

TDK-dolgozat

Jósár Petra

Budapesti Corvinus Egyetem
Adatelemzés és Informatika Intézet
Számítástudományi Tanszék

Gépi tanulás és kvantuminformatika

Készítette: Jósár Petra
Gazdaságinformatikus BSc Szak (IV. évfolyam)
2023

Témavezető:
Dr. Fodor Szabina Eszter
egyetemi docens

Tartalomjegyzék

Ábrajegyzék	4
Táblázatjegyzék.....	5
Absztrakt	7
1. Bevezetés.....	9
2. Kvantuminformatika bemutatása	11
2.1. Kvantuminformatika alapjai	11
2.2. Néhány fontos kvantumfizikai fogalom	13
2.3. Logikai kapuk a kvantuminformatikában.....	15
2.4. Kvantumszámítógépek helyzete jelenleg és a főbb gyártók.....	17
2.5. Kvantuminformatika alkalmazási területei, jövő.....	18
3. A gépi tanulás bemutatása.....	21
3.1. Fajtái	22
3.2. Kvantum gépi tanulás	25
4. Kutatási kérdés és módszertan bemutatása	31
4.1. Eszközök bemutatása.....	31
4.2. Felhasznált algoritmusok.....	35
4.2.1. Kvantum neurális hálózat	35
4.2.2. Klasszikus gépi tanuló algoritmusok	37
4.3. Felhasznált adatok	41
5. Eredmények bemutatása.....	46
5.1. Az összehasonlításhoz használt mérőszámok.....	47
5.2. Kvantum neurális hálózat eredményei	51
5.3. Kiválasztott klasszikus gépi tanuló algoritmusok eredményei.....	56
5.4. Kiértékelés - Kvantum és klasszikus változatok összehasonlítása.....	63
6. Konklúzió.....	66
Irodalomjegyzék.....	68

Ábrajegyzék

1. ábra: A klasszikus bit és a qubit összehasonlítása (Forrás: Hussain, Z. (2016). Classical Bit Vs Qubit [Image]. ResearchGate. https://www.researchgate.net/figure/Figure-1-Classical-Bit-Vs-Qubit_fig2_308414229 saját fordítás).....	13
2. ábra: "Quantum Computing" kifejezésre való keresések alakulása az elmúlt 5 évben világszerte (Forrás: https://trends.google.com/trends/explore?date=today%205-y&q=Quantum%20Computing adatai alapján saját szerkesztés).....	19
3. ábra: Korreláció a módosított, bináris klasszifikációra alkalmas Iris adatkészletben (Forrás: saját szerkesztés)	43
4. ábra: Korreláció a Pima Indians Diabetes adatkészletben (Forrás: saját szerkesztés)	45
5. ábra: Konfúziós mátrix (Forrás: saját szerkesztés)	47

Táblázatjegyzék

1. táblázat: Az Iris adatkészlet módosított, bináris klasszifikációra alkalmas változatának összefoglaló statisztikája (Forrás: saját szerkesztés).....	42
2. táblázat: a Pima Indians Diabetes adatkészlet összefoglaló statisztikája (Forrás: saját szerkesztés).....	45
3. táblázat: Az Iris adatbázison futtatott kvantum neurális hálózat betanulási ideje, osztályozási ideje és pontossági jellemzői különböző n_epochs paraméter-beállítások mellett (Forrás: saját szerkesztés).....	52
4. táblázat: A Pima Indians Diabetes adatbázison futtatott kvantum neurális hálózat betanulási ideje, osztályozási ideje és pontossági jellemzői különböző n_epoch paraméter-beállítások mellett (Forrás: saját szerkesztés)	53
5. táblázat: Az Iris adatbázison futtatott kvantum neurális hálózat betanulási ideje, osztályozási ideje és pontossági jellemzői $n_epoch = 100$ paraméter-beállítás mellett (Forrás: saját szerkesztés)	55
6. táblázat: A Pima Indians Diabetes adatbázison futtatott kvantum neurális hálózat betanulási ideje, osztályozási ideje és pontossági jellemzői $n_epoch = 100$ paraméter-beállítások mellett (Forrás: saját szerkesztés).....	56
7. táblázat: A klasszikus SVM pontossági jellemzői, valamint a tanuláshoz és az osztályozáshoz szükséges idők az Iris adatbázis esetén (Forrás: saját szerkesztés)	57
8. táblázat: A klasszikus k-NN pontossági jellemzői, valamint a tanuláshoz és az osztályozáshoz szükséges idők az Iris adatbázis esetén (Forrás: saját szerkesztés)	58
9. táblázat: Az automatikus gépi tanulás segítségével létrehozott algoritmus-együttes pontossági jellemzői, valamint a tanuláshoz és az osztályozáshoz szükséges idők az Iris adatbázis esetén (Forrás: saját szerkesztés).....	59
10. táblázat: A klasszikus SVM pontossági jellemzői, valamint a tanuláshoz és az osztályozáshoz szükséges idők a Pima Indians Diabetes adatbázis esetén (Forrás: saját szerkesztés).....	60
11. táblázat: A klasszikus k-NN pontossági jellemzői, valamint a tanuláshoz és az osztályozáshoz szükséges idők a Pima Indians Diabetes adatbázis esetén (Forrás: saját szerkesztés).....	61

12. táblázat: Az automatikus gépi tanulás segítségével létrehozott algoritmus-együttes pontossági jellemzői, valamint a tanuláshoz és az osztályozáshoz szükséges idők a Pima Indians Diabetes adatbázis esetén (Forrás: saját szerkesztés).....	62
13. táblázat: Az Iris adatbázisra vonatkozóan az egyes modellek átlagos tanulási és osztályozási idői, valamint pontossági jellemzői (Forrás: saját szerkesztés).....	63
14. táblázat:A Pima Indians Diabetes adatbázisra vonatkozóan az egyes modellek átlagos tanulási és osztályozási idői, valamint pontossági jellemzői (Forrás: saját szerkesztés).....	64

Absztrakt

Napjainkban egyre többet hallhatunk a kvantuminformatikáról, amely a klasszikus informatikától teljesen eltérő, kvantummechanikai alapokon működve jelentős számítási előnyöket biztosít a hagyományos architektúrákkal szemben. Így az előzetes várakozások alapján számos, jelenleg nehezen és lassan megoldható probléma válik teljesíthetővé a kvantumalgoritmusok által.

Mivel az elmúlt években hatalmas fejlődés ment végbe a kvantuminformatika területén, különösen a hardverek oldalán, így egyre több cég (pl. IBM, Microsoft, Google) lát potenciált a kvantumszámítógépek építésében, hozzáférhetővé tételében. Egyes architektúrák már kereskedelmi forgalomban is kaphatók, azonban szélesebb elterjedésüket az ingyenes, felhő-alapú szolgáltatás formájában történő elérésük biztosítja. Ezekhez a rendszerekhez szabványos programozási nyelveket és fejlesztői környezeteket hoztak létre, így biztosítva bárki számára, hogy megismerkedhessen ezen új informatikai paradigmával és a benne rejlő lehetőségekkel. A kvantumalgoritmusok fejlesztése tehát egy kiterjedtebb kör számára vált elérhetővé, fő fókuszukban pedig jelenleg olyan témák állnak, mint a gépi tanulási és optimalizálási feladatok kutatása.

Jelen kutatásban arra kerestem a választ, hogy a mindenki számára elérhető, Python alapú szoftver-megoldások esetén megfigyelhető-e a technológiailag megjósolt kvantumfölény, vagyis az előzetes várakozásoknak megfelelően valóban jobban és gyorsabban teljesít-e egy kvantum megoldás a klasszikus megoldásokkal szemben. Ezt kvantum és klasszikus gépi tanuló algoritmusok pontosságának és teljesítményének összehasonlításával vizsgáltam meg. Publikusan elérhető adatbázisok felhasználásával bináris klasszifikációs problémára hasonlítottam össze egy, a szakirodalomban már publikált kvantum neurális hálózatot klasszikus Support Vector Machine és k-Nearest Neighbor algoritmusokkal, valamint egy szintén klasszikus alapon működő automatizált gépi tanulás segítségével összeválogatott algoritmus-együttessel. Az összehasonlításakor megvizsgáltam az osztályozás olyan pontossági jellemzőit, mint az accuracy, precision, recall, F1, AUC, valamint a tanuláshoz és osztályozáshoz szükséges időket. A kapott eredményeim nem igazolták a kvantum gépi tanuló algoritmus fölényét. A vizsgálat során azt tapasztaltam, hogy a pontossági jellemzők esetén a kvantum gépi tanuló algoritmus megközelítette a klasszikusokat, de a betanítás ideje lényegesen hosszabb volt. Kutatásom eredményei alátámasztják a szakirodalomból ismert véleményt, hogy az általam is használt hibrid megoldás - a kvantumosság hozzáadása a klasszikus gépi tanuláshoz - nem igazi megoldás. A valóban eredményes megoldásokhoz olyan

új algoritmusok kellenek, amelyek képesek a kvantumtulajdonságok kihasználása mellett hatékonyan elemezni az adatokat.

