УДК 004.9 **doi: 10.26906/SUNZ.2026.2.095**

В. Г. Знайдюк, В. Б. Тухтаров

Харківський національний університет радіоелектроніки, Харків, Україна

# АНСАМБЛЕВА МОДЕЛЬ ПРОГНОЗУВАННЯ ВІДМОВ ЗАВДАНЬ У ХМАРНИХ ОБЧИСЛЕННЯХ

**Анотація. Актуальність.** Хмарні обчислення є ключовим елементом сучасної IT-інфраструктури, однак проблема відмов завдань негативно впливає на якість обслуговування та ефективність використання ресурсів. Зростання складності хмарних систем та обсягів даних зумовлює необхідність застосування інтелектуальних методів прогнозування відмов, що дозволяють переходити від реактивних до проактивних і стійких підходів управління ресурсами. **Об'єкт дослідження:** процеси прогнозування відмов завдань у хмарних обчислювальних системах. **Мета статті:** розробка ансамблевої моделі прогнозування відмов завдань у хмарних обчисленнях на основі поєднання методів машинного навчання. **Результати дослідження.** У статті запропоновано ансамблеву модель, що базується на використанні методу стекінгу та поєднує алгоритми K-найближчих сусідів і штучної нейронної мережі з мета-моделлю на основі логістичної регресії. Проведено попередню обробку та аналіз даних набору Google Cluster Trace, виконано інженерію ознак і побудовано прогностичну модель. Експериментальні результати показали, що запропонована ансамблева модель забезпечує підвищення точності прогнозування та покращення показників F1-міри, прецизійності та повноти порівняно з окремими моделями. Встановлено, що використання ансамблевого підходу дозволяє зменшити ефект перенавчання та підвищити надійність прогнозів. **Висновки.** Запропонована модель є ефективним інструментом для прогнозування відмов завдань у хмарних системах та може бути використана для оптимізації планування ресурсів і підвищення відмовостійкості. Використання ансамблевого підходу сприяє зниженню витрат ресурсів і підтримує концепцію «зелених» обчислень. **Сфера використання отриманих результатів:** системи планування завдань, управління ресурсами та підвищення відмовостійкості у хмарних обчисленнях і центрах обробки даних.

**Ключові слова:** хмарні обчислення; прогнозування відмов; ансамблева модель; машинне навчання; штучна нейронна мережа; KNN; стекінг; логістична регресія; відмовостійкість; планування ресурсів.

## Вступ

**Постановка проблеми.** Хмарні технології стали критичним елементом у сучасній екостістемі та довели свою здатність сприяти зростанню різних секторів. Хмарні обчислення передбачають дослідження та вдосконалення алгоритмів для оптимізації ефективності різних аспектів, включаючи розподіл ресурсів, балансування навантаження та надійність. Крім того, вони спрямовані на покращення якості обслуговування шляхом скорочення середнього часу між відмовами системи. Наразі відбувається зсув у бік впровадження автоматизованого підходу, який має на меті мінімізувати випадки людських помилок та усунути надлишкові завдання. Автоматизація компонентів прийняття рішень у хмарі досягається за допомогою методів машинного навчання (МН) та глибокого навчання (ГН). Поява цих методів породила нову область досліджень, яка називається інтелектуальними хмарними обчисленнями. Це дозволяє зосередитися на вдосконаленні хмарної інфраструктури шляхом впровадження різноманітних інтелектуальних методів. Підвищення відмовостійкості є фундаментальним аспектом хмарних обчислень.

Існують різні категорії відмов у хмарі, які можуть створити каскадну подію відмов. Для пом'якшення цих відмов існуючі системи використовують різні заходи для забезпечення безперервності обслуговування в разі виникнення вузького місця. Впровадження управління відмовами в хмарному середовищі слугує для підвищення стійкості системи та створення відчуття надійності для клієнтів під час використання ними хмарних сервіс-провайдерів, тим самим забезпечуючи відмовостійкість. Існують три окремі класифікації методів відмовостійкості. Реактивний метод широко використовується хмарними провайдерами як основний підхід для швидкого виділення ресурсів у відповідь на збої в обслуговуванні. Використання цієї конкретної методології вважається придатним для досягнення відмовостійкості, незважаючи на те, що вона передбачає значну кількість додаткових ресурсів. Тривалість накладних витрат може становити від мінімум 5 секунд до максимум 15 секунд, що може порушити операційну діяльність бізнесу клієнта. Термін «накладні витрати» стосується часової вимоги для запуску нового ресурсу або перезавантаження сервісу.

Крім того, існують проблеми, пов'язані з повторюваними завданнями, зокрема процес визначення порогового значення. Ця процедура виконується вручну і тому несе ризик людської помилки. Спостерігається зростаюча тенденція до впровадження автоматизованих систем із самоусвідомленням, зокрема у сфері розуміння відмов, як засобу зменшення кількості повторюваних завдань. Завдяки використанню моніторингу журналів відмов стає можливим передбачити ймовірність виникнення відмов і згодом ініціювати процес розподілу ресурсів за допомогою вищезазначених реактивних методологій. Використання проактивних заходів з метою виявлення відоме як проактивні методи. Було розроблено різні методології, такі як самовідновлення, випереджувальна міграція, моніторинг, S-Guard та програмне оновлення. Були проведені обширі дослідження щодо явища самовідновлення та отримання випереджувальної міграції з використанням методологій машинного навчання.

Багато досліджень використовували різноманітні алгоритми МН для прогнозування найбільш відповідного порогового значення для випереджувальних міграцій, а також для моніторингу ресурсів за

 

допомогою сигналів «heartbeat». Займаючись прогнозуванням надійності, ми можемо робити обґрунтовані висновки щодо кількості порогових значень. Наразі існує новий підхід, яка поєднує реактивну та проактивну методології, зазвичай відома як стійкий підхід. Такий підхід є був вдосконалений для ефективного реагування на відмови, використовуючи наявний набір даних про відмови як основу.

Наше дослідження має на меті розробити модель, яка включає здатність прогнозувати відмови через передбачення відмови завдань у хмарних обчисленнях. Цього можна досягти шляхом визначення таких особливостей, як запитуваний ресурс і стан події завдання. Згодом модель може бути навчена розпізнавати шаблон відмови завдання за допомогою алгоритму МН. У цьому дослідженні буде використано три різні моделі для виконання завдання класифікації: K-найближчих сусідів (KNN), штучна нейронна мережа (ШНМ) та ансамблева модель, яка поєднує як KNN, так і ШНМ за допомогою методу стекінгу. Метод стекінгу об'єднує результати за допомогою логістичної регресії. У подальших застосунках цю модель можна використовувати для виявлення випадків відмови завдань під час планування завдань шляхом призначення пріоритету завдання та подальшого спрямування його в окрему чергу. Однією з переваг цього підходу є те, що виділення ресурсів для завдання не відбудеться, доки планувальник не обробить чергу. Цей підхід потенційно може зменшити виділення ресурсів, що призведе до зниження вартості хмарних послуг і сприятиме впровадженню практик «зелених» обчислень.

