

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРИВОРІЗЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ
Фізико-математичний факультет
Кафедра інформатики та прикладної математики

«Допущено до захисту»

Завідувач кафедри

_____ Соловйов В. М.

Реєстраційний № _____

«__» _____ 2020 р.

«__» _____ 2020 р.

ДОСЛІДЖЕННЯ МОЖЛИВОСТЕЙ
КВАНТОВОГО ПРОГРАМУВАННЯ
ДЛЯ РЕАЛІЗАЦІЇ ЗАДАЧ МАШИННОГО НАВЧАННЯ

Кваліфікаційна робота студента групи І-16
ступінь вищої освіти «бакалавр»
спеціальності 014.09 Середня освіта (інформатика)
Загородька Павла Володимировича

Керівник доктор педагогічних наук, професор
Семеріков Сергій Олексійович

Оцінка:

Національна шкала _____

Шкала ECTS _____ Кількість балів _____

Голова ЕК _____

(підпис)

(прізвище, ініціали)

Члени ЕК _____

(підпис)

(прізвище, ініціали)

(підпис)

(прізвище, ініціали)

(підпис)

(прізвище, ініціали)

(підпис)

(прізвище, ініціали)

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1 ОСНОВИ ІНЖЕНЕРІЇ КВАНТОВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	5
1.1 Основні поняття дослідження.....	5
1.2 Поняття про інженерію квантового програмного забезпечення.....	5
1.3 Огляд засобів інженерії квантового програмного забезпечення	9
Висновки до розділу 1	16
РОЗДІЛ 2 КВАНТОВО ПОКРАЩЕНЕ МАШИННЕ НАВЧАННЯ	17
2.1 Квантові моделі машинного навчання.....	17
2.2 Огляд засобів квантово покращеного машинного навчання у Qiskit..	19
Висновки до розділу 2	21
РОЗДІЛ 3 ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА ЕФЕКТИВНОСТІ КВАНТОВО ПОКРАЩЕНИХ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ	22
Висновки до розділу 3	24
ВИСНОВКИ.....	25
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	27
ДОДАТКИ.....	39
А Код для квантово покращеного машинного навчання на наборі даних wine (сервер ibmq_burlington, 5 кубітів).....	39
Б Код для квантово покращеного машинного навчання на наборі даних breast_cancer (локальний квантовий емулятор, 5 кубіти)	40
В Код традиційного машинного навчання на наборі даних breast_cancer	41

ВСТУП

Актуальність теми. Традиційно квантові обчислення визначаються як тип неklasичних обчислень, що використовує квантові стани субатомних частинок, які представляють дані у квантових бітах (кубітах). До зчитування даних з кубіту він зберігає усі можливі значення одночасно в стані суперпозиції. Кубіти можуть бути пов'язані один з одним властивістю заплутаності. Квантові алгоритми маніпулюють пов'язаними кубітами в їх невизначеному (заплутаному стані), потенційно надаючи можливість розв'язувати задачі з величезною комбінаторною складністю [83], досягнувши «квантового панування».

К. Панетта (Kasey Panetta), визначаючи потенційні застосування квантових обчислень, вказує, що вони «будуть вузькими та цілеспрямованими, оскільки квантові обчислення загального призначення, швидше за все, ніколи не будуть економічними» [65]. На його думку, квантові обчислення можуть забезпечити прориви у машинному навчанні, фінансах, медицині, створенні нових матеріалів, штучному інтелекті (для чого необхідні сотні та тисячі кубітів), хімії та біохімії (100-200 кубітів). Зокрема, для фінансів квантові обчислення можуть надати можливість створення більш швидших та складних моделей з використанням методу Монте-Карло (для торгівлі, оптимізації траєкторій, прогнозування нестабільності ринку, стратегій оптимізації цін та хеджування) та методів машинного навчання, які у загальному випадку зводяться до задач пошуку екстремуму багатовимірної функції з нелінійною поверхнею відгуку.

На поточний момент обчислювальні пристрої, здатні виконувати квантові обчислення (квантові комп'ютери), для споживачі обчислювальних послуг доступні за моделлю QaaS (quantum computing as a service – квантові обчислення як послуга). Станом на червень 2020 року максимальна кількість кубітів, доступних для одночасного використання, не перевищує 60, що суттєво менше за кількість, необхідну для досягнення «квантового панування». Це

породжує проблему дослідження можливостей квантового програмування для реалізації задач машинного навчання, а саме – застосування алгоритмів машинного навчання, реалізованих мовою квантового програмування, для аналізу традиційних даних, та порівняння продуктивності квантових і фон-немайновських реалізацій на сучасному етапі їх розвитку.

Мета дослідження – виконати порівняльний аналіз реалізації алгоритмів квантово покращеного машинного навчання, реалізованих мовою квантового програмування.

Для досягнення мети дослідження були поставлені такі **завдання**:

1. Проаналізувати засоби інженерії квантового програмного забезпечення з метою вибору засобу, доцільного для реалізації задач машинного навчання.
2. Розглянути квантово покращені алгоритми машинного навчання.
3. Виконати порівняльний аналіз ефективності квантово покращених і традиційних алгоритмів машинного навчання.

Об’єкт дослідження – інженерія квантового програмного забезпечення.

Предмет дослідження – алгоритми машинного навчання, реалізовані мовами квантового програмування.

Методи дослідження: *аналіз* джерел та програмного забезпечення з метою визначення стану розв’язання проблеми дослідження та добору засобів реалізації алгоритмів квантово покращеного машинного навчання, *методи інженерії квантового програмного забезпечення* (проектування, розробка, тестування) для досягнення мети дослідження.

Практичне значення одержаних результатів полягає у порівняльному аналізі ефективності квантово покращених і традиційних алгоритмів машинного навчання на платформі IBM Q Experience.

Структура роботи. Робота складається зі вступу, трьох розділів, висновків, списку використаних джерел (116 найменувань) та трьох додатків.

РОЗДІЛ 1

ОСНОВИ ІНЖЕНЕРІЇ КВАНТОВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Основні поняття дослідження

Квантовий комп'ютер – обчислювальний пристрій, що використовує явища квантової механіки (квантову суперпозицію, квантову сплутаність та ін.) для передавання та опрацювання даних.

Квантове програмування – процес розробки програм для квантового комп'ютера.

«Класичні» застосування квантових комп'ютерів (за Р. Фейнманом) – моделювання складних [багаточасткових фізичних] систем: алгоритм Залкі-Візнера.

«Нові» застосування квантових комп'ютерів – задачі, що потребують перебору великої кількості варіантів: алгоритми Гровера (загальна задача), Шора (факторизація), Абрамса-Ллойда (виявлення періодичних властивостей) та ін.

Квантове машинне навчання – застосування алгоритмів машинного навчання для аналізу квантових даних.

Квантово покращене машинне навчання – застосування алгоритмів машинного навчання, реалізованих мовою квантового програмування, для аналізу традиційних даних.

Інженерія програмного забезпечення – системне застосування інженерних підходів до проектування, реалізації, тестування та документування програмного забезпечення.

1.2 Поняття про інженерію квантового програмного забезпечення

Перший системний виклад концепції інженерії квантового програмного забезпечення був зроблений Дж. Кларком (John Clark) та С. Степні (Susan Stepney) у 2002 році [14]. Дослідники виходять з того, що квантові обчислення

не можуть бути ефективно реалізовані у традиційній комп'ютерній архітектурі Дж. фон Неймана, математичною моделлю якої є машина Тюрінга. До основних викликів, що постануть перед інженерією квантового програмного забезпечення у 2020 році, автори [14] відносять:

- питання, якій бути мові квантового програмування – розширенням традиційних мов, мовою логічного програмування, мовою низькорівневого програмування або мовою, що реалізує нову парадигму;
- потребу у розробці компіляторів для квантових мов програмування;
- необхідність розробки нових квантових алгоритмів та визначення класів традиційних алгоритмів, що можуть бути квантизованими;
- доцільність розробки квантових симуляторів для використання на традиційних комп'ютерних системах;
- попри принципову неспостережуваність роботи квантового комп'ютера, для квантових мов програмування необхідні засоби налагодження та тестування;
- квантові алгоритми потребують візуалізації для їх розуміння, вдосконалення та реалізації.

Критерії та показники успішності інженерії квантового програмного забезпечення, запропоновані Дж. Кларком та С. Степні, узагальнені в таблиці 1.1.

Таблиця 1.1

Критерії та показники успішності інженерії квантового програмного забезпечення (за [14])

Критерій	Показник
ґрунтується на науковому інтересі щодо основ, природи та меж наукової дисципліни	квантові обчислення розширили фундаментальні межі комп'ютерних наук та інженерії програмного забезпечення
можливість створення нових інженерних рішень	фізична інфраструктура постійно розвивається, кожне рішення є новим
технологічна наступність	існування мов високого рівня та технологій розробки, які можуть використовуватись фахівцями, підготовленими за попередньої парадигми (відсутня необхідність масового навчання основ квантової механіки)
підтримка дослідницького співтовариства	наявна підтримка всіх зацікавлених у нових

Критерій	Показник
	обчислювальних парадигмах та нових рівнях комп'ютерної потужності
міжнародний характер досліджень	нова фундаментальна область інженерії програмного забезпечення
є загальнозрозумілим та захоплює увагу широкої громадськості, а також викликає повагу фахівців із суміжних дисциплін	не є загальнозрозумілим, але є відомим завдяки багатосвітовій інтерпретації
проблема має давнє формулювання і досі не розв'язана	сформульована Р. Фейнманом наприкінці 1970-х рр.
обіцяє вийти за початкові межі можливого і вимагає технологій із засобів, невідомих на початку проекту	проблеми існують на кожному рівні – від розробки цілісної нової концептуальної парадигми до створення засобів моделювання
потребує планової співпраці між певними дослідницькими групами	необхідними є дослідження за рядом напрямів (мови програмування, алгоритми, інструменти, моделювання, візуалізація тощо)
виграє від конкуренції серед дослідників та команд з чіткими критеріями визначення переможця	не повинно бути єдиного «переможця» – як і в класичній інженерії програмного забезпечення слід заохочувати різноманіття рішень для широко прикладного застосування
можливо визначення проміжних цілей дослідження, досягнення яких приносить наукову чи економічну користь, навіть якщо проект у цілому зазнає невдачі	є кілька складових проблеми, які можна досліджувати і
призведе до радикальної зміни парадигми	квантові обчислення є докорінною зміною парадигми