1. Bevezetés

A gépi tanulás jelenleg rendkívül népszerű témakör mind az üzlet, mind pedig az informatika világában. Emellett az elmúlt években hatalmas fejlődés ment végbe a kvantum-informatika területén is, különösképpen a hardverek oldalán, így egyre több óriásvállalat (pl. IBM, Microsoft, Google) lát jelentős potenciált a kvantumszámítógépekben. A klasszikus rendszereken alkalmazott gépi tanulás területén pillanatnyilag az egyik legjelentősebb kihívást az egyre növekvő adatmennyiség gyors, pontos feldolgozásához szükséges számítási idő és kapacitás korlát jelenti, amelyekre a kvantummechanikai elveket kihasználó kvantum-architektúrák várhatóan jelentős számítási előnyöket kínálhatnak a hagyományos megoldásokkal szemben. Ezen várakozás hatására a jelenlegi kvantumalgoritmus fejlesztéseknek meghatározó kutatási irányává vált a gépi tanulási feladatok kvantumos megvalósítása, valamint annak feltárása, hogy a gépi tanulás és a kvantum-informatika keresztezése milyen lehetőségeket rejthet magában.

A dolgozat célja a technológiailag megjósolt kvantumfölény megvalósulásának eldöntése kvantum és hagyományos gépi tanuló algoritmusok pontosságának és teljesítményének összehasonlításával. Mindenki számára szabadon hozzáférhető Python alapú szoftver-megoldások segítségével, publikusan elérhető adatbázisok felhasználásával bináris klasszifikációs problémára került összehasonlításra egy a szakirodalomban korábban már publikált kvantum neurális hálózat három klasszikus gépi tanuló megoldással. Az összehasonlítás számos pontossági jellemző, valamint a tanuláshoz és az osztályozáshoz szükséges idők figyelembevételével valósult meg.

A dolgozat első részében a kvantum-informatika, a gépi tanulás és a kvantum gépi tanulás elméleti háttere kerül bemutatásra. Mivel a kvantum-informatika a klasszikustól teljesen eltérő informatikai paradigma, ezért először a kvantum-informatika alapjai, néhány fontos kvantumfizikai fogalom, logikai kapu bemutatásáról lesz szó, majd egy rövid kitérő következik a kvantumszámítógépek aktuális helyzetére és a kvantum-informatika néhány alkalmazási területére. Az ezutóbbi kapcsán megemlített gépi tanulás kerül a továbbiakban bővebb kifejtésre, ide értve a gépi tanulás fajtáit, valamint a kvantum gépi tanulást, amely áttekintésekor az egyes megközelítések, a kvantumfölény, illetve a kvantum gépi tanulás néhány jelenlegi kihívása kerül ismertetésre.

Ezt követően a dolgozat második felében egy kvantum és három klasszikus gépi tanuló algoritmus pontosságának és teljesítményének összehasonlítására kerül sor a gyakorlatban, két különböző méretű, nyilvánosan elérhető adatbázis, az Iris és a Pima Indians Diabetes

felhasználásával bináris klasszifikációs problémára. Ezen gyakorlati részben először a kvantum és klasszikus gépi tanulás megvalósításához szükséges eszközök, a bináris klasszifikációs probléma megoldásához alkalmazott algoritmusok, majd a felhasznált adatok kerülnek ismertetésre. Ezután az összehasonlításához alkalmazott mérőszámokról és az összehasonlítás eredményeinek bemutatásáról lesz szó kvantum és klasszikus megoldásonként külön-külön, majd összevetve, egyben is. Végül az összehasonlítások eredménye alapján megválaszolásra kerül a dolgozat fő kérdése, hogy a jelen kutatás során is alkalmazott, szabadon elérhető Python alapú hibrid megoldások esetén valóban megfigyelhető-e a technológiailag megjósolt kvantumfölény a gépi tanulás területén.

2. Kvantuminformatika bemutatása

Az utóbbi években egyre nagyobb figyelem került a kvantuminformatika kutatására és fejlesztésére, egyre több óriásvállalat kezdett bele kvantumszámítógépek építésébe, felhasználásába, tette ezeket könnyen elérhetővé más fejlesztők számára, akár felhő-alapú szolgáltatások formájában, akár számos emulációs környezet kialakításának segítségével. Ezen kutatások és fejlesztések elterjedésének háttérében a kvantummegoldásokkal szemben támasztott előzetes várakozások, a kvantumfölény eléréséhez fűzött remények állnak. A kvantuminformatika ugyanis a klasszikustól teljesen eltérően, kvantumfizikai alapokon működik, amelynek köszönhetően a kvantummechanikai elvek kihasználásával jelentős előnyöket kínál a hagyományos megoldásokkal szemben. Ennek hatására az informatika számos területén várhatóan meg fog valósulni a kvantumfölény, azaz sok jelenleg a klasszikus megoldások által nem vagy csak nagyon nehezen, lassan megoldható probléma válik kivitelezhetővé a kvantuminformatika algoritmusai és architektúrái által. Potenciálisan egyike ezen területeknek a jelen dolgozatban is vizsgált gépi tanulás területe. Mielőtt azonban részletesebben bemutatásra kerülne a kvantuminformatika és a gépi tanulás metszetét jelentő kvantum gépi tanulás, valamint a kvantum gépi tanuló algoritmusok pontosságának és teljesítményének összehasonlítása a klasszikus megoldásokkal, érdemes kitérni a klasszikustól alapokban eltérő kvantuminformatika rövid ismertetésére, a kvantumfölény megvalósulásához elengedhetetlen kvantummechanikai tulajdonságok bemutatására és a kvantumszámítógépek jelenlegi helyzetére.

Ez a fejezet először kitér röviden a kvantuminformatika fogalmára, az alapjait képző kvantum bitekre – másnéven qubit-ekre, valamint a klasszikus számítógépekkel szemben előnyt biztosító, a kvantumfölény elérésére várakozást keltő néhány kvantummechanikai tulajdonságára, amelyek szintén kulcsfontosságúak ahhoz, hogy pontosan érthető legyen miben különbözik egy kvantummegoldás a jelenleg elterjedt klasszikus megoldásoktól. Ezt követően a kvantumszámítástechnikában létező fontosabb logikai kapukról, a kvantumszámítógépek jelenlegi helyzetéről, valamint a főbb gyártókról lesz szó. A fejezet végén pedig megvilágításra kerül, hogy mely (üzleti) területeken és hogyan alkalmazhatóak a kvantumszámítógépek, valamint várhatóan hogyan alakíthatják át a jövőt.

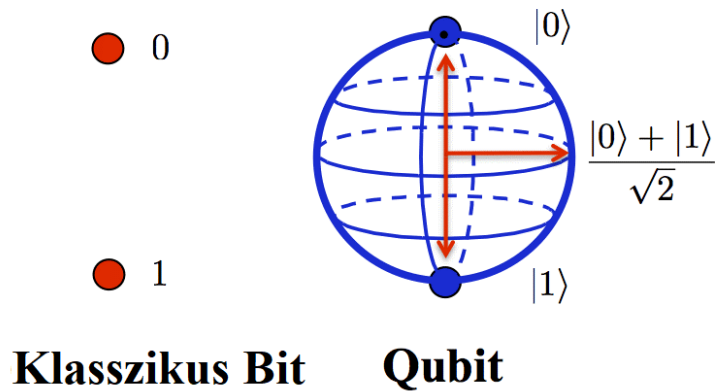
2.1. Kvantuminformatika alapjai

A kvantuminformatika az informatikának azon része, amely a klasszikus informatikától eltérően kvantummechanikai alapokon jött létre és ezen kvantummechanikai tulajdonságok

reményt nyújtanak arra, hogy bizonyos pillanatnyilag kihívást jelentő problémák megoldására – mint például az adattitkosítás területén fontos szerepet betöltő nagyszámok prímfelbontása – a jelenleg létező megoldásoknál gyorsabb és hatékonyabb algoritmusok készülhessenek el. A manapság elterjedt számítógépek – legyen szó asztali számítógépekről, laptopokról, tabletekről, okostelefonokról és egyéb okoseszközökről, vagy akár a felhő-alapú számítástechnikáról – alapja a determinisztikus bitek, amelyek két állapotot (0-át és 1-et) vehetnek fel, és egy adott időpillanatban a két állapot közül pontosan az egyikben vannak. Az atomi vagy a szubatomi részecskék esetén azonban nem a klasszikus, Newton által felállított fizika törvények érvényesek, hanem a kvantumfizika, ahol a részecskéknek számos új viselkedése figyelhető meg, mint például a szuperpozíció, a kvantum-összefonódás és a kvantuminterferencia. A kvantuminformatika ezeket az új viselkedéseket használja ki, és ezeket alkalmazza a számítások elvégzésénél (Dilmegani, 2022a; Williams, 2011).