**Аналіз останніх досліджень і публікацій.** Передбачення відмов було центральним напрямком досліджень протягом багатьох років. Було використано різні моделі для покращення виявлення проблем у хмарних обчисленнях і пом'якшення відмов за допомогою проактивних та реактивних підходів. Основною метою цього проєкту є використання методів машинного навчання для виявлення випадків хмарних несправностей. Згідно з [1], різні проблеми, пов'язані з хмарними обчисленнями, можуть бути вирішені шляхом впровадження коригувальних стратегій, таких як контрольні точки, проактивна міграція, повторні спроби, перепланування завдань та програмне омолодження. Як реактивний, так і проактивний підходи виступають за впровадження всіх цих стратегій пом'якшення. В роботі було використано кілька моделей для прогнозування відмов, що охоплюють як програмні, так і апаратні збої. Ця стаття слугує інструментом для класифікації проблем і надання цінного аналізу щодо потенційних наслідків системних збоїв, зосереджуючи увагу на конкретних проблемах, таких як відмова додатків, стає можливим підвищити якість обслуговування (QoS).

В роботі [2] досліджено ефективне впровадження МН у комплексний та надійний спосіб. Застосування МН та штучного інтелекту продемонструвало свою ефективність у прогнозуванні хмарних збоїв. Було наведено набір даних і обговорено потенціал МН для покращення досліджень відмовостійкості. В наступному досліджені [3] ШНМ була використана для прогнозування потенційної відмови жорсткого диска на відповідному сервері. Було помічено, що комбінація ШНМ та технології самоконтролю, аналізу та звітування підвищує точність прогнозування відмови жорсткого диска. Відповідно модуль управління відмовостійкістю отримує можливість вживати проактивних заходів для запобігання виділенню віртуальних машин серверам, які демонструють потенційну вразливість до відмови. Це дослідження надає порівняльний аналіз алгоритмів ШНМ та KNN, висвітлюючи їх відповідну продуктивність з точки зору точності, яка, за повідомленнями, становить 90%. Практичне впровадження архітектури хмарних обчислень передбачає використання центрального контролера, який слугує основною сутністю, відповідальною за отримання запитів користувачів та їх подальший розподіл на фізичні машини. Друга функція стосується синхронізації кількох модулів, які відповідають за сприяння ефективному управлінню хмарною інфраструктурою. Архітектура реалізація, що пропонується, включає бази даних, які адмініструються модулями Hadoop і MapReduce, а також вторинний контролер, який контролює стан системи та надає сповіщення в разі будь-яких змін.

В роботі [4] представлено методологію, спрямовану на прогнозування відмови завдань із застосуванням різних підходів. Автор використав п'ять алгоритмів машинного навчання та оцінив їх відповідну продуктивність, оцінюючи їх точність.

Автор спробував вирішити проблему класифікації, використовуючи три різні категорії алгоритмів, а саме регресію, часові ряди та ансамбль. Алгоритм логістичної регресії є широко досліджуваним і визнаним методом регресії в статистичному аналізі. Алгоритм дерева рішень часто використовується як основний метод машинного навчання для задач класифікації, тоді як випадковий ліс класифікується як ансамблевий алгоритм. Автор також додав три окремі варіації моделей LSTM та ГН, кожна з яких відрізняється кількістю шарів. Модель ГН складається з трьох окремих підмоделей: одношарова довга короткочасна пам'ять з трьома шарами, двошарова LSTM з двома прихованими шарами та тришарова LSTM з трьома прихованими шарами. Для досягнення цієї мети алгоритм включає щільний шар для забезпечення формування єдиного значення для прогнозу. Крім того, процес навчання зупиняється, якщо не спостерігається покращення показника втрат на валідації після 10 епох. Висновок дослідження показав, що XGBoost продемонстрував вищу точність класифікації порівняно з іншими моделями, тоді як моделі випадкового лісу та дерева рішень виявилися більш придатними для прогнозування на рівні завдань.

В попередній роботі продемонстровано впровадження схеми планування для енергоощадної відмовостійкості. Ця схема використовує глибокі нейронні мережі для прогнозування відмов і планування завдань у межах репліки для виконання. Під час початкової фази завдання піддається тестуванню для оцінки ймовірності зіткнення з відмовою. Отже, воно класифікується як схильне до відмови або не схильне до відмови. Використання цього конкретного

підходу призводить до зниження енергоспоживання, що, своєю чергою, гарантує збереження якості обслуговування (QoS). Крім того, було запропоновано використовувати градієнтний спуск як метод зменшення похибки прогнозування в рамках аналізу відмов. Автор дослідив вплив різних ресурсів на виникнення відмови завдання.

В якості головного відкриття в роботі [5], пропонується використання векторного контейнера для перепланування суперзавдання на відповідному хості. Цей підхід використовує прогностичні методи для оптимізації алгоритму планування шляхом присвоєння точних числових значень параметрам алгоритму

Наступна робота [6] зосереджена на прогнозуванні відмов з використанням технології LSTM. Однак важливо зазначити, що LSTM не здатна ефективно обробляти кілька входів. Тому це дослідження надає всебічний розгляд двонапрямленої LSTM (Bi-LSTM), яка інтегрує більшу кількість вхідних характеристик.

Метою цього дослідження є визначення основних характеристик, які необхідно враховувати при розробці нашої перспективної моделі. Як навчання, так і тестування моделі будуть включати набір даних Google Cluster. Результати показують, що алгоритм генерує вихідні дані як у прямому, так і у зворотному напрямках для регулювання ваг вхідних ознак, які є близькими або далекими. Крім того, оцінка проводиться шляхом порівняння з іншими зразковими моделями з урахуванням їхньої точності, F1-міри, прецизійності та повноти. Тим не менш, точність прогнозування може знижуватися, коли часові інтервали перевищують певний поріг. Поточний результат є наслідком ретельної оцінки компромісу між величиною часового інтервалу та точністю прогнозу. Результати дослідження показують, що Bi-LSTM продемонструвала прогностичну точність 90%, коли мінімальний часовий інтервал було встановлено на рівні 15 хвилин і дотримано вимог щодо розміру.

