-CoV-2 and other microbial genome sequences [Електронний ресурс] / Xing. – 2020. – Режим доступу до ресурсу: https://doi.org/10.3389%2Ffmicb.2020.01502 (дата звернення 17.10.2021).

45. Singer J. Cov-GLUE: A web application for tracking SARS-CoV-2 genomic variation [Електронний ресурс] / Singer. – 2020. – Режим доступу до ресурсу: https://www.preprints.org/manuscript/202006.0225/v1 (дата звернення 19.11.2021).

46. Korber B. Tracking changes in SARS-CoV-2 Spike: Evidence that D614G increases infectivity of the COVID-19 virus [Електронний ресурс] / Korber. – 2020. – Режим доступу до ресурсу: https://doi.org/10.1016/j.cell.2020.06.043 (дата звернення 19.10.2021).

47. Hazarika B. Modelling and forecasting of COVID-19 spreadusing wavelet-

coupled random vector functional link networks [Електронний ресурс] / B. Hazarika, D. Gupta. – 2020. – Режим доступу до ресурсу: https://doi.org/10.1016%2Fj.asoc.2020.106626 (дата звернення 12.11.2021).

48. Wynants L. Prediction models for diagnosis and prognosis of COVID-19: Systematic review and critical appraisal [Електронний ресурс] / Wynants. – 2020. – Режим доступу до ресурсу: https://doi.org/10.1136%2Fbmj.m1328 (дата звернення 11.10.2021).

49. Aggarwal C. Frequent Pattern Mining / C. Aggarwal, J. Han., 2014.

50. Gueniche T. Compact prediction tree: A lossless model for accurate sequence prediction / T. Gueniche, P. Fournier-Viger, V. Tseng // Proceedings of Advanced Data Mining and Applications / T. Gueniche, P. Fournier-Viger, V. Tseng., 2013. – C. 177–188.

51. Gueniche T. CPT+: Decreasing the time/space complexity of the compact prediction tree / Gueniche // Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining / Gueniche., 2015. – C. 625–636.

52. Padmanabhan V. Using predictive prefetching to improve world wide web latency [Електронний ресурс] / V. Padmanabhan, J. Mogul. – 1996. – Режим доступу до ресурсу: https://doi.org/10.1145%2F235160.235164 (дата звернення 12.11.2021).

53. Pitkow J. Mining longest repeating subsequence to predict world wide web surfing / J. Pitkow, P. Pirolli // Proceedings of USENIX Symposium on Internet Technologies and Systems / J. Pitkow, P. Pirolli., 1999. – C. 13–25.

54. Saul R. Discrete sequence prediction and its applications [Електронний ресурс] / R. Saul, P. Laird. – 1994. – Режим доступу до ресурсу: http://www.emis.de/MATH-item?0808.68086 (дата звернення 11.11.2021).

55. Lempel A. Compression of individual sequences via variable-rate coding [Електронний ресурс] / A. Lempel, J. Ziv. – 1978. – Режим доступу до ресурсу: https://doi.org/10.1109%2FTIT.1978.1055934 (дата звернення 21.11.2021).

56. Benson D. GenBank [Електронний ресурс] / D. Benson. – 2013. – Режим

доступу до ресурсу: https://doi.org/10.1093%2Fnar%2Fgks1195 (дата звернення 19.10.2021).

57. Shu J. A new integrated symmetrical table for genetic codes [Електронний ресурс] / J. Shu. – 2017. – Режим доступу до ресурсу: https://doi.org/10.1016%2Fj.biosystems.2016.11.004 (дата звернення 10.10.2021).

58. Fournier-Viger P. The SPMF open-source data mining library version 2 / Fournier-Viger // Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases / Fournier-Viger., 2016. – С. 36–40.

59. Ayres J. Sequential pattern mining using a bitmap representation / Ayres // Proceedings of Knowledge Discovery and Delivery / Ayres., 2002. – С. 429–435.

60. Fournier-Viger P. TKS: Efficient mining of top-k sequential patterns / Fournier-Viger // Proceedings of Advanced Data Mining and Applications / Fournier-Viger., 2013. – С. 109–120.

61. Fournier-Viger P. Fast vertical mining of sequential patterns using co-occurrence information / Fournier-Viger // Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining / Fournier-Viger., 2014. – С. 40–52.

62. Srikant R. Fast algorithms for mining association rules in large databases. / R. Srikant, R. Agrawal // Proceedings of Very Large Databases / R. Srikant, R. Agrawal., 1994. – С. 487–499.

63. Fournier-Viger P. ERMiner: Sequential rule mining using equivalence classes / Fournier-Viger // Proceedings of Intelligent Data Analytic / Fournier-Viger., 2014. – С. 108–119.

64. Karypis G. Selective markov models for predicting web page accesses [Електронний ресурс] / G. Karypis, M. Deshpande. – 2004. – Режим доступу до ресурсу: https://doi.org/10.1145%2F990301.990304 (дата звернення 20.11.2021).

65. Watson J. Molecular Biology of the Gene / J. Watson., 2014. – (7th edition).

66. Kupferschmidt K. The pandemic virus is slowly mutating [Електронний ресурс] / Kupferschmidt. – 2020. – Режим доступу до ресурсу: https://doi.org/10.1126%2Fscience.369.6501.238 (дата звернення 15.10.2021).

67. Day T. On the evolutionary epidemiology of SARS-CoV-2 [Електронний ресурс] / Day. – 2020. – Режим доступу до ресурсу: https://doi.org/10.1016%2Fj.cub.2020.06.031 (дата звернення 17.11.2021).

68. Sanjuan R. Viral mutation rates [Електронний ресурс] / Sanjuan. – 2010. – Режим доступу до ресурсу: https://doi.org/10.1128%2FJVI.00694-10 (дата звернення 23.11.2021).

69. Vignuzzi M. Quasispecies diversity determines pathogenesis through cooperative interactions in a viral population [Електронний ресурс] / Vignuzzi. – 2006. – Режим доступу до ресурсу: https://doi.org/10.1038%2Fnature04388 (дата звернення 25.11.2021).

70. Ramakrishnan S. A short review on antibody therapy for COVID-19 [Електронний ресурс] / S. Ramakrishnan, V. Jeyanthi, G. Kumar. – 2020. – Режим доступу до ресурсу: https://pubmed.ncbi.nlm.nih.gov/1468024/ (дата звернення 23.11.2021).

71. Apriori Algorithm [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: https://www.geeksforgeeks.org/apriori-algorithm/ (дата звернення 23.11.2021).

72. Pachetti M. Emerging SARS-CoV-2 mutation hot spots include a novel RNA-dependent-RNA polymerase variant [Електронний ресурс] / Pachetti. – 2020. – Режим доступу до ресурсу: https://doi.org/10.1186%2Fs12967-020-02344-6 (дата звернення 23.11.2021).

73. George T. How to analyze coronavirus mutation with Python [Електронний ресурс] / George. – 2020. – Режим доступу до ресурсу: https://www.towardsdatascience.com/tagged/python-mutation-analysis (дата звернення 23.11.2021).

74. Luna J. Supervised Descriptive Pattern Mining / J. Luna, S. Ventura., 2018.

75. Goodfellow I. Deep Learning / Goodfellow., 2016.

Sehn J. Insertions and deletions / J. Sehn., 2015.