Kvantuminformáció alapegysége: a qubit

Ahogy a klasszikus informatikában az információ alapegységét a bináris bit jelenti, úgy a kvantuminformatikában ez a kvantumbit, vagy qubit. Amíg a klasszikus bitek két állapotot vehetnek fel, 0-át vagy 1-et, és egy adott időpillanatban a két lehetséges állapot közül pontosan az egyikben vannak, addig a qubitek 0, 1, valamint 0 és 1 állapottal is jellemezhetők. Ennek az az oka, hogy a kvantumbitek egyik kvantummechanikai tulajdonságukat, a szuperpozíciót kihasználva a 0 és 1 állapotok egy tetszőleges lineáris kombinációját képesek felvenni, azaz állapotuk egy bizonyos valószínűséggel 0, egy bizonyos valószínűséggel pedig 1 lesz. Ezt a szuperpozíciót mindaddig képesek megtartani, amíg nem kerülnek konkrét mérések elvégzésre rajtuk (Knill, 2010; Microsoft, 2023a). Az 1. ábra bemutatja, hogy miben különbözik egy klasszikus, bináris bit és egy kvantumbit egymástól. (Annak érdekében, hogy ránézésre meg lehessen különböztetni a bináris és kvantumbiteket, a qubiteknél a 0, 1 állapotokat $|0\rangle$ és $|1\rangle$ jelöléssel szokták ellátni.):



1. ábra: A klasszikus bit és a qubit összehasonlítása

(Forrás: Hussain, Z. (2016). Classical Bit Vs Qubit [Image]. ResearchGate.

https://www.researchgate.net/figure/Figure-1-Classical-Bit-Vs-Qubit_fig2_308414229

saját fordítás)

A kvantumbitek továbbá még abban is különböznek a klasszikus számítógépek bitjeitől, hogy nem szilícium alapú chipekre épülnek, hanem általában mesterséges vagy természetes atomokból, befogott ionokból, fotonokból, vagy kvázirészecskékből készülnek el (Knill, 2010; Microsoft, 2023a).

2.2. Néhány fontos kvantumfizikai fogalom

A kvantumszámítógépek kvantumfizikai alapokon történő működésének köszönhetően a kvantumalgoritmusok olyan kvantummechanikai jelenségekre építhetnek, mint a szuperpozíció, a kvantum-összefonódás vagy az interferencia. Ezen képességük pedig bizonyos esetekben előnyt biztosíthat számukra a klasszikus megoldásokkal szemben, hiszen a szuperpozíció, a kvantum-összefonódás és az interferencia hármának köszönhetően a kvantum megoldások esetén olyan jelentős számítási kapacitás alakulhat ki, amely a klasszikus számítógépeknél exponenciálisan gyorsabb problémamegoldásra képes (Knill, 2010; Microsoft, 2023a). Ebben az alfejezetben ezen kvantumjelenségek – először a szuperpozíció, majd a kvantum-összefonódás, végül az interferencia – kerülnek bővebb kifejtésre.

Szuperpozíció

A kvantumfizikában előforduló szuperpozíció jelenségét legkönnyebben talán Schrödinger macskájának példáján keresztül lehet szemléltetni. A példa leegyszerűsítve, röviden a következő: egy macskát és egy méreggel teli üveget egy kívülről nem belátható dobozba zártak. A kérdés pedig az, hogy elpusztult-e a macska a méreggel teli üveg eltörésének

hatására, vagy még mindig él, mivel a méreggel teli üveg sértetlen maradt. Ahhoz, hogy kiderüljön, él-e még a macska, ki kell nyitni a dobozt, hiszen a doboz kinyitását megelőzően a macska az élő és holt állapot szuperpozíciójában van, vagyis egy bizonyos valószínűséggel él még, és egy bizonyos valószínűséggel elpusztult már. A mérést, vagyis jelen esetben a doboz kinyitását követően azonban megszűnik ez a szuperpozíció, hiszen ekkorra már egyértelművé válik, hogy a két állapot közül (élő és holt) végül melyikbe került a macska (Bala, 2020; Monroe, Meekhof, King, - Wineland, 1996). Ez alapján pedig amennyiben A és B két érvényes kvantumállapotát jelentik egy kvantumrészcskének, úgy ezen állapotok bármely lineáris kombinációja ugyanúgy érvényes kvantumállapotnak számít:

$$\text{kvantumrészcské állapota} = \alpha A + \beta B$$

ahol α és β az A és B kvantumállapotok valószínűségi amplitúdói és igaz rájuk, hogy $|\alpha|^2 + |\beta|^2 = 1$. A kvantumbitek szuperpozícióra való képessége nagy előnyt jelent a klasszikus bitek determinisztikusságával szemben, hiszen a szuperpozíciónak köszönhetően a kvantumbiteken alapuló rendszerek információmennyiséget ábrázoló képessége exponenciálisan növekszik. Ezt a következő példa szemlélteti igazán: míg két klasszikus bit esetén összesen 2^2 , azaz 4 lehetséges értékpáros kerülhet ábrázolásra (00, 01, 10, 11), de egyszerre a négy értékpáros közül csak egy, addig két kvantumbit esetében mindkettő felvehet 0-át, 1-et, valamint ezen állapotok szuperpozícióját is, amelynek eredményeképp a két qubit egymással szimultán ábrázolhatja a korábbi példa 4 lehetséges értékpárosát. Az ábrázolható információmennyiség szuperpozíció hatására történő exponenciális növekedésével szemben az információ feldolgozása viszont sokkal kevesebb idő alatt történik meg egy kvantumalgoritmus esetén, mint akármelyik, jelenleg leggyorsabbnak számító klasszikus rendszer alkalmazásakor (Microsoft, 2022; Wittek, 2014).

Kvantum-összefonódás

A kvantum-összefonódás egy olyan kvantummechanikai jelenség, amely során két vagy több kvantumbit összefonódik egymással. Ez az összefonódás tulajdonképpen egy kvantum korreláció a qubitek között. Ez azt jelenti, hogy egy, az összefonódásban résztvevő qubit állapota nem írható le az összefonódásban résztvevő, egyéb qubitek állapotától függetlenül, még akkor sem, ha nagy távolság választja el őket egymástól. Ennek eredményeképp, ha megváltoztatásra vagy megmérésre kerül egy az összefonódásban jelen lévő qubit állapota, akkor a vele összefonódott, más qubitek állapota is megváltozik, illetve közvetlen mérés nélkül is megismerhető lesz, még akkor is, ha egymástól végtelen távolságra vannak. Ez az

összefonódás jelenség az egyik legnagyobb különbség a klasszikus- és a kvantumfizika között, hiszen míg az előbbiben egyáltalán nem fordul elő, addig az utóbbinak az elsődleges jellemzői között tartható számon (Ganguly, 2021; Horodecki, Horodecki, Horodecki, - Horodecki, 2009; Microsoft, 2022, 2023a).

Interferencia

Mivel a szuperpozíció alapján mindegyik qubit állapotot egy, a hullámok amplitúdójához hasonló valószínűségi amplitúdó ír le (α és β), ezért a kvantumbitek állapotai interferenciát mutatnak egymással, és ez a kvantum-interferencia hat a qubitek állapotára oly módon, hogy mérés során befolyásolja egy bizonyos kimenet valószínűségét. Tehát az interferencia hatására dől el, hogy méréskor, azaz a szuperpozíció megszűnésekor melyik állapotba kerül a kvantumbit ($|0\rangle$ vagy $|1\rangle$) (Ganguly, 2021; Microsoft, 2022).

2.3. Logikai kapuk a kvantuminformatikában

Ahogy a klasszikus informatikában is számos logikai művelet, úgynevezett logikai kapuk találhatók, és ezekből különböző körök építhetők, úgy a kvantuminformatikában is rendelkezésre állnak kvantum logikai kapuk, amelyekből kvantumkörök (quantum circuit) építésére nyílik lehetőség. A kvantum logikai kapuk tehát egy vagy több kvantumbit transzformációját jelentik, a legtöbb klasszikus logikai kapuval szemben a kvantumváltozatok azonban megfordíthatók (Khan – Robles-Kelly, 2020). A továbbiakban néhány kvantum logikai kapu kerül röviden bemutatásra.

Identitás kapu

Az identitás kapu egyetlen kvantumbites kapu, amelynek segítségével megtartható a qubit eredeti értéke. Leginkább a klasszikus informatikában található pufferekhez hasonlítható, amelyek egy bit értékének tárolására szolgálnak abban az esetben, ha valami miatt kérés jelentkezik a körben (Bala, 2020; Ganguly, 2021).

$$|0\rangle \rightarrow |0\rangle$$

$$|1\rangle \rightarrow |1\rangle$$

Negált kapu (Pauli-féle X kapu)

Az identitás kapuhoz hasonlóan a negált kapu is egyetlen kvantumbites kapu. A negált kapu a bemeneti kvantumbit értékét az eredeti állapotának az ellenkezőjére változtatja (Bala, 2020; Ganguly, 2021).

$$|0\rangle \rightarrow |1\rangle$$

$$|1\rangle \rightarrow |0\rangle$$

Kontrollált negált kapu

A kontrollált negált kapu két qubites kapu, amely az egyik kvantumbit állapotával kontrollálja, irányítja a másik kvantumbit állapotát. Az irányítás a következőképpen történik: amennyiben az irányító qubit állapota $|0\rangle$, akkor a bemeneti qubit állapota változatlanul marad, azonban abban az esetben, amikor az irányító qubit állapota $|1\rangle$, akkor a bemeneti qubit állapota megfordításra kerül (Bala, 2020; Ganguly, 2021; Monroe, Meekhof, King, Itano, - Wineland, 1995).

$$\text{ha } |0\rangle \text{ az irányító qubit értéke: } |0\rangle, |1\rangle \rightarrow |0\rangle, |1\rangle$$

$$\text{ha } |1\rangle \text{ az irányító qubit értéke: } |1\rangle, |1\rangle \rightarrow |1\rangle, |0\rangle$$

A kontrollált negált kaput a legtöbb esetben a bemeneti kvantumbitek közül a második kvantumbit állapotának mérésére használják fel (Khan – Robles-Kelly, 2020).