Авторами [7] представлено фреймворк планування завдань, який інтегрує усвідомлення відмов, дозволяючи прогнозувати статус завершення завдання в реальному часі та вживати відповідних коригувальних заходів. Існування цієї характеристики призвело до того, що значна частина клієнтів переносить свої прикладні завдання на хмарні платформи. Фреймворк демонструє помітну здатність захищати близько 40% завдань, які, за прогнозами, зазнають відмови, шляхом ефективного виконання коригувальних заходів. Як наслідок, досягається економія ресурсів кластера, таких як центральні процесори та оперативна пам'ять. Крім того, проблема вибору дії формалізується в цьому дослідженні за допомогою моделі цілочисельного лінійного програмування. У сфері хмарних обчислень не рідкістю є відмови завдань в результаті різних факторів, включаючи, але не обмежуючись, дефекти програмного забезпечення, апаратні збої та неадекватний розподіл ресурсів. Наявність таких відмов може потенційно мати згубний вплив на QoS, що надається.

Дослідження [8] конкретно зосереджується на точці зору хмарних сервіс-провайдерів. На надійність хмарних застосунків можуть впливати різні фактори, включаючи характеристики завдань, конфігурації хмари та динамічні стани хмарної системи. В роботі використовується статистичний аналіз збоїв завдань, щоб виявити потенційні зв'язки між цими збоями та важливими обмеженнями планування, операціями вузлів та атрибутами користувачів у контексті хмарних обчислень. Дослідники пропонують низку потенційних стратегій для підвищення надійності хмарних додатків, як це пропонується їхніми емпіричними спостереженнями. Стратегії включають проактивне обслуговування вузлів та впровадження обмежень на частоту повторних надсилань завдань. Існує значний рівень зацікавленості в розумінні впливу планування завдань та обслуговування вузлів на виникнення збоїв завдань.

Впровадження складних алгоритмів МН призвело до помітного покращення в методології, причому спостерігається зростання тенденції використовувати ці методи для прогностичних цілей у сфері хмарних обчислень. Багатошарові персептрони (БП), різновид штучної нейронної мережі, продемонстрували значні перспективи в різних задачах прогнозування. Здатність точно відображати складні нелінійні зв'язки робить їх добре придатними для інтеграції в хмарні середовища, які за своєю суттю є динамічними та гетерогенними. В багатьох дослідженнях [9] підкреслюється важливість інженерії ознак для підвищення ефективності прогностичних моделей. У сфері хмарних систем включення конкретних характеристик, таких як системні журнали, показники використання ресурсів та історичні дані про збої завдань, відіграє життєво важливу роль у визначенні ефективності прогностичних моделей. В роботі [10] підкреслюється потенціал методологій МН, особливо нейронних мереж, таких як БП, для ефективного вирішення проблем, пов'язаних з прогнозуванням збоїв завдань у хмарних системах. Очікується, що зростаюча складність хмарних середовищ посилить важливість передових моделей машинного навчання.

В роботі [11] представлено аналіз комплексних трас робочого навантаження, зокрема тих, які були оприлюднені компанією Google. Вищезазначені дослідження показали, що значна частина часу в кластері була виділена на виконання завдань, які зрештою не досягли успішного завершення. Вищезазначені результати підкреслюють критичну необхідність глибокого розуміння механізмів та обґрунтувань, що лежать в основі завершення завдань у великомасштабних системах. Експоненціальне зростання обсягу даних у цих системах не супроводжувалося пропорційним покращенням надійності та безвідмовності. Питання надійності виходить за межі окремої системи. Проблема надійності завдань створює значні виклики як у традиційних системах високопродуктивних обчислень, які схильні до частих аварійних завершень додатків, так і в середовищах хмарних обчислень, які виконують різноманітні робочі навантаження на складних програмних стеках та гетерогенному обладнанні. В обох контекстах робочі навантаження демонстрували підвищену сприйнятливість до дефектів та помилок.

Основні напрями проаналізованих досліджень зосереджені на проактивній стратегії підвищення

відмовостійкості. Крім того, більшість робіт зосереджуються на традиційних моделях машинного навчання, хоча певні моделі, такі як Bi-LSTM, можуть включати модифікації своєї логіки. Однак важливо зазначити, що ці моделі схильні до перенавчання і можуть демонструвати субоптимальну продуктивність. Ще одним аспектом, який необхідно враховувати, є попередня обробка даних. Важливо визнати, що не всі дані можуть бути використані для розробки моделі прогнозування відмов. В зв'язку з цим необхідно ретельно вивчити призначення набору даних і внести відповідні зміни для вилучення релевантної інформації. Згодом набір даних повинен пройти попередню обробку, щоб полегшити розуміння моделлю та створити набір ознак, які підвищать точність моделі.

**Вищенаведене зумовило мету** даної роботи, а саме – розробку ансамблевої моделі для прогнозування відмовостійкості у хмарних обчисленнях. Запропоноване рішення поєднує в собі використання двох моделей KNN та ШНМ. А також використання методології стекінгу. Таким чином запропонована модель повинна значно підвищити точність прогнозу відмов в хмарному середовищі.

## Основний матеріал

У нашому дослідженні було взято до уваги набір даних Google trace. Набір даних містить інформацію, отриману з серверів Google Borg, охоплюючи загалом вісім окремих кластерів Borg. Система надає дані щодо використання ЦП, запитуваного використання ЦП та розподілу пам'яті для кожного завдання. Крім того, вона надає інформацію щодо зв'язку між кожним завданням і відповідним йому процесом, а також ієрархічних відносин між майстер- та робочими вузлами, що використовуються у фреймворку MapReduce [12]. Набір даних було використано для сприяння аналізу розподілу ресурсів під час запуску завдання та у разі відмови, тим самим допомагаючи в розумінні основного процесу. Стійкий підхід передбачає використання штучного інтелекту для розуміння розподілу ресурсів до фактичного процесу розподілу. Цей підхід підвищує надійність системи, впроваджуючи методології для прогнозування потенційної відмови процесу до її виникнення. Для інтеграції цієї концепції необхідно розробити модель, яка може ефективно прогнозувати відмову завдання.

Попередні дослідження вивчали різні моделі, які успішно досягли цієї мети. Тим не менш, ці моделі стикаються з проблемою перенавчання, яка зазвичай пов'язана з обмеженістю навчальних даних через міркування конфіденційності. Тому наше дослідження зосередилося на двох ключових факторах для прогнозування відмови завдань. По-перше, увага на обмежену доступність даних, що зумовило необхідність використання загальнодоступних наборів даних, які надають надійні поведінкові атрибути, пов'язані з обробкою завдань. По-друге, підхід, який ефективно вирішує проблему перенавчання. У цьому конкретному завданні було вирішено використати ансамблеву методологію. Цей підхід підкреслює комбінований ефект результатів, отриманих від кількох моделей, для вирішення внутрішніх невідповідностей, присутніх у результатах. Ансамблевий підхід можна класифікувати на три окремі категорії, а саме бустинг, беггінг та стекінг. В попередніх дослідженнях було розглянуто бустинг, однак необхідно також розглянути методологію стекінгу.