У 2020 році інженерія квантового програмного забезпечення включає в себе такі складники [80]:

1. Парадигми розробки квантового програмного забезпечення
2. Проектування квантового програмного забезпечення
3. Тестування квантового програмного забезпечення
4. Верифікація квантового програмного забезпечення
5. Практики кодування квантового програмного забезпечення
6. Повторне використання квантового програмного забезпечення
7. Експериментування із квантовим програмним забезпеченням
8. Виконання квантового програмного забезпечення
9. Промислові застосування квантового програмного забезпечення
10. Емпіричні застосування квантового програмного забезпечення

У лютому 2020 року на QANSWER 2020 – першому міжнародному семінарі з інженерії квантового програмного забезпечення та програмування –

було прийнято Талаверський маніфест, що містить наступні принципи [68, с. 3-4]:

Інженерія квантового програмного забезпечення

– *інваріантна щодо квантових мов програмування та технологій;*

– *підтримує співіснування класичних та квантових обчислень* та виступає за використання методів реінжинірингу для інтеграції нових квантових алгоритмів до існуючих класичних інформаційних систем, а також для аналізу та подання інформації з квантових програм, яка має бути інтегрована до класичних програм;

– *підтримує управління проектами розробки квантового програмного забезпечення*, поставляючи квантове програмне забезпечення, яке відповідає вихідним бізнес-цілям та вимогам, одночасно забезпечуючи належне дотримання обмежень щодо якості, часу та витрат: методології розробки квантових програм повинні бути створені або адаптовані з існуючих; також необхідно передбачити методи оцінки зусиль для розробки квантового програмного забезпечення;

– *ураховує еволюцію квантового програмного забезпечення*: квантове програмне забезпечення повинно підтримуватися та розвиватися від початку експлуатації до виводу з неї, а квантова еволюція програмного забезпечення повинна здійснюватися протягом усього життєвого циклу квантового програмного забезпечення;

– *спрямована на створення квантових програм із бажаними нульовими дефектами*: це потребує визначення та застосування методів тестування і налагодження квантових програм у такий спосіб, щоб більшість дефектів можна було виявити та усунути до випуску програми;

– *забезпечує якість квантового програмного забезпечення*: управління якістю як процесу, так і продукту є важливим для створення квантового програмного забезпечення з очікуваними рівнями якості; оскільки ми не можемо покращити те, що ми не можемо виміряти, необхідно розробити нові метрики для квантових програм та квантових процесів;

– *сприяє повторному використанню квантового програмного забезпечення*, допомагаючи розробникам спільно використовувати, індексувати та знаходити квантове програмне забезпечення, яке можна використовувати повторно: це потребує вивчення дизайну та шаблонів проектування квантових програм, сприяння технічній комунікації та роботі над створенням бібліотек довідкових прикладів і демонстраційних програм;

– *розв’язує питання безпеки та конфіденційності шляхом відповідного проектування*: квантові інформаційні системи повинні бути захищеними та гарантувати конфіденційність даних і користувачів з початкових фаз розробки квантового програмного забезпечення, тобто проектування;

– *охоплює питання управління програмним забезпеченням*: менеджери повинні знати про конкретні процеси, організаційні структури, принципи, політику та рамки, інформацію, культуру, етику та поведінку, людей, навички та компетенції, а також про послуги, інфраструктуру та додатки, пов’язані з квантовим програмним забезпеченням, що надаються (або повинні бути надані) організаціями.

Автори маніфесту окремо звертаються до освітян із закликом уведення інженерії квантового програмного забезпечення до існуючих програм підготовки та/або окремі дисципліни, чітко визначивши, які компетентності та навички необхідні для майбутніх фахівців з інженерії квантового програмного забезпечення [68, с. 5].

1.3 Огляд засобів інженерії квантового програмного забезпечення

Виконання квантових програм на персональному комп’ютерному обладнанні є важкодоступним через його недостатню поширеність, тому більш ніж чверть сторіччя основним засобом їх виконання були квантові симулятори – програмні засоби, що моделювали квантові схеми. Перша згадка про QaaS (Quantum Computing as a Service – квантові обчислення як послуга) зустрічається лише 2015 року у статті [95] М. Рахмана (Mijanur Rahaman) та М. М. Іслама (Md. Masudul Islam).

У таблиці 1.2 подано узагальнення проведеного огляду засобів інженерії квантового програмного забезпечення.

Таблиця 1.2

Квантові симулятори, бібліотеки, візуалізатори та хмарні квантові сервіси

Засіб	Версія (дата)	Мови програмування	Опис
CHP	(2005)	C++	високоєфективний симулятор стабілізованих схем – квантових ланцюгів, які складаються із вентилів CNOT, Адамара та $\pi/2$ -фазового, а також однокубітного вимірювального вентилю [1]
Cirq	(2020)	Python	бібліотека для створення, редагування й оптимізації квантових схем та запуску їх на квантових комп'ютерах і симуляторах [88]
DDSIM	(2020)	C++	квантовий симулятор на основі діаграм рішень [42]
Eqcs	0.0.8 (2012)	C++	бібліотека для симуляції квантового комп'ютера [9]
FEYNMAN	1 (2009)	Maple	моделювання n-кубітної квантової системи [94]
Forest	2.20.0 (2020)	Python	набір інструментів та API для квантового програмування з відкритим вихідним кодом, заснований на мові квантових інструкцій Quil [96]
Intel Quantum Simulator (IQS)	(2020)	C++, Python	багатопоточний розподілений симулятор квантових ланцюгів, що моделює довільні однокубітні логічні вентиля та контрольовані (універсальні) двокубітні вентиля. IQS орієнтований на розробників алгоритмів, які хочуть протестувати своє програмне забезпечення на симуляторів та на розробників апаратного забезпечення для визначення характеристик пристроїв [44]
jaQuzzi	0.1 (2000)	Java	інтерактивний квантовий симулятор [97]
jQuantum	2.3.1 (2010)	Java	симулятор квантових схем з візуальним редактором [19]
JSQ	(2009)	Java	симулятор квантової криптографії [11]
jsqis	(2014)	JavaScript	браузерний квантовий симулятор з повним, точним і наочним зображення квантових бітів та регістрів [31]
LanQ	0.6.3 (2007)	Java	багатопроесна квантова мова програмування [48]
libquantum	1.1.1 (2013)	C++	бібліотека для квантових обчислень [12]
LibQuantumJava (LQJ)	(2018)	Java	реалізація libquantum з підтримкою до 2048 кубітів [6]
LIQ i>	(2018)	F#	мовно-інтегрований квантовий симулятор [47]
M-fun for QC Progs	(2010)	MATLAB	набір функцій для програмування квантових обчислень [108]
Microsoft Quantum	(2020)	Q#	мова та SDK квантового програмування від

Засіб	Версія (дата)	Мови програмування	Опис
Development Kit			Microsoft [55]
Multiplexor Expander	1.6 (2014)	Java	дозволяє користувачеві замінювати квантові мультиплексори послідовністю елементарних вентилів [107]
Ocean	2.2.0 (2020)	Python, C++	набір інструментів, розроблений D-Wave, який дозволяє користувачам формулювати проблеми в форматах моделі Ізінга та QUBO – результати можна отримати, надіславши дані на квантовий онлайн-комп'ютер у Leap [25]
OpenQASM	2.0 (2020)	QASM	квантовий асемблер з розширеним набором функцій (частина IBM QISKit) [78]
OpenQUACS	(2000)	Maple	симулятор квантового комп'ютера [53]
OpenQubit	0.2.0 (1999)	C++	бібліотека для симуляції квантового комп'ютера [70]
ProjectQ	0.5.1 (2020)	Python	програмне забезпечення для квантових обчислень, яке може бути перенесене до квантового симулятора або реального квантового комп'ютера [71]
PyQu	(2016)	Python	модуль квантового програмування високого рівня (за допомогою libquantum) [72]
pytket	0.5.5 (2020)	Python	квантове середовище програмування і оптимізуючий компілятор, розроблений Cambridge Quantum Computing, призначений для симуляторів і деяких квантових апаратних компонентів [103]
Q++	(2005)	C++	бібліотека шаблонів C++ для моделювання квантових обчислень, розроблена Cybernet [73]
QCF	0.1 (2007)	MATLAB	бібліотека для квантових обчислень [29]
QCGPU	(2019)	Rust	високоєфективний, апаратно прискорений за допомогою OpenCL квантовий комп'ютерний симулятор [74]
qchas	1.1.0.1 (2018)	Haskell	[2] Quantum Computing Library
QCircuits	(2020)	Python	бібліотека програмування квантових схем з простим API, розробленим для студентів [114]
QCL	0.6.4 (2014)	C++	мова, схожа на C або Паскаль, інтегрована з симулятором квантових обчислень [64]
QCLib	(2020)	C++	бібліотека квантових алгоритмів [90]
QCSim	(2018)	Python	квантовий симулятор [116]
QDD	0.3 (2007)	C++	бібліотека для симуляції квантового комп'ютера [40]
QDENSITY	(2007)	Mathematica	бібліотека для роботи з матрицями щільності [34]
QETLAB	0.9 (2016)	MATLAB	інструментарій для квантової теорії інформації та запутаності [76]
QGAME	(2004)	C++, Lisp	квантовий симулятор [102]
Q-gol	3 (1998)	CaML	система для симуляції квантових обчислень [4]
QI	(2020)	Mathematica	бібліотека для символічного аналізу квантових станів та операцій [56]