Hadamard kapu

A Hadamard kapu a $|0\rangle$ és $|1\rangle$ kvantumállapotok szuperpozíciójának létrehozásáért felel, azaz ez a kapu függetlenül attól, hogy a bemeneti qubit eredeti állapota $|0\rangle$ vagy $|1\rangle$ volt, a $|0\rangle$ és $|1\rangle$ állapotok egyenletes eloszlású, azaz 0,5 valószínűséggel $|0\rangle$, és szintén 0,5 valószínűséggel $|1\rangle$ állapotú szuperpozícióját teremti meg. A Hadamard kapu egyetlen bemeneti qubittel működik (Bala, 2020; Ganguly, 2021; Salehian, - Mohammadne, 2008).

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Egyéb kvantum logikai kapuk

A már említett négy kvantumkapun kívül még olyan más kvantum logikai kapuk is léteznek, mint például a Pauli-féle Y és Z kapuk, amelyek a kvantumbitek y és z tengelyei körüli forgatását végzik el, a Toffoli kapu, a SWAP kapu, a Deutsch kapu, és még számos más (Roell, 2018). A jelenlegi dolgozatnak azonban nem célja ezen egyéb kvantumkapuk bővebb kifejtése, bemutatása.

2.4. Kvantumszámítógépek helyzete jelenleg és a főbb gyártók

Kvantumszámítógépek építésébe és fejlesztésébe egyelőre főleg csak néhány nemzetközi óriásvállalat, valamint kiemelten magas szakértelemmel bíró startup vállalkozás kezdett. Ennek az a fő oka, hogy a mostani kvantumszámítógépek még nem képesek a stabil működésre, sok meghibásodás jelentkezik az üzemeltetésük során, mivel a jelenleg legelterjedtebb technológiák olyan speciális körülményeket igényelnek a kvantumszámítógépek működtetéséhez, mint a kifejezetten alacsony, abszolút nulla körüli hőmérséklet - hogy a túlmelegedés elkerülhető legyen - ez pedig speciális berendezések szükségességét vonja maga után. Ennek következményeként kiemelten korlátozottá vált a kvantumszámítógépek szélesebb körben történő elterjedése, valamint hozzáférhetővé tétele. Ezért az utóbbi években jelentős kezdeményezések indultak a kvantumszámítógépek működtetésének olcsóbbá, kevésbé infrastruktúra-igényesebbé, és szélesebb körben hozzáférhetővé tételére, valamint arra, hogy szobahőmérsékleten is üzemképesé válhassanak. Ezutóbbi igényre elsőként a kanadai Xanadu startup vállalkozás válaszolt, amikor megalkotta az első, szupravezetők helyett a fényt felhasználó, fotonika-alapú kvantumszámítógép platformját, amely már lehetővé teszi a kvantumszámítógépek szobahőmérsékleten történő működtetését (Ménard, Ostojic, Patel, - Volz, 2020; Partida, 2022).

Ami pedig a hozzáférhetőséget illeti, elsőként az IBM vezette be 2016-ban a kvantumszámítógépeihez hozzáférést biztosító felhő-alapú szolgáltatását, amely bárki számára lehetővé teszi virtuálisan egy kvantumszámítógép használatát. Ezáltal önkéntes fejlesztők sokasága kezdett el kvantumszámítógépeken dolgozni, amely az egész technológia állandó, további fejlődését szolgálja. A felhő-alapú szolgáltatás így azóta domináns üzleti modellé vált nemcsak az IBM, hanem más cégek kvantumszámítógépei esetén is (Partida, 2022).

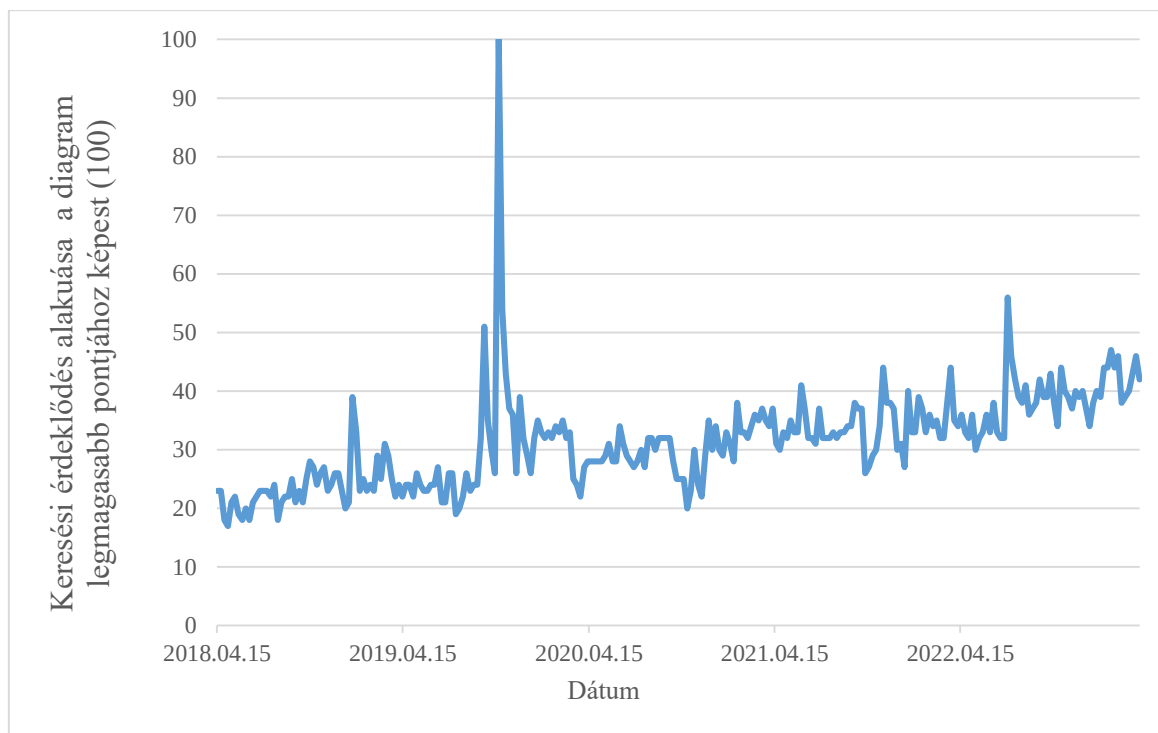
A kvantuminformatika területén mindenképpen érdemes az olyan óriások nevét is megemlíteni, mint a Google vagy a Microsoft, az előbbi főként 2019-ben ért el nagy áttörést a szakterületen, amikor kvantumfölényt jelentett be a szuperszámítógépek felett, míg az utóbbi a Microsoft Azure Quantum segítségével biztosít mindenki számára hozzáférést

kvantumszámítógépeihez, ezen kívül pedig kiterjedt dokumentációval segíti a technológia megismerését. A kevésbé ismert, de a kvantuminformatikában élen járó vállalatok közé tartozó D-Wave Systems, és Quantum Computing Inc. (QCI) szintén jelentős szerepet töltenek be a szakterületen (Ménard, Ostojic, Patel, - Volz, 2020; Partida, 2022; Srivastava, Choi, Cook, - Team, 2016).

2.5. Kvantuminformatika alkalmazási területei, jövő

A kvantuminformatika alkalmazási területei és jövője kapcsán elsőként azt érdemes kiemelni, hogy a kvantuminformatika, és ezen belül a kvantumszámítógépek nem azért jöttek létre, és foglalkozik egyre több vállalat, valamint szakértő a fejlesztésükkel, minél stabilabbá tételükkel, hogy ezáltal a jelenlegi klasszikus számítógépek kiváltásra kerüljenek. A jövőt sokkal inkább úgy lehetne elképzelni, hogy a kvantum- és klasszikus alapokon működő számítógépek párhuzamosan, saját erősségeiket kihasználva működnének egymás mellett (Ménard, Ostojic, Patel, - Volz, 2020).

Ahogy a fejezeten belül többször is megvilágításra került már, a kvantumszámítógépek exponenciálisan növekvő információmennyiség ábrázolásra képesek, ennek eredményeképp pedig olyan nagyméretű, komplex problémák gyors megoldására is lehetőséget nyújtanak, amelyek a legjobb jelenlegi szuperszámítógépeknek szinte lehetetlenek, vagy legalábbis rengeteg időbe telnének. Ezt támasztotta alá a Google 2019-es bejelentése is, amelyben kijelentették a kvantumfölényt, hiszen 2019 októberében a Google 54 kvantumbites Sycamore processzora 200 másodperc alatt képes volt elvégezni olyan számítást (ez magába foglalta például véletlenszámok generálását is), amely a világ leggyorsabb, klasszikus alapokon működő szuperszámítógépének a Google szerint körülbelül 10.000 évébe telt volna (Arute et al., 2019). Ennek és egyéb fejlesztések hatására az utóbbi években élénkebb lett a figyelem a kvantuminformatika témaköre iránt, ezt a 2. ábra is mutatja, amely egy, a témakörre irányuló Google kereséseket szemléltető Google Trends ábra:



2. ábra: "Quantum Computing" kifejezésre való keresések alakulása az elmúlt 5 évben világszerte
(Forrás: <https://trends.google.com/trends/explore?date=today%205-y&q=Quantum%20Computing>
adatai alapján saját szerkesztés)

A grafikon azt ábrázolja, hogy az elmúlt 5 évben, világszerte hogyan alakult Google-ön a keresési érdeklődés a „Quantum Computing”, vagyis kvantuminformatika, kvantumszámítástechnika kifejezés iránt a grafikon 100-al jelölt, legmagasabb pontjához, azaz 2019.10.20-hoz képest. Látható, hogy a Google 2019-es bejelentését követően a témakörre keresések száma folyamatosan növekedett.