Завдяки використанню стекінгу може бути досягнуте більш повне візуальне представлення продуктивності нашої моделі, що полегшить необхідні коригування. На рис. 1 представлено метод стекінгу, який складається з двох окремих етапів. На першому рівні вибирається кілька моделей, кожна з яких генерує індивідуальні прогнози. Метою використання другого рівня є об'єднання прогнозу, яке часто називають мета-модель навчання.

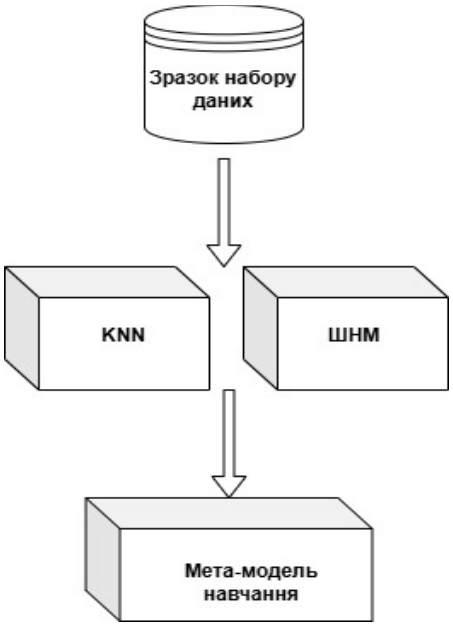


**Рис. 1**. Метод стекінгу

На початковому рівні нашої ансамблевої моделі відображено дві моделі: ШНМ та KNN. Модель ШНМ може бути представлено в різних варіантах, такі як нейронна мережа прямого поширення та рекурентна нейрона мережа. Ці варіанти відрізняються з точки зору напрямку, в якому дані зациклюються: мережа прямого поширення зациклює дані в одному напрямку, тоді як рекурентна мережа зациклює дані симетричним двох-направленим способом.

Хоча KNN вважається фундаментальною моделлю машинного навчання, здатною до розпізнавання патернів, важливо зазначити, що ця модель не покладається на жодні припущення щодо даних. Ця модель дозволяє класифікувати дані шляхом визначення їх близькості до заданої групи, тим самим відносячи їх до окремого кластеру. Для досягнення цілей в рамках цього дослідження обидві моделі були розглянуті, оскільки вони доповнюють одна одну з точки зору вирішення проблем «чорної скрині» та перенавчання. KNN також чутливий до нерелевантних ознак, тоді як ШНМ має сильну взаємозалежність ознак, оскільки вона створює зв'язки між ознаками для класифікації класів. Як уже зазначалося, ШНМ вважаються ефективними моделями для розпізнавання

патернів та класифікації завдань завдяки їхній здатності встановлювати зв'язки між вхідними параметрами. З іншого боку, KNN – це алгоритм класифікації, який не потребує жодних специфічних параметрів, але може демонструвати упереджений характер у своїх результатах класифікації. Отже, комбінуючи ці дві моделі, можна пом'якшити проблеми перенавчання та упередженості. Цього можна досягти шляхом навчання другого рівня ансамблевої моделі з використанням прогнозів, згенерованих KNN та ШНМ. В роботі було використано логістичну регресію на другому рівні для об'єднання результатів. Наступний етап передбачає аналіз даних, оскільки доступний набір даних має значний обсяг, що вимагає використання репрезентативної підмножини. Було проведено консолідацію полів даних, охоплюючи обмежену кількість інформації з кожної з восьми кластерів. Кожен кластер складається з чотирьох таблиць даних, в яких зберігається інформація, пов'язана з подіями машини, подіями колекції, подіями екземплярів та таблицею використання екземплярів. Набір даних, наданий для аналізу, складається приблизно з чотирьох мільйонів записів. Цей набір даних включає складове поле, яке охоплює різноманітну інформацію щодо запитуваної пам'яті ЦП для розподілу ресурсів, даних про використання ЦП, резервування спільних ресурсів для наборів розподілу та відносин між завданнями та їх батьківськими сутностями.

На рис. 2 представлена прогностична моделі, яка використовує ансамблевий підхід.

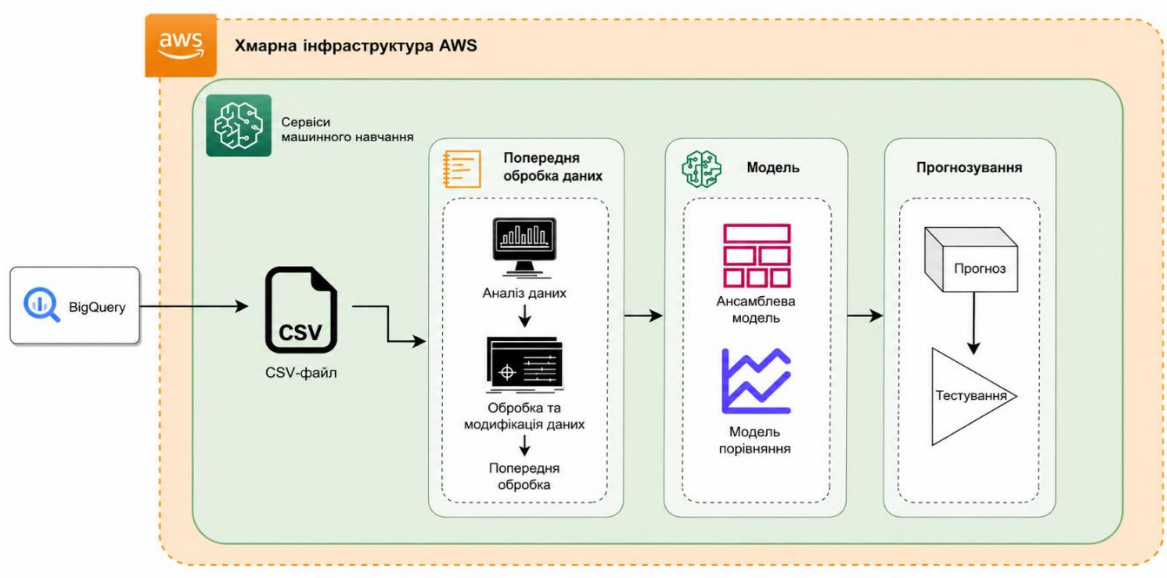


**Рис. 2.** Прогностична модель

Ансамблева модель є складною моделлю, яка потребує двох рівнів, що робить використання хмарних обчислень найбільш оптимальним підходом для досягнення швидших і точніших результатів. Спочатку необхідно встановити набір даних та провести процедури попередньої обробки. Дані демонструють часовий вимір і значну кореляцію, причому негативна кореляція позитивно пов'язана з відмовою завдання і слугує причинним фактором відмови завдання.