Засіб	Версія (дата)	Мови програмування	Опис
QIMatrix	1.0 (2003)	C++	шаблонний матричний клас із усіма звичайними операторами та деякими додатковими функціями, які зазвичай використовуються в квантовій механіці [17]
Qinf	(2018)	Maxima	бібліотека для квантової теорії інформації [49]
QIO	1.3 (2016)	Haskell	Quantum IO Monad – бібліотека для квантових обчислень в Haskell; може розглядатися як вбудована у Haskell мова з функціями для імітації виконання квантових обчислень [39]
QISKit	0.19.4 (2020)	Python	платформа квантових обчислень для використання сучасних квантових процесорів і проведення досліджень [77]
qitensor	0.11 (2013)	Python	матричні квантові операції у Sage [79]
qlambda	(2003)	Scheme	функціональна мова на основі Scheme для вираження та моделювання квантових алгоритмів [110]
QLib	1.0 (2007)	MATLAB	бібліотека для квантових обчислень [51]
qmatrix	(2001)	Mathematica	бібліотека об'єктів та функцій для виконання типових обчислень з теорії квантової інформації у зручний та інтуїтивний спосіб [28]
QMDD	(2016)	C++	бібліотека діаграм прийняття рішень для ефективного представлення та управління квантовою функціональністю [62]
qMIPS101	(2013)	Java	Qubit101 – корисний і ефективний симулятор квантових схем та qMIPS – симулятор процесора з інтегрованим квантовим функціональним блоком [18]
QML	(2008)	LISP / ML	функціональна квантова мова програмування [38]
QOCS	(2017)	OCaml	квантовий симулятор [22]
qotoolbox	0.15 (2002)	MATLAB	бібліотека для квантової оптики [105]
Qrack	(2020)	C++	високооптимізований квантовий симулятор, що підтримує OpenCL. Використовується або як бібліотека в більшій програмі, або безпосередньо через емулятор VM6502Q. Підтримує широкий діапазон стандартних та спеціальних вентилів, спрямований на розробку універсальних віртуальних квантових процесорів [112]
qsims	(2005)	C++	бібліотека для квантового моделювання загального призначення, здатний моделювати динаміку систем із широким діапазоном гамільтоніанів [8]
QSWalk.jl	(2020)	Julia	бібліотека для високопродуктивного аналізу квантових стохастичних блукань [32]
QTM simulator	(1995)	C++	квантовий симулятор машини Тюрінга [81]
Quacee	(2017)	Lisp	мова квантових обчислень Quacee [41]
QuaEC	1.0.1 (2018)	Python	бібліотека для роботи з квантовою корекцією помилок [37]

Засіб	Версія (дата)	Мови програмування	Опис
Quantavo	(2008)	Maple	бібліотека для лінійної оптики та квантової інформації у просторі Фока [27]
Quantencomputer-Simulator	(2005)	MATLAB	симулятор квантового комп'ютера [10]
Quantinf	0.5.1 (2013)	MATLAB	бібліотека функцій квантової інформатики [16]
Quantomatic	(2018)	Java	високорівневі графічні обчислення для квантової інформації [82]
Quantum	2.3.0 (2011)	Mathematica	бібліотека для моделювання квантових алгоритмів [35]
Quantum Algorithm Designer	(2004)	Java	графічний конструктор алгоритмів [63]
Quantum Computing Playground	(2014)	JavaScript	браузерний квантовий симулятор [84]
Quantum Computer Simulator	(2018)	JavaScript	браузерний квантовий симулятор для програмування на OpenQASM [30]
Quantum Computer Simulator	(2000)	C++	симулятор квантового комп'ютера [60]
Quantum Fog (QFog)	(2019)	Python	бібліотека аналізу класичних та квантових байєсівських мереж [20]
Quantum Network Computing	0.0.1 (2003)	C++, Python	середовище для розробки квантових комп'ютерних моделей [85]
Quantum Octave	(2020)	MATLAB	бібліотека для моделювання скінченно вимірних квантових систем [57]
Quantum Toolkit	(2013)	C++	засіб для швидкої розробки квантово-механічних моделей [86]
Quantum.NET	(2017)	C# / .NET	бібліотека для маніпулювання кубітами та моделювання квантових схем [7]
Quantum++	2.4 (2020)	C++	бібліотека квантових обчислень загального призначення, що складається виключно з файлів заголовків шаблонів. Quantum++ написаний у стандарті C++11 і має дуже низькі зовнішні залежності, використовуючи лише бібліотеку шаблонів лише лінійної алгебри Eigen 3 та, за наявності, багатопроцесорну бібліотеку OpenMP [100]
quantum-circuit	(2020)	JavaScript	20+-кубітний квантовий симулятор з можливістю імпорту з OpenQASM і Quil та експорту до OpenQASM, pyQuil, Quil, Qiskit, Cirq, TensorFlow Quantum, QSharp, і QuEST [87]
QuantumComputer	(2001)	Haskell	модуль для квантового комп'ютерного моделювання [99]
Quantum-Entanglement	0.32 (2002)	Perl	заплутування змінних у Perl [36]
QuantumOptics.jl	(2020)	Julia	бібліотека для моделювання різних видів відкритих квантових систем [46]
Quantum-Superpositions	2.0.3 (2020)	Perl	квантово-механічна суперпозиція у Perl [50]
QuantumUtils	(2018)	Mathematica	бібліотека для квантових обчислень [89]

Засіб	Версія (дата)	Мови програмування	Опис
QuantumWalk	(2020)	Julia	бібліотека для побудови алгоритмів на основі квантових блукань [33]
QUBIT4MATLAB	5.6 (2019)	MATLAB	бібліотека для квантових обчислень та квантової оптики [106]
Qubiter	(2020)	Python	бібліотека для читання, запису, складання, моделювання квантових комп'ютерних схем [21]
Qubiter	1.11 (2007)	C++	програма для моделювання квантових комп'ютерів та квантових байєсівських мереж [109]
QuCalc	(2000)	Mathematica	бібліотека для квантових обчислень [23]
QuCoSi	(2011)	C++	бібліотека для симуляції квантового комп'ютера [104]
QuEST	(2020)	C	Quantum Exact Simulation Toolkit являє собою високоефективний симулятор універсальних квантових схем, векторів стану та матриць щільності. Підтримує OpenMP та MPI, працює на GPU, може керувати одно- та багатокубітними системами у чистому і змішаному станах та імітувати дефазуючий та деполяризуєчий шум. Є розподіленим симулятором, що забезпечує підтримку розподіленої матриці щільності [91]
QuESTlink	(2020)	Mathematica	бібліотека для системи Mathematica, що використовує QuEST для симуляції квантового комп'ютеру [92]
QuIDDPro	(2007)	C++	використовує структуру даних квантової інформаційної схеми прийняття рішень (QuIDD) для моделювання ряду важливих ланцюгів, що використовують асимптотично менше ресурсів часу виконання і пам'яті, ніж методи моделювання на основі явних матриць і векторів [111]
QuIDE	(2016)	C#	платформа квантового комп'ютерного моделювання [66]
Quipper	0.9.0.0 (2019)	Haskell	убудована, масштабована функціональна мова програмування для квантових обчислень [98]
QuSAnn	1.6 (2014)	Java	генератор коду для квантового модельного відпалювання [107]
QuTiP	4.5.0 (2020)	Python	засіб для розв'язання відкритих квантових систем з використанням широкого спектру основних рівнянь і стохастичних методів [61]
QWalk	(2014)	C	симулятор квантових блукань для одно- та двовимірних решіток [52]
QWIRE	(2020)	Coq	автоматизована система доведення правильності квантових програм [43]
QX Simulator	1.0 (2016)	C++	високоефективний універсальний квантовий комп'ютерний симулятор, який імітує виконання квантових схем на ідеальних або схильних до помилок квантових комп'ютерах. Вхід QX-симулятора – це квантові схеми, які описані за допомогою QASM [93]
ScaffCC	5.0 (2020)	C	компілятор C-подібної мови Scaffold у OpenQASM для визначення та реалізації квантових алгоритмів

Засіб	Версія (дата)	Мови програмування	Опис
			[26]
sparse_pauli	(2019)	Python	мінімалістична реалізація великих розріджених операторів Паулі з використанням пар множин [15]
SpinDec	(2015)	C++	бібліотека спінової декогеренції, що реалізує кластерне кореляційне розширення [5]
sqcct	(2019)	C++	програмне забезпечення для точного та наближеного синтезу однокубітних схем [45]
Squankum	(2012)	Java	простий інтерактивний квантовий симулятор з візуалізацією [113]
staq	1.3 (2020)	C++	бібліотека для синтезу, перетворення, оптимізації та складання квантових схем. Може бути використана для прямої підтримки розбору та маніпулювання мікросхемами, описаними мовою опису QASM [101]
Strange	(2020)	Java	квантовий симулятор з API [75]
TRQS	0.2.2 (2012)	Mathematica	бібліотека для генерування справжніх випадкових квантових станів [58]
VQS - Visual Quantum Simulator	(2020)	Scala	універсальний квантовий симулятор [54]

Найбільші у світі постачальники QCaaS:

- D-Wave Systems Inc. (Канада) – SDK Ocean (Python, C++);
- International Business Machines Corporation (США) – SDK ProjectQ (Python), Qiskit (Python);
- Cambridge Quantum Computing Limited (Велика Британія) – SDK t|ket> (Python);
- QC Ware, Corp. (США) – SDK Forge (Python);
- StationQ - Microsoft (США) – SDK LIQUi|> (F#), Microsoft Quantum Development Kit (F#);
- Rigetti Computing (США) – SDK Forest (Python).

Таким чином, основна мова програмування для хмарного доступу до квантових обчислень – Python. Інший критерій вибору постачальника QCaaS – обчислювальна потужність, вимірювана у кубітах. Цей показник є найбільшим у D-Wave Advantage – 5000 (у кластерах по 8) кубітів на основі квантової нормалізації, що звужує сферу його застосування до розв’язання задач оптимізації, які зводяться до пошуку основного стану для набору спінів. Для

універсальних квантових комп'ютерів на основі квантових схем кількість кубітів суттєво нижча й на сьогодні (червень 2020 року) є найвищою у IBM Q 53 (53 кубіти) та Google Bristlecone (72 кубіти). На жаль, Quantum Computing Playground [84] від Google є браузерним квантовим симулятором, і хмарний доступ до Google Bristlecone. Для хмарного доступу до IBM Q можна використати як їх власну бібліотеку – Qiskit, так й сторонню – ProjectQ. Ураховуючи, що найбільший рівень спеціалізації надає власний SDK, для подальшої роботи було обрано саме Qiskit.