Az elmúlt néhány évben az üzleti világ is kezd egyre nagyobb figyelmet fordítani a kvantuminformatika által kínált lehetőségekre, és egyre több olyan terület nevezhető meg, amelybe a kvantumszámítógépek jelentős előnyöket, sikereket érhetnek el akár már a közeljövőben is. Ilyen például az optimalizáció, legyen szó akár a logisztika területén az út- és forgalom-, ellátási lánc, és raktárkészlet optimalizálásáról, akár a pénzügyi területen belül a portfóliók optimalizálásáról, vagy a gyártás során különböző modellek, design-ok optimalizálásáról. A pénzügyi szektort azonban nemcsak a portfóliók optimalizálásával segítheti a kvantuminformatika, hanem még olyan egyéb esetekben is, mint a csalásfelderítés, a kockázatok elemzése, pénzügyi piacokhoz kapcsolódóan különböző előrejelzések elkészítése, valamint a szintén ezen a területen történő kereskedés automatizálása révén (Dilmegani, 2022b; Ménard, Ostojic, Patel, - Volz, 2020).

A kvantuminformatika továbbá az energia szektor és a gyógyszeripar, orvostudomány számára is jelentős potenciálokat rejthet, hiszen míg előbbiben az energia kihasználásokra képes különböző predikciókat nyújtani, valamint pontosabb időjárás-előrejelzés érhető el a kvantumszámítógépeknek köszönhetően, addig a gyógyszeripar és orvoslás esetén olyan áttöréseket eredményezhet, mint a személyre szabott kezelések létrejötte a beteg genomikájának figyelembevételével és a különféle gyógyszerköölcsönhatások előrejelzése. Ezen kívül a jövőben a kvantumszámítógépeken végzett szimulációk segítségével akár számottevő mértékben is kiválthatók lesznek a kutatások során a laboratóriumokban végzett kémiai és biológiai kísérletek, ezzel jelentősen lecsökkentve egy-egy új gyógyszer tesztelési idejét és költségeit. Emellett a kvantumszámítógépek működési alapjainak köszönhetően olyan molekuláris szintű szimulációk elvégzésére is lehetőség nyílik, amelyek a klasszikus számítógépek számára lehetetlennek tűnnek. Ezen molekuláris szimulációk pedig további innovációkhoz vezethetnek a biológia, fizika, kémia, a különböző anyagtudományok, és persze az egészségügy területén is, amely a gyakorlatban például hosszú élettartammal rendelkező akkumulátorok, magas hőmérsékletet nem igénylő katalizátorok, szobahőmérsékleten üzemelő szupravezetők megjelenését, illetve az orvoslás terén a különböző betegségek adott egyén szervezetére gyakorolt konkrét hatásainak feltárása során alkalmazható (Dilmegani, 2022b; Ménard, Ostojic, Patel, - Volz, 2020).

A kvantuminformatika alkalmazási területeinél kiemelten fontos továbbá megemlíteni a kriptográfiát és a kémkedést is. A kvantumszámítógépek már bizonyítottan képesek lehetnek a jövőben (például a Shor algoritmus alkalmazásával) a jelenleg legelterjedtebb, legbiztonságosabbnak vélt titkosítási algoritmusok, köztük az aszimmetrikus, nyílt-kulcsú RSA eljárás a feltörésére is (Bhatia – Ramkumar, 2020), ezzel arra ösztönözve a világ legbefolyásosabb óriásvállalatait és kormányait, hogy belekezdjenek kvantumkész titkosítási eljárások kifejlesztésébe, így hozzájárulva a kvantuminformatika további fejlődéséhez. Ezen kívül, a jelszó feltörések és a különböző kibertámadások bekövetkezését is megkönnyíthetik a kvantumszámítógépek (Mosca, 2018), hiszen a klasszikus számítógépeknél sokkal alkalmasabbnak bizonyulnak olyan keresési problémák megoldásában, amelyeknél egy alapvetően nem kereshető struktúrában történik a keresés, vagy olyan esetekben, ahol legalább annyiféle megoldás születhet az adott problémára, amennyi bemenettel rendelkezett az eredetileg (Dilmegani, 2022b).

Végül pedig fontos megemlíteni a jelen dolgozat fókuszában álló gépi tanulást is, mint a kvantuminformatika egyik jelentős alkalmazási területét, hiszen a kvantumszámítási kapacitás révén szignifikánsan felgyorsulhat az összetettebb, hatalmas számítások elvégzését

igénylő gépi tanulási algoritmusok betanulási folyamata (Dilmegani, 2022b). Azokat a gépi tanuló algoritmusokat, amelyek kvantumszámítógépeken, vagy azok egy emulációs környezetén futnak, a szakirodalom kvantum gépi tanuló algoritmusoknak nevezi, a területet pedig kvantum gépi tanulásnak, röviden QML-nek (Quantum Machine Learning). Ezekről bővebben a „3.2. Kvantum gépi tanulás” című fejezetben lesz szó.

3. A gépi tanulás bemutatása

Az elmúlt néhány évben egyre népszerűbb lett a gépi tanulás témaköre, az üzleti-, kormányzati- vagy akár az oktatási szféra is egyre szélesebb körben kezdett bele ezen terület kutatásába, fejlesztésébe és felhasználásába. Sokan azonban hajlamosak összekeverni a gépi tanulást a mesterséges intelligencia fogalmával, és tévesen azonosnak gondolni a két területet, pedig valójában a gépi tanulás a mesterséges intelligencián belül egy terület, tehát a mesterséges intelligencia részhalmaza, valamint erősen épít a statisztikára is. Már az 1950-es években elkezdtek foglalkozni a gépi tanulással, ekkor született meg a terület talán legismertebb definíciója Arthur Samuel megfogalmazásában, amely alapján a gépi tanulás egy olyan tudományterület, ami anélkül biztosítja a számítógépek számára a tanulás képességét, hogy kifejezetten erre programozták volna őket (Schuld, Sinayskiy, – Petruccione, 2015). Igazán jelentős eredményeket azonban csak az utóbbi időben kezdett el elérni a gépi tanulás, hiszen eddigre sikerült olyan hardverelemeket létrehozni – erősebb CPU-kat és GPU-kat – amelyek képesek a gépi tanuló algoritmusok sokszor kiemelten nagy számítási kapacitásigényeinek megfelelni, valamint annyi adatot gyűjteni és tárolni, amely egy gépi tanuló algoritmus megfelelő működéséhez szükséges. A gépi tanuló algoritmusok ugyanis az adatokban mintákat azonosítanak, adatmodelleket, valamint jövőbeni adatokra vonatkozóan előrejelzéseket készítenek. Ahogy az emberek is a tanulási folyamataik során minél többet gyakorolnak, annál ügyesebbek lesznek egy adott területen, úgy a gépi tanuló algoritmusok eredményei és előrejelzései is egyre pontosabbá válnak a tanulás során a rendelkezésükre álló adatok és tapasztalatok mennyiségének növekedésével (Microsoft, 2023b).

A gépi tanulás esetében a tanulás három fő fajtája különböztethető meg: a felügyelt, a felügyelet nélküli, valamint a megerősítéses tanulás. A fejezet további részében ezen tanulási fajták rövid bemutatására kerül sor, végül pedig a kvantum gépi tanulás, azaz a gépi tanuló algoritmusok kvantumszámítógépeken történő felhasználásáról, illetve kvantumváltozataikról lesz szó.

3.1. Fajtái

Ebben az alfejezetben a gépi tanulás fő fajtái kerülnek bemutatásra. Először a felügyelt és a felügyelet nélküli, majd a megerősítéses tanulásról lesz szó. Egyes tanulási fajták esetén megemlítsre kerülnek az adott fajtán belül legnépszerűbbnek számító algoritmusok is, valamint az egyes gépi tanulási fajták néhány jellemző alkalmazási területei.

Felügyelt tanulás

A felügyelt tanulás (supervised learning) esetén a gépi tanuló algoritmus olyan adatkészleten tanul, amelyben az adatok címkézettek, tehát a célváltozójuk alapján egy konkrét kérdés megválaszolható. A tanuló adatkészlet egyik attribútuma tartalmazza minden egyed esetén a keresett végeredményt például, hogy egy adott egyed beteg-e vagy sem, egy e-mail a spam kategóriába sorolható-e stb. A gépi tanuló algoritmus úgy tud tanulni az adatokból, hogy minden egyednél az attribútumok (inputok) és a célváltozók (outputok) sorozatát látja, így megtanulva, hogy milyen attribútumokhoz általában a célváltozónak mely értéke, vagyis milyen output tartozik. A felügyelt tanulásnak az a fő célja, hogy a tanuló adatkészleten megtanult összefüggéseket felhasználva a modell képes legyen megbecsülni egy új, korábban a tanulás során általa még nem látott egyed esetén, hogy az adott egyednél a célváltozó melyik értéket veheti fel (Ganguly, 2021; Vas, 2020a).