Під час етапу попередньої обробки даних певні поля, які вважаються непотрібними, імпутуються, а категоріальні та порядкові значення перетворюються на двійкові представлення. Це перетворення полегшує розуміння та виявлення патернів у даних. Під час етапу обробки створено проєкт, який дозволяє вибрати відповідні ознаки з попередньо оброблених даних після ретельного аналізу. Далі переходимо до вибору моделей, які потенційно можуть підвищити точність наших прогнозів. Враховуючи фактори перенавчання та явища «чорної скрині», було обрано три моделі. Перший рівень буде навчатися за допомогою двох моделей, тоді як другий рівень буде використано для об'єднання результатів двох моделей шляхом навчання іншої моделі для розуміння патерну. Дві моделі, KNN та ШНМ, складаються поверх моделі логістичної регресії, як показано на рисунку 1, і використовуючи цей підхід, досягаємо вищої точності прогнозування. У майбутньому цю модель можна запакувати та застосувати до будь-якого API, який можна додатково під'єднати до сервісу черги, що може діяти як мікросервіс і надавати ймовірність відмови завдання назад алгоритму планування завдань для прогнозування розподілу ресурсів.

Модель виконується з використанням хмарного сервісу AWS, який надає можливість виконувати складні моделі без необхідності турбуватися про придбання ресурсів. SageMaker надає доступ до обчислювально-оптимізованих екземплярів. Крім того, підвищення ефективності нашого експерименту, було за рахунок виброру підмножини даних, яку можна зібрати протягом обмеженого періоду часу. Крім того, було використано ядро PyTorch, яке належить до контейнерів глибокого навчання. Ядро охоплює всеосяжну колекцію основних фреймворків та бібліотек, які зручно зберігаються в образі Docker.

Для реалізації моделі відповідно до наших визначених критеріїв використовували Python для створення блокнота експерименту в AWS. Реалізація була розділена на чотири етапи, а саме: попередня обробка

даних, інженерія ознак, прогнозування та порівняння моделей.

Дані, отримані з Google Cluster Trace Version 3 2019, були перетворені на вибірку CSV. Дані піддалися аналізу для виявлення як нульових значень, так і числових значень, що дозволяє визначити поля, які містять нульові значення. Далі визначали всі поля, класифіковані як числові, а також ті поля, які демонструють категоріальні та порядкові характеристики. Після цього проведено оцінювання, чи містить наша навчальна ознака збалансований розподіл даних. Було помічено, що набір даних демонструє дисбаланс, що спонукало до необхідності використання методів вибірки даних. Процес вибірки даних дозволяє навчати модель, використовуючи однакову кількість істинно позитивних і хибно позитивних результатів. Використання цього підходу забезпечить підвищену точність прогнозування та розуміння показників ефективності, включаючи F1-міру, повноту та прецизійність. Наступний етап аналізу полягає у визначенні відповідних ознак для включення, чому сприяє візуалізація даних за допомогою графічних представлень, таких як коробчасті діаграми або гістограми.

Цей етап в основному підкреслює модифікацію ознак і вибір тих, які є найбільш придатними для ефективного проектування даних. Після проведення аналізу даних основна увага спрямовується на розробку ознак і трансформацію даних з метою підвищення продуктивності алгоритму машинного навчання. При спостереженні за даними стає очевидним, що запитуваний ресурс зберігається у форматі JSON. Отже, необхідно перетворити дані на кілька колонок, щоб встановити зв'язок для моделі. Ця функціональність згодом використовується для усунення кореляції, щоб визначити, чи є поле придатним для підвищення прогностичної продуктивності моделі. Під час процесу спостереження за даними необхідно масштабувати числові поля, щоб узгодити їхні величини з величинами інших полів. Необхідно виключити поле результату «fail» («відмова»), щоб ефективно навчати модель. Для нормалізації значень використовується стандартний скейлер (Standard Scaler). Для того щоб комп’ютер міг інтерпретувати кардинальні та порядкові значення як вхідні дані, їх необхідно перетворити на двійкові представлення; це досягається за допомогою функції кодувальника міток. Під час цього процесу перетворення стає очевидним, які значення є найбільш сумісними з моделлю. Після того як дані перетворено, приступимо до прогнозування значення. Цей процес передбачає розділення набору даних на дві підмножини: 80% виділяється для навчання, а решта 20% зберігається для тестування.

Реалізація прогнозування моделі на цьому етапі використовує бібліотеку Sklearn. Ініціалізація моделі передбачає виклик конструкторів відповідних моделей. Для проведення порівняльного аналізу необхідно ініціалізувати моделі KNN, ШНМ та логістичної регресії як перші кроки в процесі прогнозування.

Цей крок передбачає навчання моделі шляхом її підгонки на 80% наявних даних. У цій діяльності були використані всі три підходи. Наступний етап після підгонки моделі полягає в оцінці навченої моделі шляхом її тестування на підмножині, що становить 20% наявних даних. Результат цієї функції буде збережено у змінній, яка містить префікс test. Функція predict (прогнозувати) відповідає за проведення валідації нашої навчальної моделі та створення матриці згортки. Це дозволяє дослідити показники ефективності моделей. Далі використовується бібліотека Sklearn metrics для виклику статичної функції під назвою «classification report» з метою створення порівняльного звіту між підігнаним та протестованим значеннями. Кроки, пов'язані з реалізацією моделі ШНМ та KNN, виконуються відповідно. Однак при роботі з ансамблевою моделлю необхідно ініціалізувати об'єкт з параметрами оцінювача та кінцевого оцінювача, який також називається шаром мета-модель навчання. У процесі ініціалізації ансамблевої моделі необхідно спочатку ініціалізувати об'єкти моделей ШНМ та KNN. Ці ініціалізовані об'єкти моделей зберегаться в масиві та згодом передаються як аргументи параметру estimator об'єкта Stacking classifier. Крім того, об'єкт ініціалізованої логістичної регресії передається параметру final estimator об'єкта Stacking classifier. Після ініціалізації маємо дотримуватися вищезазначеного процесу підгонки, прогнозування та порівняння.