Висновки до розділу 1

1. Квантовий комп'ютер є обчислювальним пристроєм, що використовує явища квантової механіки (квантову суперпозицію, квантову сплутаність та ін.) для передавання та опрацювання даних.

2. Квантове програмування – процес розробки програм для квантового комп'ютера (практики кодування квантового програмного забезпечення) – є складовою інженерії квантового програмного забезпечення, до складу якої входять також парадигми розробки, проектування, тестування, верифікація, повторне використання, експериментування, виконання, промислові та емпіричні застосування квантового програмного забезпечення.

3. Виконання квантових програм є можливим на традиційному комп'ютерному обладнанні в режимі емуляції та на квантових комп'ютерах із хмарним доступом (QaaS, квантові обчислення як послуга). Проведений огляд більш ніж 100 засобів інженерії квантового програмного забезпечення надав можливість виділити основні їх класи (квантові симулятори, бібліотеки, візуалізатори та хмарні квантові сервіси) та рекомендувати до використання IBM Q в якості апаратної платформи для квантових обчислень, Qiskit в якості бібліотеки квантових алгоритмів, Python в якості мови програмування та IBM Quantum Experience в якості постачальника QaaS.

РОЗДІЛ 2

КВАНТОВО ПОКРАЩЕНЕ МАШИННЕ НАВЧАННЯ

2.1 Квантові моделі машинного навчання

С. Аруначалам (Srinivasan Arunachalam) та Р. де Вульф (Ronald de Wolf) у [3, с. 6-8] пропонують три квантові моделі машинного навчання:

1. *Квантове точне навчання* (exact learning) на основі запитів приналежності (membership queries) для пошуку якомога більш точної невідомої функції (квантова задача апроксимації). Ефективність квантових алгоритмів по відношенню до класичних в цьому випадку залежить від того, як вимірюється ефективність навчання. Якщо мірою ефективності є час навчання, то існують такі класи функцій, для яких квантові алгоритми значно швидше за класичні за умови можливості здійснення запитів, які перебувають у квантовій суперпозиції.

2. *Квантове ймовірнісне наближене коректне навчання* (Probably Approximately Correct, PAC) для пошуку невідомої функції за набором зразків (квантове навчання із учителем). Відмінністю квантового PAC-навчання від класичного є те, що набір даних може перебувати у стані квантової суперпозиції.

3. *Квантове агностичне навчання* (agnostic learning) для пошуку $(n+1)$ -го біту, що є продовженням послідовності з n бітів (квантова задача прогнозування).

Автори вказують на три типи складності, що виникають при застосуванні квантових моделей навчання [3, с. 20]:

а) складність запитів квантового точного навчання: кількість квантових запитів приналежності, необхідних для точного навчання, концептуально може бути поліномально меншою, ніж кількість запитів у класичній моделі точного навчання, але не набагато меншою;

б) складність зразків: для незалежних від розподілу PAC-моделей та моделей агностичного навчання опрацювання квантово заплутаних наборів

даних не дає істотної переваги над класичними випадковими вибірками: для кожного концептуального класу, складності класичної та квантової вибірок однакові з точністю до сталої. Навпаки, для деяких фіксованих розподілів (наприклад, однорідних) квантові вибірки можуть бути набагато краще, ніж класичні;

в) часова складність: для всіх квантових моделей навчання існують концептуальні класи даних, які можуть бути опрацьовані суперполіноміально швидше за допомогою квантових комп'ютерів, ніж класичні, наприклад, на основі алгоритму Шора чи Саймона.

У випадку застосування квантових моделей машинного навчання до аналізу традиційних даних мова йде про квантово-покращене машинне навчання. Ф. Філіпсон (Frank Phillipson) у [67] визначає три його головні переваги:

– покращення часу виконання (наприклад, за допомогою квантової гібридної машини Гельмгольца);

– поліпшення здатності до навчання (наприклад, за допомогою квантової нейронної мережі Хопфілда);

– підвищення ефективності навчання: менший обсяг вибірки або більш прості моделі, необхідні для отримання однакових результатів або встановлення більш складних зв'язків.

Для підвищення ефективності навчання можуть бути застосовані різноманітні алгоритми, одним з яких є варіаційні квантові схеми (variational quantum circuits – VQC) [67, с. 55].

Свідченням інтенсивності розвитку квантово покращеного машинного навчання є той факт, що систематичний огляд проблеми 2016 року, виконаний П. Віттеком (Peter Wittek) у [115], на сьогодні (червень 2020 року) вже вважається класичним, про що вказано співавтором його нового огляду [24].

В. Дунько (Vedran Dunjko) та П. Віттек також виокремлюють такі перспективні напрями розвитку квантового машинного навчання у цілому:

а) для навчання з учителем та без учителя: неперервно-змінні квантові

нейронні мережі, квантові згорткові нейронні мережі, квантові алгоритми для нейронних мереж з прямим поширенням помилки, байєсівське глибоке навчання, сублінійні квантові алгоритми для навчання лінійних та заснованих на ядрі класифікаторів;

б) навчання з підкріпленням: квантові алгоритми розв'язання задач динамічного програмування (у тому числі приховані квантові Марковські моделі), квантові методи градієнтного спуску.

Автори роблять висновок, що «вся галузь «дійсно квантового» машинного навчання (де самі дані є квантовими) все ще шукає своє місце та повне визнання. Можливо, коли квантові технології стануть зрілими, а проблеми квантового навчання стануть дійсно практичними, галузь буде кристалізуватися і зростати. ... Таким чином, квантове машинне навчання різноманітне, розширюване, інклюзивне і сповнене відкритих питань. ... Виокремлення всіх [його] трендів, які в кінцевому підсумку стануть центральними, на даний момент є неможливим завданням – і, в певному сенсі, це ключове повідомлення цієї статті» [24].

2.2 Огляд засобів квантово покращеного машинного навчання у Qiskit

Qiskit надає можливість розробки квантового програмного забезпечення як на рівні квантових схем з використанням OpenQASM, так й на високому рівні абстракції з використанням Python у Jupyter notebook. Основними складовими бібліотеки є

- засоби для моделювання квантових схем (Terra);
- реалізація стандартних квантових алгоритмів (Aqua – Algorithms for QUantum Applications), зокрема, для розв'язання задач оптимізації;
- засоби для хмарних квантових обчислень (Aer);
- засоби для моделювання квантового «шуму» (Ignis).

На рис. 2.1 показано місце Aqua в екосистемі Qiskit. До складу Aqua входять модулі для досліджень у галузі фінансів (qiskit.finance), машинного навчання (qiskit.ml), оптимізації (qiskit.optimization) та хімії (qiskit.chemistry).

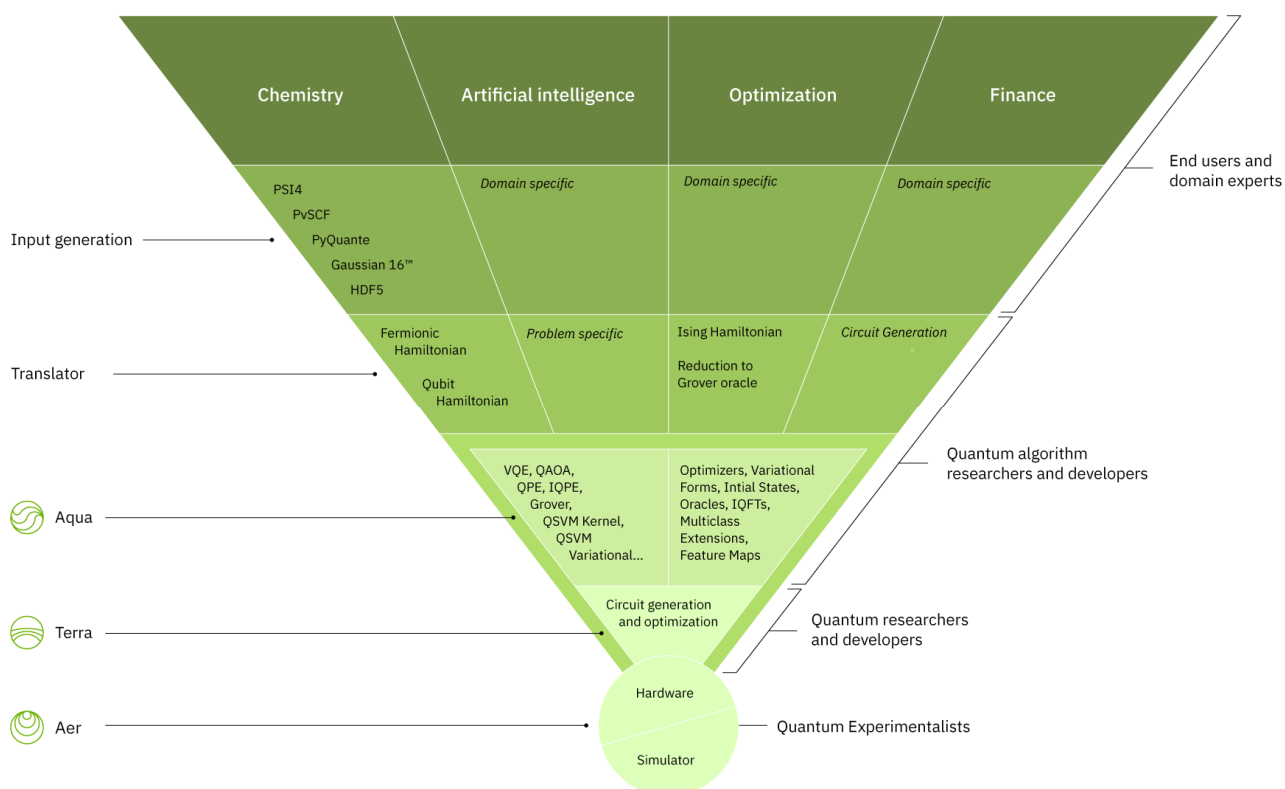


Рис. 2.1. Місце Aqua в екосистемі Qiskit (за [69])

Модуль для машинного навчання містить стандартні набори даних і способи доступу до користувачських. Для їх опрацювання можуть бути використані різні алгоритми оптимізації:

`ADMMOptimizer` – реалізація методу пошуку змінних множників (ADMM alternating direction method of multipliers);

`CobyalaOptimizer` – реалізація методу обмеженої оптимізації за допомогою лінійної апроксимації (COBYLA – Constrained Optimization BY Linear Approximation);

`CplexOptimizer` – CPLEX-оптимізатор для задач лінійного, цілочисельного та квадратичного програмування;

`GroverOptimizer` – реалізація адаптивного пошуку Гровера для пошуку мінімуму QUBO-функції (Quadratic unconstrained binary optimization – квадратична необмежена бінарна оптимізація);

`MinimumEigenOptimizer` – пошук мінімальних власних значень;

`RecursiveMinimumEigenOptimizer` – метаалгоритм для рекурсивної

оптимізації MinimumEigenOptimizer.