A felügyelt tanulásnak két fő típusa van: az osztályozás, másnéven klasszifikáció, és a regresszió. Osztályozásról akkor lehet szó, hogyha a célváltozó kategorikus változó, például lehet bináris: igen/nem döntés, a két osztályból melyikbe sorolható az adott egyed, de lehet többosztályos is a klasszifikáció ha több, mint két értéket vehet fel a célváltozó. A másik fő típus, a regresszió akkor fordul elő, hogyha a célváltozó folytonos változó, ilyen például a jövedelmek becslése vagy a különböző árfolyamok előrejelzése. A regresszió belül lineáris és nem lineáris regresszió különíthető el. Azonban fontos megjegyezni, hogy a felügyelt tanulás ezen két fő típusa nem mindig választható el élesen egymástól, vannak olyan esetek, amikor a két típus keveredik egymással. Ilyen eset például, ha klasszifikáció esetén az egyes osztályokba tartozás helyett az egyes osztályokba való tartozás valószínűségének keresésére irányul a becslés, vagy ha regresszióként a folytonos változó értékének becslése helyett kategóriák kialakítása a cél, valamint arra becslést adni, hogy a folytonos változó a kialakított kategóriák melyikébe tartozik (például jövedelem kategóriák becslése). A felügyelt gépi tanulásnál klasszifikáció esetén olyan gépi tanuló algoritmusok a legelterjedtebbek, mint a Döntési fa, a Naive-Bayes, a k-Nearest Neighbor (k-NN), a Support Vector Machine (SVM), regresszió esetén pedig a Support Vector Regression (SVR), Stochastic Gradient Descent Regressor

(SGDRegressor), regressziós döntési fák vagy a regressziós k-Nearest Neighbor (regressziós k-NN) (Ganguly, 2021; Vas, 2020b).

A felügyelt gépi tanuló algoritmus betanítása előtt érdemes szétválasztani az adatkészletet tanuló és teszt adatokra (ezt általában 70%-30%, 75%-25% vagy 80%-20%-os arányban szokták megtenni), hogy megfelelően értékelhetők legyenek a betanított gépi tanuló modell eredményei, és elkerülhetővé váljon a gépi tanulás során egyik leggyakrabban előforduló hiba, a túltanulás. Túltanuláskor a modell nem képes megfelelően általánosítani, hanem olyannyira „bemagolja”, „rátanul” a betanulás során kapott adatokra, hogy egy teljesen új, a modell által korábban még nem látott egyed esetén jelentősen lecsökken a modell osztályozóképességének, előrejelzésének a minősége a betanuláskor alkalmazott adatokra adott becslésekhez képest (Vas, 2020b).

A kutatásban felügyelt tanulás került alkalmazásra. Az osztályozásra alkalmas, a kutatás során felhasznált kvantum és klasszikus gépi tanuló algoritmusok, valamint néhány paraméter-beállításuk bővebben a „4. Kutatási kérdés és módszertan bemutatása” című fejezetben kerülnek kifejtésre.

Felügyelet nélküli tanulás

Felügyelet nélküli (unsupervised learning) esetén címkékkel nem rendelkező adatkészleten tanul a gépi tanuló algoritmus, és a tanulás során az a cél, hogy a gépi tanuló algoritmus önállóan, mindenféle külső felügyelet nélkül megtalálja a természetes struktúrát, mintázatokat az adatokban, valamint megtanulja egymáshoz viszonyítani, szétválogatni, illetve összevetni a nem címkézett célváltozóval rendelkező példákat. Felügyelet nélküli tanulás esetén a gépi tanuló algoritmus a teljes adatkészlettel rendelkezik betanuláskor, tehát nincs szükség az adatkészlet tanuló és teszt adatokra való szétbontására. A felügyelet nélküli tanulás a következő fő típusokkal rendelkezik: klaszterezés, anomáliakeresés, illetve dimenzió csökkentés, másnéven információtömörítés (Ganguly, 2021; Ghahramani, 2004).

Klaszterezés esetén a gépi tanuló algoritmus az adatpontokat csoportosítja, klaszterekbe sorolja hasonlóságuk alapján. A csoportosítás során az a cél, hogy az egy klaszterbe tartozó adatpontok a lehető legjobban hasonlítsanak egymáshoz, míg a különböző klaszterekben lévő adatpontok a lehető legkülönbözőbbek legyenek. A klaszterezés leggyakrabban ügyfél- vagy egyéb szegmentációs problémák megoldására alkalmazott (Ganguly, 2021; Vas, 2020a).

Anomáliakeresés során a gépi tanuló algoritmus azt keresi, hogy a többség által létrehozott szerkezetbe mely adatok nem illeszkednek bele, melyek a jelentősen kilógó adatok. Az anomáliakeresésnek rendkívül jelentős szerepe van a kiberbiztonság, valamint a

csalásfelderítés területén (például bank- és hitelkártya csalások), ugyanis segítségével könnyen és gyorsan kiszűrhetők abnormális tevékenységek, tranzakciók a gépi tanuló algoritmus által már ismert normális tevékenységek, tranzakciók közé be nem illeszkedésük miatt (Ganguly, 2021; Vas, 2020a).

Dimenzió csökkentésre, vagy információtömörítésre akkor kerül sor, hogyha az adatkészlet rengeteg (akár több száz) input változót (attribútumot) is tartalmaz és az a cél, hogy az adatkészlet dimenziója (input változóinak a száma) lecsökkenjen. Ezen dimenzió csökkentés az adatok varianciájának nagyrészt megragadó jellemzők kinyerésével történik meg, ezáltal kiszűrve azokat az input változókat, amelyek a leginkább leírják az adatokat (Vas, 2020a).

Nem felügyelt tanulásnál klaszterezés esetén olyan gépi tanuló algoritmusok használata a legelterjedtebb, mint a hierarchikus klaszterezés vagy a k-mean algoritmus, dimenzió csökkentés esetén pedig a főkomponens-analízis (Principal Component Analysis) (Ganguly, 2021; Ghahramani, 2004). A felügyelet nélküli tanulást alkalmazó módszerek használata olyan kihívásokat rejthet magában, mint az eredmények pontosságának nehéz ellenőrizhetősége, hiszen nincsen címkével ellátott, kitüntetett célváltozó az adatkészletben, vagy a kijött eredmények értelmezéséhez sok üzleti (domain) tudás szükségessége (Vas, 2020a).

Megerősítéses tanulás

A megerősítéses tanulás (reinforcement learning) a gépi tanulási fajták közül a legjobban az emberi tanulásra és az emberek a tanulás folyamatáról alkotott elképzelésére hasonlító megközelítés. Megerősítéses tanulás esetén ugyanis a gépi tanuló algoritmus megtanulja, hogy milyen cselekvéseket kell elvégeznie, hogyan szükséges viselkednie egy adott szituációban, ezt pedig nem előre meghatározott módon teszi, hanem úgy, hogy folyamatosan kapcsolatban áll a környezetével, sorozatosan döntéseket hoz a cselekedeteiről és ezért a környezetétől jutalmat vagy büntetést kap. A megerősítéses gépi tanuló algoritmus fő célja a környezetétől hosszú távon összegyűjtött jutalmak maximalizálása. A tanuló algoritmusnak tehát nem kerül előre megadásra, hogy melyik cselekvéseket válassza, hanem helyette ő végez el több, különböző cselekvést is és ezek közül kiválasztja azt, amelyikre válaszul a környezetétől a legnagyobb jutalmat kapta. Míg más gépi tanulási fajta esetén nem jellemző az idő dimenzióra való kiterjedés, addig megerősítéses tanulás esetén az adatok idő-dimenzióval is rendelkeznek, ez pedig azt jelenti, hogy a tanuló algoritmus által hozott döntéseket a megelőző időpillanatban hozott döntések és a környezet reakciója ezen döntésekre is befolyásolják. Másszóval egy adott időpillanatban hozott döntés nem csak az aktuális szituációt befolyásolhatja, de a következő szituációkat, ezáltal pedig a jövőbeni döntéseket is. A megerősítéses tanulásnak tehát a

próbálkozás-visszajelzés (trial-and-error) és a késleltetett, hosszútávú jutalom (delayed reward) a két fő megkülönböztető tulajdonsága a többi gépi tanulási fajtával szemben. Megerősítéses tanulás leggyakrabban az automatizálás, az önvezető járművek és a robotika területén kerül alkalmazásra (Sutton – Barto, 2018).