Заключним етапом є порівняння матриці ефективності кожної моделі. Ця матриця дає розуміння того, як модель працювала, аналізуючи F1-міру, повноту, прецизійність, точність, $RMSE$ та $R^2$.

Прогностична модель враховувала невелику кількість значущих ознак, які були проаналізовані під час етапу попередньої обробки даних. Модель демонструє підвищену продуктивність, коли присутня негативна кореляція, що дозволяє виявляти обмежену кількість завдань, які можуть завершитися невдачею, на основі попередніх запитів. Дані дають нам розуміння успішних та неуспішних завдань у всіх восьми кластерах. Кластер виділяє екземпляри для виконання завдання, і зі збільшенням кількості екземплярів обробляється більша частина завдання. На рис. 3, відображено, що кластери 3, 4 та 6 мають найбільшу кількість екземплярів під час виконання завдання, таким чином даючи уявлення про те, що кількість виконаних завдань більша в цих трьох кластерах.

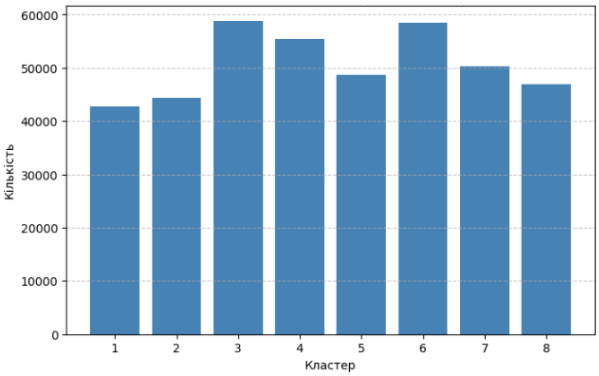


**Рис. 3.** Розподіл екземплярів за кластерами

На рис. 4 представлено середнє призначення пам’яті по кластерам. Розуміння розподілу пам'яті в

усіх кластерах обмежується базовим рівнем. Це спостереження передбачає, що, незважаючи на відносно низьку частоту виникнення подій у кластері 8, значний обсяг пам'яті виділяється ресурсу.

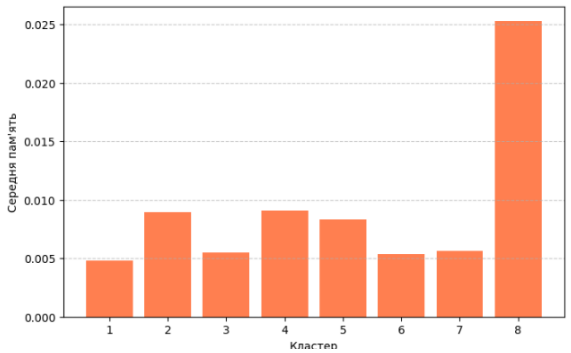


**Рис. 4.** Середнє призначення пам’яті по кластерах

Це відхилення від норми в поведінці може вказувати на те, що завдання потребує додаткової пам'яті для свого опрацювання, або що завдання зазнає невдач, що призводить до повторних спроб і, згодом, до більшого споживання пам'яті порівняно з іншими кластерами. Дослідження розподілу пам'яті є ключовим фактором у нашому передбаченні відмови завдання. Існує обернена кореляція між виділенням пам'яті та виникненням аномальної поведінки в кластері. При ретельному вивченні набору даних стає очевидним, що більший обсяг пам'яті зазвичай виділяється термінованим ресурсам.

Навпаки, більшість невдалих завдань демонструють вищий рівень споживання введення/виведення (I/O), що вказує на переважання збоїв запису на диск. Крім того, можна зробити висновок, що основною причиною відмов завдань у кластері є відмова ресурсів. Це спостереження підкреслює важливість включення використання та розподілу ресурсів як ключових факторів у розробці нашої прогностичної моделі.

Ці результати мали важливе значення в процесі вибору ознак. Було визначино наявність запитуваних ресурсів, які надають інформацію про необхідну пам'ять для завдання, а також тривалість кожного завдання. Ці спостереження реєструвалися з інтервалом у 5 хвилин, що дозволяє аналізувати час, необхідний для завершення завдання, та подальший час, необхідний для повторних спроб завдання. Вищезазначений фактор відіграє ключову роль у розумінні виникнення відмови завдання та сприятиме передбаченню відмови завдання. Було проведено порівняльне дослідження між кількома моделями, щоб показати найкращу модель для прогнозування відмови завдань.

Експериментальні дослідження також надають розуміння того, яка модель є найкращою для такого типу даних. Оскільки зосереджуємося на класифікації завдання між успіхом і невдачею, під час експеременті використали результати двох ансамблевих моделей і двох моделей машинного навчання, щоб забезпечити порівняльне дослідження між обома підходами. Також було враховано такі фактори, як F1-міра, точність та повнота, щоб глибше зрозуміти, яка модель найкраще підходить для таких прогнозів. Розділили дані на 80% навчальних даних і 20% тестових даних для валідації прогнозу. Під час експериментів вимірюємо продуктивність моделі. Давайте визначимо матрицю продуктивності нижче:

На рис. 5 представлено ROC криві всіх 3 моделей.

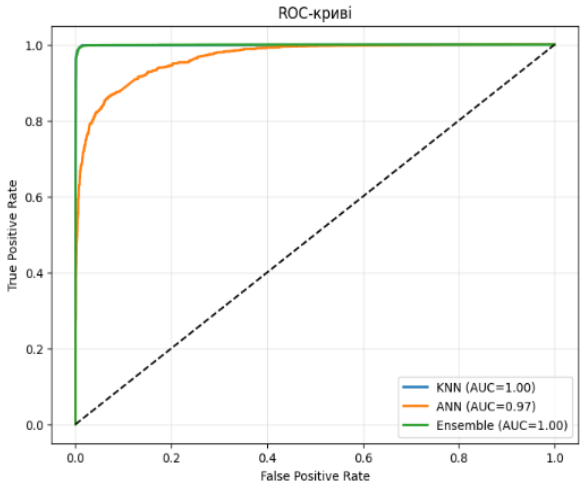


**Рис. 5.** ROC криві моделей
(KNN, ШНМ, ансамблева модель)

На рис. 6-8 представлені матриці помилок для наших трьох моделей KNN, ШНМ, ансамблева модель.