Модуль `qiskit.aqua.components.optimizers` пропонує набір алгоритмів для локальної (квантовий градієнтний спуск, обмежена оптимізація за допомогою оптимізатора лінійної апроксимації, оптимізатор Нелдера-Міда, алгоритм Наканіші-Фуджі-Тодо, оптимізатор Пауелла, обмежений метод Ньютона та ін.) і глобальної оптимізації (контрольований випадковий пошук із локальним оптимізатором мутацій, еволюційний пошук та ін.).

Для розв'язання задач класифікації доцільним є використання квантового методу опорних векторів (QSVM) та варіаційного квантового класифікатора (VQC).

Висновки до розділу 2

1. Застосування алгоритмів машинного навчання для аналізу квантових даних (так зване «справжнє квантове машинне навчання») може бути описане трьома квантовими моделями машинного навчання (квантове точне навчання, квантове ймовірнісне наближене коректне навчання та квантове агностичне навчання), при застосуванні яких виникають три типи утруднень, пов'язаних із складністю запитів квантового точного навчання, квантовою заплутаністю наборів даних та чутливістю до них квантових алгоритмів.

2. Квантово покращене машинне навчання – застосування квантових моделей навчання для аналізу традиційних (не квантових) даних – є перспективним напрямом розвитку машинного навчання, реалізація якого в Qiskit Aqua 0.7.3 є обмеженою розв'язанням задач класифікації з використанням квантового методу опорних векторів та варіаційного квантового класифікатора.

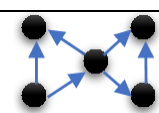
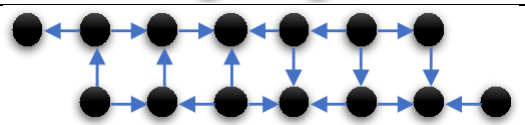
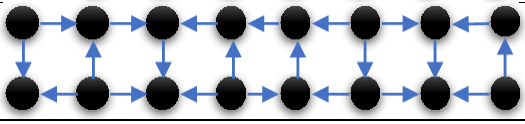
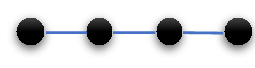
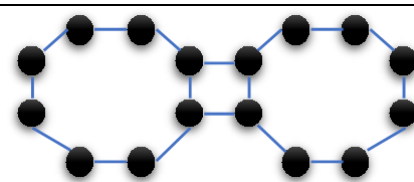
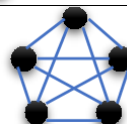
РОЗДІЛ 3

ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА ЕФЕКТИВНОСТІ КВАНТОВО ПОКРАЩЕНИХ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ

При реалізації квантових обчислень на реальних архітектурах необхідно, крім класичної проблеми декогеренції, що суттєво обмежує час роботи квантових алгоритмів (70-150 мкс для 20-кубітної системи IBM Q System One із максимальною кількістю одночасно заплутаних кубітів, рівною 6), також урахувувати помилки зчитування результатів. Автори [13] надають наступні значення для квантових комп'ютерів, доступних через QaaS (таблиця 3.1).

Таблиця 3.1

Характеристики поширених квантових комп'ютерів (за [59])

Комп'ютер	Кількість кубітів	Кількість двохкубітних вентилів	Тривалість когеренції, мкс	Однокубітні помилки, %	Двохкубітні помилки, %	Помилки зчитування результату, %	Топологія кубітів
IBM Q5 Tenerife	5	6	40	0,2	4,76	6,21	
IBM Q14 Melbourne	14	18	30	1,19	7,95	9,09	
IBM Q16 Rueschlikon	16	22	40	0,22	7,14	4,15	
Rigetti Agave	4	3	15	3,68	10,8	16,37	
Rigetti Aspen1	16	18	20	3,43	8,92	5,56	
Rigetti Aspen3	16	18	20	3,79	5,37	6,65	
UMD Trapped Ion	5	10	$1,5 \cdot 10^6$	0,2	1,00	0,6	

Таким чином, на поточному стану розвитку квантових технологій

необхідним є використання таких наборів даних, кроки опрацювання яких алгоритмами квантово покращеного машинного навчання відповідають вимогам до обраної платформи – ІВМ Q.

Машинне навчання виконувалось на наборах даних wine та breast_cancer з використанням бібліотек Qiskit (квантово покращене) та sklearn (традиційне). Тестування виконувалось на локальному двохкубітному квантовому симуляторі з 8 Гб оперативної пам'яті та серверах ibmqx2 (2 кубіти), ibmq_16_melbourne (16 кубітів), ibmq_vigo, ibmq_london та ibmq_burlington (всі – 5 кубітів). процедура машинного навчання для кожного набору даних повторювалась 6 разів. Результати подано у таблиці 3.2.

Таблиця 3.2

Експериментальна перевірка ефективності машинного навчання на квантових та традиційних архітектурах

Сервер	Час квантово покращеного машинного навчання, с		Час традиційного машинного навчання, с	
	wine	breast_cancer	wine	breast_cancer
ibmqx2	24.8	18		
	24.7	18.1		
	24.5	17.9		
	24.6	18.1		
	24.8	18.1		
	24.7	17.8		
ibmq_16_melbourne	25	18.9		
	25.2	18.7		
	25.2	10.2		
	25.1	10.2		
	25.2	18	0.021	0.027
	24.8	18.9	0.021	0.027
ibmq_vigo	28.3	20.4	0.026	0.026
	28.3	20.4	0.026	0.027
	28.1	20.4	0.026	0.027
	28	20.7	0.026	0.026
	26.9	20.4		
	28.3	20.7		
ibmq_london	26.1	19.4		
	26.2	19.3		
	26.5	19.6		
	26.5	19.5		
	26.3	19.4		
	26.2	19.7		
ibmq_burlington	27.3	20.4		
	27.1	20.4		

Сервер	Час квантово покращеного машинного навчання, с		Час традиційного машинного навчання, с	
	wine	breast cancer	wine	breast cancer
	26.7	20.3		
	27.1	20.5		
	27	20.3		
	27.1	20.1		
локальний квантовий емулятор	111	22.2	0.190	0.021
	105.1	23.3	0.020	0.023
	102.8	32.7	0.028	0.021
	103.5	31.7	0.021	0.021
	111.2	22.4	0.020	0.026
	99.9	22.1	0.022	0.021

При застосування sklearn на IBM Q Experience можливості визначити, на якому саме сервері відбувається виконання, не було – багаторазове повторення тестів у різний час доби не призвело до суттєвої зміни результатів.

Код для всіх видів тестів подано у додатках до роботи.

Висновки до розділу 3

1. Аналіз таблиці 3.2 надає можливість зробити висновок про те, що на поточному етапі розвитку квантових технологій традиційне машинне навчання забезпечує більшу продуктивність, ніж квантово покращене. При цьому квантово покращені алгоритми машинного навчання виявились обернено чутливими до складності набору даних: навчання на більш складному наборі даних breast_cancer (30 вхідних параметрів, 2 вихідних, 569 елементів) виконувалось із більшою швидкістю, ніж навчання на менш складному наборі даних wine (13 вхідних параметрів, 3 вихідних, 178 елементів), у той час як за традиційного машинного навчання спостерігалась пряма залежність, підтверджена [13] та іншими джерелами.

2. Результати аналізу надають можливість зробити припущення про те, що квантово покращене машинне навчання доцільно застосовувати до наборів даних з великою розмірністю входу, передбачуване значення для яких є ймовірністю вибору одного з двох наборів класів – такі класи ефективно опрацьовуються однокубітними системами.

ВИСНОВКИ

Розв'язання задач дослідження надало можливість зробити наступні висновки:

1. Ядром інженерії квантового програмного забезпечення є квантове програмування – процес розробки програм для квантового комп'ютера: обчислювального пристрою, що використовує явища квантової механіки для опрацювання даних. Через низький рівень доступності таких пристроїв доцільним є доступ до них за моделлю QaaS – «квантові обчислення як послуга». Проведений огляд засобів інженерії квантового програмного забезпечення надав можливість виділити основні їх класи (квантові симулятори, бібліотеки, візуалізатори та хмарні квантові сервіси) та рекомендувати до використання IBM Q в якості апаратної платформи для квантових обчислень, Qiskit в якості бібліотеки квантових алгоритмів, Python в якості мови програмування та IBM Quantum Experience в якості постачальника QaaS.

2. Застосування алгоритмів машинного навчання для аналізу квантових даних може бути описане трьома квантовими моделями машинного навчання (квантове точне навчання, квантове ймовірнісне наближене коректне навчання та квантове агностичне навчання), при застосуванні яких виникають три типи утруднень, пов'язаних із складністю запитів квантового точного навчання, квантовою заплутаністю наборів даних та чутливістю до них квантових алгоритмів. Перспективним напрямом розвитку машинного навчання є застосування квантових моделей навчання для аналізу традиційних даних, реалізація яких в Qiskit Aqua 0.7.3 є поки що обмеженою розв'язанням задач класифікації.