3.2. Kvantum gépi tanulás

A kvantum gépi tanulás a jelenlegi kutatások két izgalmas területének, a gépi tanulásnak és a kvantuminformatikának a metszetét jelenti. A kvantum gépi tanulás (QML) olyan megközelítéseket keres és alkalmaz, amelyek segítségével szinergia fedezhető fel az említett két terület között. Ezen szinergia célja az, hogy az egyik terület problémáira a másik terület megoldásokat kínáljon (Alchieri, Badalotti, Bonardi, - Bianco, 2021). Ennek értelmében a kvantum gépi tanulás azt vizsgálja, hogy a kvantum kutatások és a kvantuminformatika területén elért eredmények hogyan segíthetnék elő egyes gépi tanulással kapcsolatos problémák megoldását, valamint azt, hogy hogyan hozható létre és implementálható olyan kvantumszoftver, amely lehetővé teszi a klasszikus változatnál gyorsabb gépi tanulás létrejöttét. A kvantum gépi tanulás területe számos jelentős potenciállal rendelkezik, hiszen a kvantumszámítógépek sokkal nagyobb számítási kapacitással bírnak, mint a klasszikus változatok, emiatt pedig jónéhány, a klasszikus változatokkal nem vagy csak nagyon lassan megoldható, hatalmas adatmennyiség feldolgozását igénylő gépi tanuló probléma válik kivitelezhetővé. Amíg a klasszikus gépi tanulás során az a cél, hogy mintázatok, összefüggések kerüljenek feltárára az adatokból és ezen megtanult összefüggések alapján becsléseket lehessen tenni korábban még nem látott, ismeretlen adatokra vonatkozóan, addig a klasszikus változatokkal szemben a kvantum rendszerek atipikus mintázatok hatékony előállítására, felfedezésére is képesek, ezzel azt a feltételezést erősítve, hogy gépi tanuló feladatok esetében a kvantumszámítógépek megelőzhetik a hagyományosakat (Biamonte et al., 2017; Mishra et al., 2021).

Jelen alfejezet először kitér a kvantum gépi tanulás témakörében előforduló főbb megközelítésekre, valamint ezek kategóriáira, azonban konkrét kvantum gépi tanuló algoritmusok nem kerülnek bemutatásra, ezt követően a nagy reményekkel kecsegtető kvantumfölénnyről és annak leglényegesebb dimenzióiról lesz szó, végül pedig röviden összefoglalásra kerül, hogy jelenleg mik okozzák a legjelentősebb kihívásokat a kvantum gépi tanulás területén.

Megközelítések

A kvantum gépi tanuló algoritmusok kapcsán alapvetően három különböző megközelítést érdemes megemlíteni: a tisztán klasszikus megoldásokat, a hibrid megoldásokat és a tisztán kvantum megoldásokat. Ezen három különböző megközelítés az alapján alakult ki, hogy a gépi tanulás során felhasznált adatok hogyan jöttek létre és kerültek feldolgozásra: az adatok kvantum vagy klasszikus rendszerek által generáltak lehetnek, valamint a feldolgozásukhoz használt eszköz szintén lehet klasszikus vagy kvantum alapokon működő. Mivel a tisztán klasszikus és tisztán kvantum megoldásokon belül egy-egy, a hibrid megoldásokon belül pedig két kategória különíthető el, ezért összesen négy különböző kategória említhető meg a gépi tanulás és kvantumosság egyesítésével kapcsolatban (Aïmeur, Brassard, - Gambs, 2006; Alchieri, Badalotti, Bonardi, - Bianco, 2021). A következőkben először röviden szó lesz a tisztán klasszikus megoldásokról, majd egy általános leírás következik a hibrid megoldásokról és az ezen megközelítésen belül elkülöníthető két kategóriáról, végül pedig a tisztán kvantum megoldások kerülnek bemutatásra.

A tisztán klasszikus megoldások esetén a hagyományos gépi tanulási megközelítés értendő, tehát az, amikor klasszikusan létrehozott adat klasszikus módon kerül feldolgozásra, azonban kvantum kutatásokból átemelt módszerek segítségével. Ezt a megközelítést a szakirodalom kvantumszerű gépi tanulásnak definiálja, hiszen a modelleket és optimalizációjukat gyakran olyan kvantumtulajdonságok, jelenségek inspirálják, mint a szuperpozíció vagy a kvantum-összefonódás (Alchieri, Badalotti, Bonardi, - Bianco, 2021; Wittek, 2014).

Ahogy korábban említésre került, a hibrid megoldásokon belül két kategória különül el. Az első kategóriában az adatokat előállító rendszer kvantum alapokon működik, azonban klasszikus eszközök segítségével kerülnek az adatok feldolgozására (QC eset). Ez az eset a leggyakrabban annak a megvizsgálására alkalmazható, hogy hogyan tud a klasszikus gépi tanulás segíteni a kvantumszámítások javításában. Például ha az a cél, hogy a lehető legkevesebb mérés elvégzésével lehessen meghatározni egy kvantumszámítógép belső állapotát, akkor ebben számos gépi tanuló algoritmus nyújthat segítséget (Aaronson, 2007). A hibrid megoldások második kategóriájában a klasszikus adatok kvantumszámítógép segítségével kerülnek feldolgozásra (CQ eset). Ezen eset két további típusra bontható aszerint, hogy egy már meglévő modellt használnak-e fel (translational approach) vagy egy teljesen új modellt hoznak létre (exploratory approach). Meglévő modell felhasználása esetén egy klasszikus gépi tanuló modell kvantum környezetbe történő átvitele valósul meg. Ez azt jelenti, hogy a klasszikus modellnek az általános logikája nem módosul, azonban a modell legalább

egy része valamilyen célból - például gyorsabb megoldhatóság - kvantumszámítógépen kerül végrehajtásra. Ezen már meglévő modelleket felhasználó típusok általában klasszikus, elterjedt algebrai algoritmusokon alapulnak, így a kvantumszámítást főként csak felhasználják, de nem eredményeznek teljesen új logikára építő megoldásokat. A teljesen új modellt létrehozó típus azonban sokkal szélesebb körben használja ki a kvantuminformatika nyújtotta lehetőségeket. Ezen típus fő célja, hogy egy adatokból tanulni képes modellt hozzon létre bármely a kvantummechanika törvényeinek eleget tevő, ezáltal pedig a kvantumtulajdonságok előnyeinek kihasználására képes rendszer alkalmazásával. Ennek köszönhetően a teljesen új modellt létrehozó típusok olyan új, innovatív megoldások létrehozását is támogatják, amelyeknek nem feltétlenül rendelkeznek klasszikus megfelelővel (Alchieri, Badalotti, Bonardi, - Bianco, 2021).

A tisztán kvantum megoldások esetén kvantumszámítógépek segítségével kvantum adatok kerülnek feldolgozásra. Ez kétféleképpen érhető el. Az első lehetőség az, ha a kvantumszámítógép egy kvantum rendszer dinamikájának szimulálására kerül felhasználásra. Ekkor a kvantum gépi tanuló modell bemenete maga az állapot is lehet, ezáltal lehetővé téve a kvantumszámítógép számára az információkhoz történő gyors hozzáférést. A második lehetőség pedig az, ha az adat már eleve egy fizikai kísérlet során előálló kvantum rendszeren végzett mérésekből származtatható, majd az így kapott értékek egy teljesen különálló kvantum eszköznek kerülnek átadásra (Alchieri, Badalotti, Bonardi, - Bianco, 2021).

Kvantumfölény

Ahogy az előző fejezet „2.5. Kvantuminformatika alkalmazási területei, jövő” című alfejezetében olvasható, az elmúlt néhány évben egyre több óriásvállalat és néhány startup kutatási témájává vált a kvantumfölény vizsgálatának témaköre. Rengeteg várakozás kapcsolódik a témakörhöz, ezért érdemes kitérni arra, hogy milyen típusú előnyök biztosítására lehetnek képesek a kvantum gépi tanuló algoritmusok a klasszikus gépi tanuló algoritmusokkal szemben. A kvantumfölény kapcsán legtöbbször csak a problémák gyorsabb megoldhatósága kerül fókuszba, azonban a kvantum gépi tanulásra vonatkozó kutatások alapján valójában négy különböző dimenziót is érdemes figyelembe venni, amikor annak a vizsgálatára kerül sor, hogy milyen javulások érhetők el kvantum gépi tanulás alkalmazásával klasszikus helyett. Ez a négy dimenzió pedig a számítási komplexitás, a minta összetettsége, a zaj elleni robosztusság és a modell összetettsége. A következőkben ez a négy dimenzió kerül részletesebben kifejtésre (Alchieri, Badalotti, Bonardi, - Bianco, 2021).

A számítási komplexitás, vagy másnéven futásidejű komplexitás a legszélesebb körben ismert dimenzió, hiszen a már említett gyorsabb megoldhatóságot takarja. Azon algoritmusok,

amelyek ezen dimenzió mentén szeretnének előnyt elérni a klasszikus változatokkal szemben - tehát fő céljuk a számítások elvégzésének felgyorsítása - általában a korábban említett hibrid megoldások közül kerülnek ki (Alchieri, Badalotti, Bonardi, - Bianco, 2021).