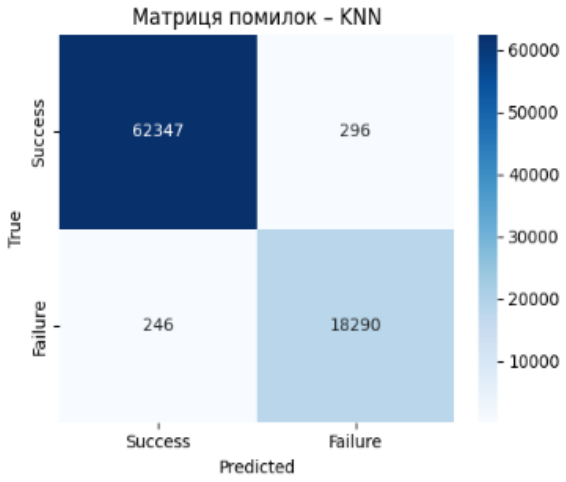


**Рис. 6.** Матриця помилок KNN

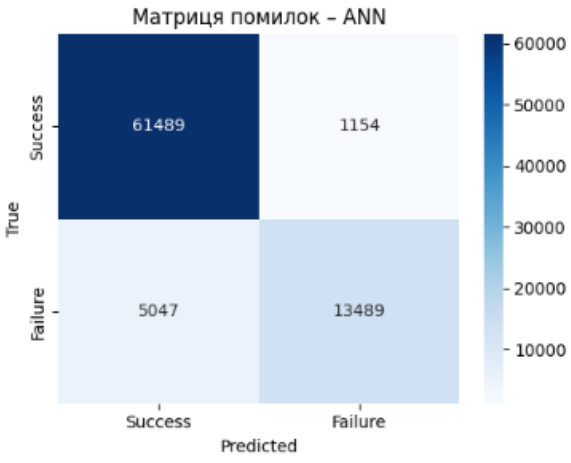


**Рис.7.** Матриця помилок ШНМ

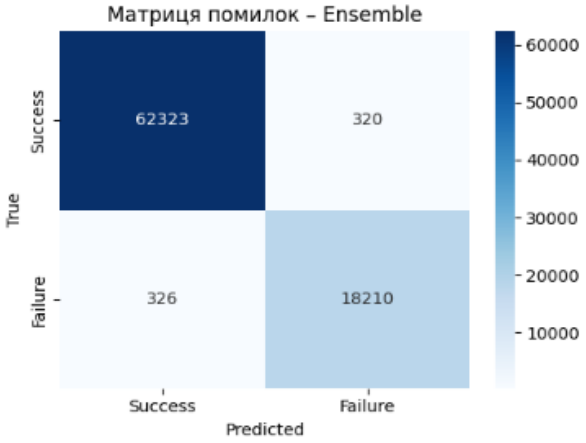


**Рис. 8.** Матриця помилок ансамблева модель

Ансамблева модель, використана під час експериментів, використовує підхід стекінгу для вирішення обмежень як KNN, так і ШНМ. Складність цієї моделі робить її вигідною порівняно з альтернативними моделями. Однак складність моделі вимагає значних витрат часу та обчислювальних ресурсів як для навчання, так і для тестування. Ще один аспект, який заслуговує на увагу при спостереженні, це використання складних моделей, таких як ШНМ, які складаються з кількох шарів.

В табл. 1 представлено порівняльні результати експериментів з різними моделями.

Аналіз продуктивності кожної моделі дозволяє здійснити всебічну оцінку різних факторів, що діють у кожній моделі. Для використання будь-якої з моделей необхідно мати глибоке розуміння даних.

*Таблиця 1* – **Порівняльна таблиця**

| | **Accuracy** | **Precision** | **Recall** | **F1-** | **RMSE** | $R^2$ |
|---|---|---|---|---|---|---|
| KNN | 0.9913 | 0.9841 | 0.9867 | 0.9854 | 0.0706 | 0.9717 |
| ШНМ | 0.9236 | 0.9212 | 0.7277 | 0.8131 | 0.2426 | 0.6660 |
| Ансамбль | 0.9920 | 0.9927 | 0.9924 | 0.9926 | 0.0789 | 0.9746 |

Продуктивність моделі визначається даними, і процес вибору ознак суттєво впливає на прогнозну здатність моделі. Використовуючи показники продуктивності, отримали подальшу інформацію про продуктивність кожної моделі. Можна помітити досить високу точність, F1-міру та повноту. Це означає, що модель є дійсно оптимальною для класифікації успіху чи невдачі завдання. Модель ШНМ схильна до перенавчання через її надмірно високу точність, що вказує на те, що модель має здатність запам'ятовувати патерни, а не дійсно навчатися їм. Алгоритм KNN також схильний до тієї самої проблеми. Однак ансамблева модель включає шар мета-навчання для агрегування результатів, що значно знижує ймовірність перенавчання. Крім того, проведено навчання моделі, яка налаштована на Google Cloud Trace та оптимізована для покращення продуктивності кластерів Google. Ансамблева модель демонструє надійні результати, однак вона має вищий рівень складності через об'єднання двох алгоритмів машинного навчання. Крім того, було додано нейронну мережу, що ще більше сприяє складному характеру ансамблевої моделі, що призводить до збільшення потреби в ресурсах порівняно з альтернативними моделями. Окрім цих проблем, ансамблева модель страждає від проблем конфіденційності через відсутність прозорості.

## Висновки

На основі вищезазначених експериментів можна зробити висновок, що використання ансамблевої моделі дає надійну модель прогнозування з покращенням точності від 1 до 15 відсотків порівняно з ШНМ та KNN відповідно. Тим не менш, важливо визнати, що модель має певні обмеження. Очікувалось, що ансамблева модель буде демонструвати кращу продуктивність при спільному використанні двох моделей машинного навчання, ніж одна модель глибокого навчання. Альтернативно, можна використовувати модель глибокого навчання, таку як ШНМ або алгоритм ГН, шляхом зменшення кількості ознак та нормалізації даних. Цей підхід має на меті підвищити продуктивність моделі. Навчання двох моделей в ансамблевій моделі часто потребує значного обсягу пам'яті, що вказує на те, що ансамблева модель залежить від обчислювально-інтенсивних ресурсів. Це дає розуміння, що вибір ансамблевої моделі, повинен здійснюватися за умови послідовного використання пакетних завдань як основного способу виконання завдань у нашому робочому процесі. Запропонована реалізація сприяє розробці алгоритмів планування завдань у центрах обробки даних.

На основі цього прогнозу планувальник завдань виділяв би ресурси відповідно до пріоритету, визначеного зберіганням завдання в черзі, за умови, що прогнозоване значення потрапляє у вказаний діапазон. Створення такого застосунку є надзвичайно корисним, оскільки його інтеграція в алгоритм планування завдань надає можливість покращити алгоритм за допомогою методів машинного навчання. Це може сприяти подальшим дослідженням у сфері «зелених» обчислень, зокрема зосереджуючись на використанні стійкого підходу для забезпечення відмовостійкості.