3. Результати експерименту із застосуванням варіаційного квантового класифікатору на двох наборах даних показали, що на поточному етапі розвитку квантових технологій традиційне машинне навчання забезпечує більшу продуктивність, ніж квантово покращене. При цьому застосування

квантово покращених алгоритмів машинного навчання для задач бінарної класифікації навіть з високою розмірністю вхідних даних дає суттєве (у кілька разів) прискорення порівняно із задачами тернарної класифікації, у той час як при застосуванні традиційного машинного навчання час виконання зростає у залежності від обсягу набору даних та його розмірності. Проведений аналіз результатів експерименту надав можливість зробити припущення про те, що квантово покращене машинне навчання доцільно застосовувати до наборів даних з великою розмірністю входу, передбачуване значення для яких є ймовірністю вибору одного з двох наборів класів – такі класи ефективно опрацьовуються однокубітними системами.

Перспективи подальших досліджень полягають у системному дослідженні можливостей інженерії квантового програмного забезпечення та його застосування до розв'язання задачі прогнозування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Aaronson S. CHP: CNOT-Hadamard-Phase [Electronic resource] / Scott Aaronson, Daniel Gottesman. – Access mode : <http://www.scottaaronson.com/chp/>
2. Ardelean M. S. qchas: A library for implementing Quantum Algorithms [Electronic resource] / Mihai Sebastian Ardelean. – 2018. – Access mode : <https://hackage.haskell.org/package/qchas>
3. Arunachalam S. A Survey of Quantum Learning Theory [Electronic resource] / Srinivasan Arunachalam, Ronald de Wolf. – arXiv:1701.06806 [quant-ph]. – 2017. – Access mode : <https://arxiv.org/abs/1701.06806>
[Electronic resource]. – Access mode :
4. Baker G. D. Q-gol [Electronic resource] / Gregory David Baker. – Access mode : <http://www.ifost.org.au/~gregb/q-gol/>
5. Balian S. sbalian / spindec – Bitbucket [Electronic resource] / Seto Balian. – 2015. – Access mode : <http://bitbucket.org/sbalian/spindec>
6. Banegas G. S. gbanegas/libQuantumJava [Electronic resource] / Gustavo Souza Banegas. – 2018. – Access mode : <https://github.com/gbanegas/libQuantumJava>
7. Baudin P.-H. phbaudin/quantum-computing: Quantum.NET: a library to manipulate qubits and simulate quantum circuits [Electronic resource] / Pierre-Henry Baudin. – 2017. – Access mode : <https://github.com/phbaudin/quantum-computing>
8. Beals T. qsims: Quantum Simulation Software [Electronic resource] / Travis Beals. – 2005. – Access mode : <http://qsims.sourceforge.net/>
9. Belkner P. Eqcs-0.0.8 [Electronic resource] / Peter Belkner. – 2012. – Access mode : <http://home.snafu.de/pbelkner/eqcs/>
10. Bornemann F. Lehrstuhl Numerische Mathematik und Wissenschaftliches Rechnen - M3/Software - QCWebHome [Electronic resource] / Folkmar Bornemann. – 2005. – Access mode : <http://www-m3.ma.tum.de/twiki/bin/view/Software/QCWebHome>

11. Bouvarel B. Simulateur de Cryptographie Quantique download | SourceForge.net [Electronic resource] / Benoit Bouvarel, Olivier Oudin, Lionel Vallier. – 2009. – Access mode : <http://sourceforge.net/projects/simu-quantique/>
12. Butscher B. libquantum - Simulation of quantum mechanics [Electronic resource] / Bjoern Butscher, Hendrik Weimer. – 2013. – Access mode : <http://www.enyo.de/libquantum/>
13. Cheng C.-H. New entropy clustering analysis method based on adaptive learning / Ching-Hsue Cheng, Liang-Ying Wei // Proceedings of the 10th Joint Conference on Information Sciences 2007 / Ed. by Paul P Wang. – 2007. – P. 1196–1202. – DOI : 10.1142/9789812709677_0169
14. Clark J. A. Quantum Software Engineering [Electronic resource] / John Clark, Susan Stepney // Workshop on Grand Challenges for Computing Research. – e-Science Institute, Edinburgh, Scotland – 24-26 November 2002. – Access mode : <http://www.ukcrc.org.uk/press/news/call/a5.cfm>
15. Criger B. bcriger/sparse_pauli: Implementation of large, sparse Pauli operators using pairs of sets. Contains absolutely minimal functionality, and is an example of extreme “physicist code” [Electronic resource] / Ben Criger. – Access mode : https://github.com/bcriger/sparse_pauli
16. Cubitt T. Toby 'qubit' Cubitt - Matlab code [Electronic resource] / Toby Cubitt. – 2013. – Access mode : <http://www.dr-qubit.org/matlab.html>
17. Dawson C. QIMatrix: A Quantum Information Matrix Toolkit [Electronic resource] / Christopher Dawson. – 2003. – Access mode : <http://web.archive.org/web/20091221074946/http://www.physics.uq.edu.au:80/people/dawson/matrix/doc/>
18. de Portugal Vazquez J. C. qMIPS101 Quantum Computing Simulators [Electronic resource] / Jaime Coello de Portugal Vazquez. – Access mode : <http://institucional.us.es/qmipsmaster/>
19. de Vries. jQuantum Quantum Computer Simulation [Electronic resource] / de Vries. – 2010. – Access mode : <http://jqquantum.sourceforge.net/>
20. Dekant H. artiste-qb-net/quantum-fog: Python tools for analyzing both classical

- and quantum Bayesian Networks [Electronic resource] / Dekant Henning, Tregillus Henry, Tucci Robert, Yin Tao. – 2019. – Access mode : <https://github.com/artiste-qb-net/quantum-fog>
21. Dekant H. *artiste-qb-net/qubiter*: Python tools for reading, writing, compiling, simulating quantum computer circuits. Includes numpy and tensorflow backends. “Quantum Space, the final frontier. These are the voyages of the starship Qubiter. Its five-year mission: to explore strange new worlds, to seek out new life and new civilizations, to boldly go where no man has gone before.” [Electronic resource] / Dekant Henning, Tregillus Henry, Tucci Robert, Yin Tao. – 2020. – Access mode : <https://github.com/artiste-qb-net/qubiter>
 22. *dillanchang/QOCS* [Electronic resource]. – 2017. – Access mode : <https://github.com/dillanchang/QOCS>
 23. Dumais P. *QuCalc: A Quantum Computation Package* -- from Wolfram Library Archive [Electronic resource] / Paul Dumais. – Access mode : <https://library.wolfram.com/infocenter/MathSource/657/>
 24. Dunjko V. A non-review of Quantum Machine Learning: trends and explorations / Vedran Dunjko, Peter Wittek // *Quantum Views*. – 2020. – Vol. 4. – P. 32. – DOI : 10.22331/qv-2020-03-17-32
 25. D-Wave's Ocean Software [Electronic resource]. – 2020. – Access mode : <https://ocean.dwavesys.com/>
 26. *epiqc/ScaffCC*: Compilation, analysis and optimization framework for the Scaffold quantum programming language [Electronic resource]. – Access mode : <https://github.com/epiqc/ScaffCC>
 27. Feito Boirac A. *alphydan/Quantavo: Quantum Optics Maple Toolbox* [Electronic resource] / Alvaro Feito Boirac. – 2012. – Access mode : <http://www3.imperial.ac.uk/quantuminformation/research/downloads>
 28. Felbinger T. *qmatrix: A Package for Quantum Information Theory* -- from Wolfram Library Archive [Electronic resource] / Timo Felbinger. – 2001. – Access mode : <http://library.wolfram.com/infocenter/MathSource/1893/>
 29. Fox C. QCF download | SourceForge.net [Electronic resource] / Charles Fox. –

- Access mode : <https://sourceforge.net/projects/qcf/>
30. Gaitatzis T. Quantum Computer Simulator [Electronic resource] / Tony Gaitatzis. – 2018. – Access mode : <https://backupbrain.github.io/quantum-compiler-simulator/>
 31. Garrison J. garrison/jsqis: Javascript quantum information simulator [Electronic resource] / Jim Garrison. – 2014. – Access mode : <https://github.com/garrison/jsqis>
 32. Glos A. iitis/QSWalk.jl: QSWalk.jl: simulating the evolution of open quantum systems on graphs [Electronic resource] / Adam Glos, Jarosław Adam Mischczak, Mateusz Ostaszewski. – 2020. – Access mode : <https://github.com/QuantumWalks/QSWalk.jl>
 33. Glos A. iitis/QuantumWalk.jl: QuantumWalk.jl: Package for building algorithms based on quantum walks [Electronic resource] / Adam Glos, Jarosław Adam Mischczak. – 2020. – Access mode : <https://github.com/QuantumWalks/QuantumWalk.jl>
 34. Gomez J. L. QDENSITY download page [Electronic resource] / Jose Luis Gomez. – 2017. – Access mode : <http://www.pitt.edu/~tabakin/QDENSITY/>
 35. Gómez-Muñoz J. L. Quantum Mathematica Add-On by Jose Luis Gómez-Muñoz [Electronic resource] / José Luis Gómez-Muñoz, Francisco Delgado. – 2011. – Access mode : <http://homepage.cem.itesm.mx/lgomez/quantum/index.htm>
 36. Gough A. Quantum-Entanglement-0.32 - QM entanglement of variables in perl - metacpan.org [Electronic resource] / Alex Gough. – Access mode : <http://search.cpan.org/~ajgough/Quantum-Entanglement-0.32/>
 37. Granade C. QuaEC: Quantum Error Correction Analysis in Python – QuaEC 1.0.1 documentation [Electronic resource] / Chris Granade, Ben Criger.. – Access mode : <http://www.cgranade.com/python-quaec/>
 38. Grattage J. QML: A Functional Quantum Programming Language [Electronic resource] / Jonathan Grattage. – 2008. – Access mode : <http://web.archive.org/web/20171101170712/http://sneezy.cs.nott.ac.uk/QML/>