## Конфлікт інтересів

Автори декларують, що не мають конфлікту інтересів стосовно даного дослідження, в тому числі фінансового, особистісного характеру, авторства чи іншого характеру, що міг би вплинути на дослідження та його результати, представлені в даній статті.

## Використання засобів штучного інтелекту

Автори підтверджують, що не використовували технології штучного інтелекту при створенні представленої роботи.

СПИСОК ЛІТЕРАТУРИ

1. Shahid M. A., Islam N., Alam M. M., Mazliham M., Musa S. Towards resilient method: An exhaustive survey of fault tolerance methods in the cloud computing environment // Computer Science Review. 2021. Vol. 40. Art. 100398. DOI: https://doi.org/10.1016/j.cosrev.2021.100398
2. Agarwal K. K., Kotakula H. Fault tolerance in cloud: A brief survey // Advances in Communication, Cloud, and Big Data. Springer, 2022. P. 578–589. DOI: https://doi.org/10.1007/978-981-19-2829-6_55
3. Ragmani A., Elomri A., Abghour N., Moussaid K., Rida M., Badidi E. Adaptive fault-tolerant model for improving cloud computing performance using artificial neural network // Procedia Computer Science. 2020. Vol. 170. P. 990–997. DOI: https://doi.org/10.1016/j.procs.2020.03.049
4. Tengku Asmawi T. N., Ismail A., Shen J. Cloud failure prediction based on traditional machine learning and deep learning // Journal of Cloud Computing. 2022. Vol. 11. DOI: https://doi.org/10.1186/s13677-022-00324-9
5. Marahatta A., Xin Q., Chi C., Zhang F., Liu Z. PEFS: AI-driven prediction-based energy-aware fault-tolerant scheduling scheme for cloud data center // IEEE Transactions on Sustainable Computing. 2021. Vol. 6, No. 4. P. 655–666. DOI: https://doi.org/10.1109/TSUSC.2020.2964266
6. Gao J., Wang H., Shen H. Task failure prediction in cloud data centers using deep learning // IEEE Transactions on Services Computing. 2022. Vol. 15, No. 3. P. 1411–1422. DOI: https://doi.org/10.1109/TSC.2020.2964407
7. Alahmad Y., Daradkeh T., Agarwal A. Proactive failure-aware task scheduling framework for cloud computing // IEEE Access. 2021. Vol. 9. P. 106152–106168. DOI: https://doi.org/10.1109/ACCESS.2021.3100708
8. Jassas M. S., Mahmoud Q. H. A failure prediction model for large-scale cloud applications using deep learning // Proceedings of the IEEE International Systems Conference (SysCon). 2021. DOI: https://doi.org/10.1109/SysCon48628.2021.9447071
9. Vani K., Sujatha S. A machine learning framework for job failure prediction in cloud using hyper-parameter tuned MLP // Proceedings of the 2nd International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE). 2022. DOI: https://doi.org/10.1109/ICATIECE54045.2022.9768518
10. Ляшенко О., Михайліченко І. Модель самоадаптивної розподіленої системи керування ресурсами у хмарних обчисленнях // Herald of Khmelnytskyi National University. Technical Sciences. 2026. № 2 (363). С. 335–343. DOI: https://doi.org/10.31891/2307-5732-2026-363-46
11. El-Sayed N., Zhu H., Schroeder B. Learning from failure across multiple clusters: A trace-driven approach to understanding, predicting, and mitigating job terminations // Proceedings of the IEEE 37th International Conference on Distributed Computing Systems (ICDCS). 2017. P. 1333–1344. DOI: https://doi.org/10.1109/ICDCS.2017.155
12. Wilkes J. Google cluster-usage traces V3 [Електронний ресурс]. URL: https://github.com/google/cluster-data

Received (Надійшла) 12.01.2026
Accepted for publication (Прийнята до друку) 15.04.2026
Publication date (Дата публікації) 22.05.2026

ВІДОМОСТІ ПРО АВТОРІВ / ABOUT THE AUTHORS

**Знайдюк Василь Григорович** – кандидат технічних наук, доцент кафедри електронних обчислювальних машин, Харківський національний університет радіоелектроніки, Харків, Україна;
**Vasyl Znaidiuk** - PhD, Associate Professor of the Department of Electronic Computers, Kharkiv National University of Radio Electronics,, Kharkiv, Ukraine.
e-mail: vasyl.znaidiuk@nure.ua; ORCID Author ID: https://orcid.org/0000-0001-8590-8007;
Scopus Author ID https://www.scopus.com/authid/detail.uri?authorId=57210340749.

**Тухтаров Владислав Борисович –** аспірант кафедри електронних обчислювальних машин, Харківський національний університет радіоелектроніки, Харків, Україна;
**Vladyslav Tukhtarov** – Postgraduate student of the Department of Electronic Computers, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine;
e-mail: vladyslav.tukhtarov@nure.ua; ORCID Author ID: https://orcid.org/0009-0005-7650-965X.

**Ensemble model for task failure prediction in cloud computing**

Vasyl Znaidiuk, Vladyslav Tukhtarov

**Abstract. Relevance.** Cloud computing is a key component of modern IT infrastructure; however, task failures negatively affect service quality and resource utilization efficiency. The increasing complexity of cloud systems and the growing volume of data necessitate the use of intelligent failure prediction methods, enabling the transition from reactive to proactive and resilient resource management approaches. **Object of research:** processes of task failure prediction in cloud computing systems. **Purpose of the article:** development of an ensemble model for task failure prediction in cloud computing based on a combination of machine learning methods. **Research results.** The paper proposes an ensemble model based on the stacking method, combining the K-nearest neighbors algorithm and an artificial neural network with a meta-model based on logistic regression. Data preprocessing and analysis of the Google Cluster Trace dataset were performed, feature engineering was conducted, and a predictive model was developed. Experimental results demonstrated that the proposed ensemble model improves prediction accuracy and enhances F1-score, precision, and recall compared to individual models. It was established that the ensemble approach reduces overfitting and increases prediction reliability. **Conclusions.** The proposed model is an effective tool for task failure prediction in cloud systems and can be applied to optimize resource scheduling and improve fault tolerance. The use of the ensemble approach contributes to reducing resource consumption and supports the concept of green computing. **Scope of application:** task scheduling systems, resource management, and fault tolerance enhancement in cloud computing and data centers.

**Keywords:** cloud computing; failure prediction; ensemble model; machine learning; artificial neural network; KNN; stacking; logistic regression; fault tolerance; resource scheduling.