39. Green A. S. QIO: The Quantum IO Monad is a library for defining quantum computations in Haskell [Electronic resource] / Alexander S. Green. – Access mode : <http://hackage.haskell.org/package/QIO>
40. Greve D. QDD : A Quantum Computer Emulation Library [Electronic resource] / David Greve. – 2007. – Access mode : <http://thegreves.com/david/QDD/qdd.html>
41. Hudek K. kat31416/quacee: Quacee quantum computing language [Electronic resource] / Katherine Hudek. – 2017. – Access mode : <https://github.com/kat31416/quacee>
42. iic-jku/ddsim: DDSIM - A quantum simulator based on decision diagrams written in C++ [Electronic resource]. – Access mode : <https://github.com/iic-jku/ddsim>
43. inQWIRE/QWIRE: A quantum circuit language and formal verification tool [Electronic resource]. – Access mode : <https://github.com/inQWIRE/QWIRE>
44. iqusoft/intel-qs: High-performance simulator of quantum circuits [Electronic resource]. – [2020]. – Access mode : <https://github.com/iqusoft/intel-qs>
45. Kliuchnikov V. vadyim-kl/sqct: Exact and approximate synthesis of single qubit circuits using Clifford and T gate library [Electronic resource] / Vadym Kliuchnikov. – Access mode : <http://code.google.com/p/sqct/>
46. Krämer S. QuantumOptics.jl [Electronic resource] / Sebastian Krämer, David Plankensteiner. – 2020. – Access mode : <https://qojulia.org/>
47. Language-Integrated Quantum Operations: LIQUi> - Microsoft Research [Electronic resource]. – Access mode : <http://research.microsoft.com/en-us/projects/liquid/>
48. LanQ – a quantum imperative programming language [Electronic resource]. – 2007. – Access mode : <http://lanq.sourceforge.net/>
49. Lapeyre J. Qinf [Electronic resource] / John Lapeyre. – 2018. – Access mode : <http://www.johnlapeyre.com/qinf/index.html>
50. Lembark S. Quantum::Superpositions - QM-like superpositions in Perl - metacpan.org [Electronic resource] / Steven Lembark. – 2018. – Access mode :

- <https://metacpan.org/release/Quantum-Superpositions>
51. Machnes S. QLib, Quantum Group, Tel-Aviv University [Electronic resource] / Shai Machnes. – 2007. – Access mode : <http://www.tau.ac.il/~quantum/qlib/qlib.html>
 52. Marquezino F. L. QWalk: A Quantum Walk Simulator [Electronic resource] / F. L. Marquezino. – 2014. – Access mode : <http://www.cos.ufrj.br/~franklin/qwalk/>
 53. McCubbin C. OpenQUACS [Electronic resource] / Chris McCubbin. – Access mode : <http://web.archive.org/web/20060116174553/http://userpages.umbc.edu/~cmccub1/qaacs/qaacs.html>
 54. Ménier Gravures G. Introduction · gmenier/VisualQuantumSimulator Wiki [Electronic resource] / Gildas Ménier Gravures. – 2020. – Access mode : <https://github.com/gmenier/VisualQuantumSimulator/wiki/Introduction>
 55. Microsoft Quantum Documentation and Q# API Reference - Microsoft Quantum | Microsoft Docs [Electronic resource]. – 2020. – Access mode : <https://docs.microsoft.com/en-us/quantum/>
 56. Miszczak J. A. iitis/qi: QI package for Mathematica [Electronic resource] / J. A. Miszczak, Z. Puchała, P. Gawron. – 2020. – Access mode : <https://github.com/iitis/qi>
 57. Miszczak J. A. iitis/quantum-octave: Functions for simulatin quantum computing in GNU Octave and Matlab [Electronic resource] / J. A. Miszczak. – 2020. – Access mode : <https://github.com/iitis/quantum-octave>
 58. Miszczak J. A. TRQS - True Random Quantum States [Electronic resource] / Jarosław Adam Miszczak. – 2012. – Access mode : <https://miszczak.eu/trqs.html>
 59. Murali P. Full-Stack, Real-System Quantum Computer Studies: Architectural Comparisons and Design Insights / Prakash Murali, Norbert Matthias Linke, Margaret Martonosi, Ali Javadi-Abhari, Nhung Hong Nguyen, Cinthia Huerta Alderete // ISCA '19: Proceedings of the 46th International Symposium on

- Computer Architecture. – June 2019. – P. 527–540. – DOI : 10.1145/3307650.3322273
60. Nagai A. Quantum Computer Simulator [Electronic resource] / Ayumu Nagai, Yuuki Tokunaga. – Access mode : <http://www-imai.is.s.u-tokyo.ac.jp/~tokunaga/QCS/simulator.html>
 61. Nation P. D. QuTiP - Quantum Toolbox in Python [Electronic resource] / P. D. Nation, J. R. Johansson. – 2020. – Access mode : <http://qutip.org/>
 62. Niemann P. QMDD - a Decision Diagram Package for the Efficient Representation and Manipulation of Quantum Functionality [Electronic resource] / Philipp Niemann, Robert Wille, D. Michael Miller, Mitchell A. Thornton, and Rolf Drechsler. – Access mode : <http://www.informatik.uni-bremen.de/agra/eng/qmdd.php>
 63. O’Keefe S. Quantum Algorithm Designer [Electronic resource] / Simon O’Keefe. – Access mode : http://web.archive.org/web/20040811134547fw_/http://www-users.cs.york.ac.uk/~sok/QAD/downloads.html
 64. Ömer B. QCL - A Programming Language for Quantum Computers [Electronic resource] / Bernhard Ömer. – 2014. – Access mode : <http://tph.tuwien.ac.at/~oemer/qcl.html>
 65. Panetta K. The CIO’s Guide to Quantum Computing [Electronic resource] / Kasey Panetta // Smarter With Gartner. – April 18, 2019. – Access mode : <https://www.gartner.com/smarterwithgartner/the-cios-guide-to-quantum-computing/>
 66. Patrzyk J. QuIDE - Quantum IDE [Electronic resource] / Joanna Patrzyk, Bartłomiej Patrzyk. – 2016. – Access mode : <http://www.quide.eu/>
 67. Phillipson F. Quantum Machine Learning: Benefits and Practical Examples [Electronic resource] / Frank Phillipson // Short Papers Proceedings of the 1st International Workshop on the QuANtum SoftWare Engineering & pRogramming. Talavera de la Reina, Spain, February 11-12, 2020 / Edited by Mario Piattini, Guido Peterssen, Ricardo Perez-Castillo, Jose Luis Hevia,

- Manuel A. Serrano // CEUR Workshop Proceedings. – 2020. – Vol. 2561. – P. 51-56. – Access mode : <http://ceur-ws.org/Vol-2561/paper5.pdf>
68. Piattini M. The Talavera Manifesto for Quantum Software Engineering and Programming [Electronic resource] / Mario Piattini, Guido Peterssen, Ricardo Perez-Castillo, Jose Luis Hevia, Manuel A Serrano, Guillermo Hernández, Ignacio García Rodríguez de Guzmán, Claudio Andrés Paradela, Macario Polo, Ezequiel Murina, Luis Jiménez, Juan Carlos Marqueño, Ramsés Gallego, Jordi Tura, Frank Phillipson, Juan M. Murillo, Alfonso Niño, Moisés Rodríguez // Short Papers Proceedings of the 1st International Workshop on the QuANtum SoftWare Engineering & pRogramming. Talavera de la Reina, Spain, February 11-12, 2020 / Edited by Mario Piattini, Guido Peterssen, Ricardo Perez-Castillo, Jose Luis Hevia, Manuel A. Serrano // CEUR Workshop Proceedings. – 2020. – Vol. 2561. – P. 1-5. – Access mode : <http://ceur-ws.org/Vol-2561/paper0.pdf>
69. Pistoia M. Qiskit Aqua — A Library of Quantum Algorithms and Applications [Electronic resource] / Marco Pistoia, Jay Gambetta. – Dec 13, 2018. – Access mode : <https://medium.com/qiskit/qiskit-aqua-a-library-of-quantum-algorithms-and-applications-33ecf3b36008>
70. Pritzker Y. skwp/open-qubit: my old quantum computing simulator project from the 90's [Electronic resource] / Yan Pritzker. – 2012. – Access mode : <https://github.com/skwp/open-qubit>
71. ProjectQ – Open Source Software for Quantum Computing [Electronic resource]. – 2020. – Access mode : <https://projectq.ch/>
72. PyQu: Google Code Archive - Long-term storage for Google Code Project Hosting. [Electronic resource]. – Access mode : <http://code.google.com/p/pyqu>
73. Q++ download | SourceForge.net [Electronic resource]. – 2005. – Access mode : <http://sourceforge.net/projects/qplusplus/>
74. QCGPU - Hardware Accelerated Quantum Computer Simulation [Electronic resource]. – 2019. – Access mode : <https://qcgpu.github.io/>
75. qcjava/strange: Quantum Computing API for Java [Electronic resource]. – 2020. – Access mode : <https://github.com/gluonhq/strange>

76. QETLAB [Electronic resource]. – 2016. – Access mode : http://www.qetlab.com/Main_Page
77. Qiskit [Electronic resource]. – 2020. – Access mode : <https://qiskit.org/>
78. Qiskit/openqasm: Gate and operation specification for quantum circuits [Electronic resource]. – 2020. – Access mode : <https://github.com/Qiskit/openqasm>
79. qitensor for quantum information in python [Electronic resource]. – 2013. – Access mode : <http://www.stahlke.org/dan/qitensor>
80. Q-SE 2020: First International Workshop on Quantum Software Engineering (Q-SE 2020) co-located with ICSE 2020 [Electronic resource]. – Access mode : <https://q-se.github.io/qse2020/>
81. QTM simulator [Electronic resource]. – 1995. – Access mode : <http://web.archive.org/web/20050923134721/http://www.lri.fr/~durr/Attic/qtm/>
82. Quantomatic :: Home [Electronic resource]. – 2018. – Access mode : <http://quantomatic.github.io/>
83. Quantum Computing [Electronic resource] // Gartner Glossary. – Access mode : <https://www.gartner.com/en/information-technology/glossary/quantum-computing>
84. Quantum Computing Playground [Electronic resource] / Google. – 2016. – Access mode : <http://www.quantumplayground.net/#/home>
85. Quantum Network Computing download | SourceForge.net [Electronic resource]. – 2003. – Access mode : <http://sourceforge.net/projects/qnc/>
86. Quantum Toolkit download | SourceForge.net [Electronic resource]. – 2013. – Access mode : <http://sourceforge.net/projects/qcplusplus/>
87. quantum-circuit - npm [Electronic resource]. – 2020. – Access mode : <https://www.npmjs.com/package/quantum-circuit>
88. quantumlib/Cirq: A python framework for creating, editing, and invoking Noisy Intermediate Scale Quantum (NISQ) circuits [Electronic resource]. – 2019. – Access mode : <https://github.com/quantumlib/Cirq>
89. QuantumUtils/quantum-utils-mathematica: QuantumUtils for Mathematica is a

- software library for quantum information scientists [Electronic resource]. – 2018. – Access mode : <https://github.com/QuantumUtils/quantum-utils-mathematica>
90. Quantware MIPS Center – Library [Electronic resource]. – 2020. – Access mode : <http://www.quantware.ups-tlse.fr/QWLIB/>
 91. QuEST – Quantum Exact Simulation Toolkit [Electronic resource]. – Access mode : <https://quest.qtechtheory.org/>
 92. QuESTlink – Bringing the power of QC simulation to Mathematica [Electronic resource]. – Access mode : <https://questlink.qtechtheory.org/>
 93. QX Quantum Computer Simulator [Electronic resource]. – [2016?]. – Access mode : <http://quantum-studio.net/>
 94. Radtke T. Simulation of n-qubit quantum systems. V. Quantum measurements / T. Radtke, S. Fritzsche. – Mendeley Data. – 2020. – DOI : 10.17632/y459fn3v2s.
 95. Rahaman M. A Review on Progress and Problems of Quantum Computing as aService (QCaas) in the Perspective of Cloud Computing [Electronic resource] / Mijanur Rahaman, Md. Masudul Islam // Global Journal of Computer Science and Technology: B Cloud and Distributed. – 2015. – Vol. 15. – Iss. 4. – Access mode : https://globaljournals.org/GJCST_Volume15/3-Cloud-Data-Storage.pdf
 96. Rigetti QCS [Electronic resource]. – Access mode : <https://qcs.rigetti.com/sdk-downloads>
 97. Schuermann F. jaQuzzi - Interactive Quantum Computation [Electronic resource] / Felix Schuermann. – 2000. – Access mode : <http://www.eng.buffalo.edu/~phygons/jaQuzzi/>
 98. Selinger P. The Quipper Language [Electronic resource] / Peter Selinger. – 2019. – Access mode : <https://www.mathstat.dal.ca/~selinger/quipper/>
 99. Skibinski J. Haskell Simulator of Quantum Computer [Electronic resource] / Jan Skibinski. – 2001. – Access mode : <http://web.archive.org/web/20010803034527/http://www.numeric-quest.com/haskell/QuantumComputer.html>

100. softwareQinc/qpp: A modern C++11 quantum computing library [Electronic resource]. – Access mode : <https://github.com/softwareqinc/qpp>
101. softwareQinc/staq: A full-stack quantum processing toolkit [Electronic resource]. – 11 June 2020. – Access mode : <https://github.com/softwareqinc/staq>
102. Spector L. QGAME: Quantum and Gate Measurement Emulator [Electronic resource] / Lee Spector. – 2004. – Access mode : <http://hampshire.edu/lspector/qgame.html>
103. Technology – Cambridge Quantum Computing [Electronic resource]. – 2020. – Access mode : <https://cambridgequantum.com/technology/>
104. Thomas F. S. fthomas/qucosi [Electronic resource] / Frank S. Thomas. – 2011. – Access mode : <https://github.com/fthomas/qucosi>
105. Toolbox | Quantum Optics Theory [Electronic resource]. – 2002. – Access mode : <https://qo.phy.auckland.ac.nz/toolbox/>
106. Tóth G. QUBIT4MATLAB [Electronic resource] / Géza Tóth. – 2019. – Access mode : <http://bird.szfki.kfki.hu/~toth/qubit4matlab.html>
107. Tucci R. R. QuSAnn, Multiplexor Expander [Electronic resource] / R. R. Tucci. – 2014. – Access mode : <http://www.ar-tiste.com/qusann.html>
108. Tucci R. R. Octave/Matlab m-files for QC Programmers [Electronic resource] / R. R. Tucci. – Access mode : <http://www.ar-tiste.com/m-fun/m-fun-index.html>
109. Tucci R. R. Qubiter [Electronic resource] / R. R. Tucci. – 2007. – Access mode : <http://www.ar-tiste.com/qubiter.html>
110. van Tonder A. A lambda calculus for quantum computation [Electronic resource] / André van Tonder. – 2003. – Access mode : <http://www.het.brown.edu/people/andre/qlambda/>
111. Viamontes G. F. [Electronic resource] / George F. Viamontes, Igor L. Markov, John P. Hayes. – 2009. – Access mode : <http://vlsicad.eecs.umich.edu/Quantum/qp/>
112. VM6502Q and Qrack – qrack documentation [Electronic resource]. – 2020. – Access mode : <https://vm6502q.readthedocs.io/en/latest/>

113. Wasserman J. Squankum [Electronic resource] / Jeffrey Wasserman. – 2012. – Access mode : <http://jeffwass.github.com/Squankum/>
114. Webb A. grey-area/qcircuits: A Python package for simulating small-scale quantum computers [Electronic resource] / Andrew Webb. – 2020. – Access mode : <https://github.com/grey-area/qcircuits>
115. Wittek P. Quantum Machine Learning: What Quantum Computing Means to Data Mining / Peter Wittek. – San Diego : Academic Press, 2016. – x, 163 p.
116. Yoghurtyam/QCSim: Simulation Of A Quantum Computer In Python [Electronic resource]. – 2018. – Access mode : <https://github.com/Yoghurtyam/QCSim>

ДОДАТКИ

A Код для квантово покращеного машинного навчання на наборі даних wine (сервер `ibmq_burlington`, 5 кубітів)

```
# Importing standard Qiskit libraries and configuring account
from qiskit import QuantumCircuit, execute, Aer, IBMQ
from qiskit.compiler import transpile, assemble
from qiskit.tools.jupyter import *
from qiskit.visualization import *
from qiskit import BasicAer
from qiskit.aqua import QuantumInstance, aqua_globals
from qiskit.aqua.algorithms import VQC
from qiskit.aqua.components.optimizers import COBYLA
from qiskit.aqua.components.feature_maps import RawFeatureVector
from qiskit.ml.datasets import wine
from qiskit.circuit.library import TwoLocal
import time

# Loading your IBM Q account(s)
provider = IBMQ.load_account()

from qiskit import BasicAer
from qiskit.aqua import QuantumInstance, aqua_globals
from qiskit.aqua.algorithms import VQC
from qiskit.aqua.components.optimizers import COBYLA
from qiskit.aqua.components.feature_maps import RawFeatureVector
from qiskit.ml.datasets import wine
from qiskit.circuit.library import TwoLocal
import time

seed = 1376
aqua_globals.random_seed = seed

# Use Wine data set for training and test data
feature_dim = 4 # dimension of each data point
```

```

_, training_input, test_input, _ = wine(training_size=12,
                                         test_size=4, n=feature_dim)

instance = QuantumInstance(provider.get_backend('ibmq_burlington'),
                           shots=1024, seed_simulator=seed, seed_transpiler=seed,
                           skip_qobj_validation=True)

feature_map = RawFeatureVector(feature_dimension=feature_dim)
start_time = time.time()
vqc = VQC(COBYLA(maxiter=100),
          feature_map,
          TwoLocal(feature_map.num_qubits, ['ry', 'rz'], 'cz', reps=3),
          training_input, test_input)
result = vqc.run(instance)

print('Testing accuracy: {:.2f}'.format(result['testing_accuracy']))
print(result)

print("--- %s seconds ---" % (time.time() - start_time))

```

Б Код для квантово покращеного машинного навчання на наборі даних breast_cancer (локальний квантовий емулятор, 5 кубіти)

```

from qiskit import BasicAer
from qiskit.aqua import QuantumInstance, aqua_globals
from qiskit.aqua.algorithms import VQC
from qiskit.aqua.components.optimizers import COBYLA
from qiskit.aqua.components.feature_maps import RawFeatureVector
from qiskit.ml.datasets import breast_cancer
from qiskit.circuit.library import TwoLocal
import time
import random

seed = 1376
aqua_globals.random_seed = seed

# Use Wine data set for training and test data

```



```

feature_dim = 2 # dimension of each data point
X_train = []
Y_train = []
_, training_input, test_input, _ = breast_cancer(training_size=12,
                                                  test_size=4, n=feature_dim)
instance = QuantumInstance(BasicAer.get_backend('statevector_simulator'),
                           shots=1024, seed_simulator=seed, seed_transpiler=seed)
feature_map = RawFeatureVector(feature_dimension=feature_dim)
start_time = time.time()
vqc = VQC(COBYLA(maxiter=100),
          feature_map,
          TwoLocal(feature_map.num_qubits, ['ry', 'rz'], 'cz', reps=3),
          training_input, test_input)
result = vqc.run(instance)

print('Testing accuracy: {:.2f}'.format(result['testing_accuracy']))
print(result)

print("--- %s seconds ---" % (time.time() - start_time))

```

В Код традиційного машинного навчання на наборі даних breast_cancer

```

import numpy as np
from sklearn.datasets import load_wine
from sklearn.svm import SVC
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris, load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.svm import LinearSVC, SVC
from sklearn.metrics import classification_report
from sklearn.model_selection import GridSearchCV
import time
import pandas as pd
# Load data
x,y = load_breast_cancer(return_X_y=True)

```

```
data = load_wine()
df = pd.DataFrame(data['data'], columns=data['feature_names'])
df['Target'] = data['target']
X = df.drop('Target', axis=1)
y = df['Target']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,
random_state=1376)

start_time = time.time()
svc = SVC(kernel="linear", C=0.025)
svc.n_iter_ = 100
svc.fit(X_train, y_train)
test_predictions = svc.predict(X_test)
train_predictions = svc.predict(X_train)
print("Train:")
print(classification_report(y_train, train_predictions))
print("Test:")
print(classification_report(y_test, test_predictions))
print("--- %s seconds ---" % (time.time() - start_time))
```