A minta összetettsége dimenzió az alapján értékeli az egyes megoldásokat, hogy mekkora minta (mennyi adat) szükséges ahhoz, hogy a modell képes legyen az adatokból megfelelően általánosítani. Tehát ebben a dimenzióban azon egyedek száma kerül megvizsgálásra, amelyek az adatokban „elrejtett” összefüggések, mintázatok feltárásához, adott szinten elfogadható becsléséhez kellenek (Schuld és Petruccione, 2018). A szakirodalomban nincs egyetértés azt illetően, hogy ezen dimenzióban a kvantum megoldások felülmúlják-e a klasszikus megoldásokat vagy sem. Egyes kutatások szerint a minta összetettsége alapján nem várható előny a kvantumváltozatok esetén, ugyanis bizonyítottan polinomiálisan ekvivalensek a klasszikus változatokkal. Ez pedig azt jelenti, hogy bármely gépi tanuló probléma esetében amennyiben van egy olyan kvantum gépi tanuló algoritmus, amely polinomiális sok mintát használ, akkor állítható, hogy egy szintén polinomiális sok mintát használó klasszikus gépi tanuló algoritmusnak is léteznie kell (Alchieri, Badalotti, Bonardi, - Bianco, 2021; Servedio és Gortler, 2004). Más kutatások alapján (Aharonov, Cotler, - Qi, 2022; Chen, Cotler, Huang, - Li, 2021; Cerezo, Verdon, Huang., Cincio, - Coles, 2022; Huang, Kueng, - Preskill, 2021; Huang et al., 2022) azonban ezen dimenzióban is tapasztalható lehet a kvantumfölény, mivel exponenciális kvantumfölény figyelhető meg a minta összetettségében abban az esetben, ha kvantumszenzorok, kvantummemória és kvantumszámítógépek kerülnek felhasználásra egy-egy kísérlet során előálló kvantuminformációk kinyerésére, tárolására és feldolgozására.

A zaj elleni robusztusság azt jelenti, hogy mennyire képes az adott algoritmus az adatokban megjelenő zajt kezelni, azonban ezen dimenzió kapcsán érdemes először azt tisztázni, hogy mit jelent a gépi tanuló algoritmusok esetén a zaj fogalma. Zajnak nevezhető az adatokban található minden, valamilyen módon sérült egyed. Klasszifikációs probléma esetén például sérült egyednek, vagyis zajnak számít az, ha az adott egyed célváltozójának az értéke átállításra kerül az eredeti osztályáról egy másikra. Ebből kiindulva minden egyes egyed célváltozójának az értéke (címkéje) μ valószínűséggel sérült, és μ valószínűség növelésével néhány probléma esetén nem lesznek képesek a klasszikus gépi tanuló algoritmusok a tanulásra, mivel nem rendelkeznek kellően erős robusztussággal a jelentkező zaj elviseléséhez, míg néhány kvantumváltozat a zaj ellenére továbbra is képes lesz tanulni az adatokból, csupán több adat felhasználására lesz szükségük a betanulás során (Alchieri, Badalotti, Bonardi, - Bianco, 2021; Cross, Smith, - Smolin, 2015).

A modell összetettsége dimenzió azt vizsgálja, hogy mennyire komplex a modell. Minél komplexebb egy modell, annál több összefüggés feltárására képes az adatokban, azonban annál nagyobb eséllyel van kitéve a gépi tanulás esetén egyik leggyakrabban előforduló hibának, a túltanulásnak. Ebből kifolyólag a gépi tanulás során jellemzően a keresett modell az, amelyik a legalacsonyabb komplexitás mellett képes az adatokban rejlő mintázatokot felfedezni (Alchieri, Badalotti, Bonardi, - Bianco, 2021).

Végül a kvantumfölény kapcsán érdemes kitérni arra is, hogy a szakirodalomban mely alkalmazási területeken várják a kvantum gépi tanuló változatoktól a klasszikus változatokkal szembeni előny megszerzését. Olyan esetben, amikor az adat valamilyen kvantummechanikai folyamat során jött létre, mint például különböző kémiai, biológiai, fizikai és anyagtudományokban elvégzett kísérletek, a kvantum gépi tanuló algoritmusok exponenciálisan jobban teljesíthetnek a klasszikus algoritmusoknál. Azonban amikor az adatok teljesen klasszikus eredetűek -például egy természetes nyelv vagy képfeldolgozó, egy ajánló vagy optimalizáló rendszerben használt adatok esetén – nem rendelkeznek a kvantumváltozatok ismert exponenciális előnnyel a hagyományosakkal szemben (Cerezo, Verdon, Huang., Cincio, - Coles, 2022). Amíg a kvantumszámítógépek valószínűségi eloszlásokból is képesek mintát venni, addig a klasszikus változatok számára exponenciálisan nehéznek számít ugyanez. Ezzel tehát a kvantum megoldások további jelentős előnyre tehetnének szert a klasszikusokkal szemben, amennyiben ezen valószínűségi eloszlások valamilyen valós eloszlással megegyeznek (Arute et al., 2019; Boixo et al., 2018). A kvantumfölénnyel kapcsolatban azonban fontos kiemelni, hogy igazán mindössze akkor fog tudni teljesülni, ha a jelenlegieknél hibátűrőbb kvantumszámítógépek kerülnek kialakításra (Cerezo, Verdon, Huang., Cincio, - Coles, 2022).

Jelenlegi kihívások

Az előre nem, vagy csak nehezen látható technológiai kihívások miatt a kvantum gépi tanulás fejlődése során stagnáló időszakok következhetnek be. Ezen stagnáló időszakok elkerülése a kutatók egyik legfontosabb célja a kvantum gépi tanulás kapcsán, ehhez pedig elengedhetetlen, hogy minél gyorsabban feltárássra kerüljenek ezen kihívások és célzott kutatásokkal különböző megoldások kerülhessenek kidolgozásra rájuk (Cerezo, Verdon, Huang., Cincio, - Coles, 2022). A továbbiakban a teljesség igénye nélkül röviden megemlítsük kerül néhány jelenlegi kvantum gépi tanulást érintő legjelentősebb kihívás.

Jelenleg az egyik legnagyobb kihívását ezen a területen a klasszikus adatok, adatsémák és információ kvantum állapotokká történő legmegfelelőbb átkódolása jelenti. Ahogy a

klasszikus gépi tanulás fejlődéséhez is rendkívül nagymértékben hozzájárult a magas minőségű, szabványosított adatkészletek megjelenése, úgy a kvantum gépi tanulásban is várhatóan további áttörésekhez vezethetne, ha nem főként klasszikus adatkészletekkel, hanem egyszerűen előállítható kvantumállapotokkal rendelkező szabványosított kvantum adatkészletekkel dolgoznának a kvantumváltozatok (Cerezo, Verdon, Huang., Cincio, - Coles, 2022). Amint az alfejezet előző, „Kvantumfölény” részében is megemlítsre került, a kvantumadatokat felhasználó kvantummodellek felé erősebb várakozásokat támaszt a szakirodalom a kvantumfölény elérésével kapcsolatban, mint a nem kvantum adatokat használók felé (Chen, Cotler, Huang, - Li, 2021, 2022; Cotler, Huang, - McClean, 2021; Huang et al., 2022). Ezen kihívás megoldására a szakirodalom szerint a kvantum gépi tanuló algoritmusok különböző interfész eszközök (például qRAM) integrálását igénylik a klasszikus információk kvantummechanikai változatának előállítása érdekében (Khan – Robles-Kelly, 2020).

A második kihívásként a kvantumzajok hatásait érdemes megemlíteni. A hardverek működése során fellépő zajok torzíthatják az információt egy kvantum áramkörben történő áramlaskor, ez a torzítás pedig felerősödhet abban az esetben, ha mélyebb, összetettebb, nagyobb futási időt igénylő kvantumáramkörökön halad végig az információ. Ezért kiemelten fontos lenne a kvantumhardvereket praktikusabbá, stabilabbá tenni (Cerezo, Verdon, Huang., Cincio, - Coles, 2022).

További kihívást jelent még a kvantum gépi tanuló modellek paramétereinek az optimalizálása, azaz olyan paraméterkészletek megtalálása, amelyekkel minimalizálható egy adott feladat megoldásakor a kvantum modell által elkövetett hibák száma. A kvantum neurális hálózatok kapcsán pedig a hatékony és eredményes felépítés, architektúra megtervezése jelent kihívást. Ezutóbbi kihívás megoldása azért is olyan fontos a kvantum gépi tanulás számára, mert a klasszikus gépi tanuláshoz hasonlóan kvantum esetben is a neurális hálózatok alapvető eszközei lehetnek számos felügyelt, nem felügyelt és megerősítéses tanulási probléma megoldásának is (Cerezo, Verdon, Huang., Cincio, - Coles, 2022). Szintén a kihívások közé sorolható az is, hogy bár az adatok feldolgozásában jelentősen gyorsabb lehet egy kvantumváltozat a klasszikusnál, de az adatok beolvasásánál már nem állítható ugyanez. Néhány esetben ugyanis a kvantum megoldások alkalmazásakor az adatok beolvasásának költsége meghaladhatja az adatok feldolgozásának gyorsaságával járó előnyöket (Mishra et al., 2021).

4. Kutatási kérdés és módszertan bemutatása

Jelen dolgozat arra a kérdésre keresi a választ, hogy a mindenki számára ingyenesen, szabadon elérhető Python-alapú szoftver-megoldások esetén valóban tapasztalható-e a technológiailag megjósolt kvantumfölény a gépi tanulás témakörében, azaz az előzetes várakozásoknak megfelelően ténylegesen jobb és gyorsabb teljesítmény figyelhető-e meg egy kvantumváltozat alkalmazásakor a klasszikus változatokkal szemben. Ezen kérdés megvizsgálására kvantum és klasszikus gépi tanuló algoritmusok pontossága és teljesítménye került összehasonlításra. Két különböző méretű, nyilvánosan elérhető adatbázis felhasználásával a jelenleg legnépszerűbbnek számító gépi tanulási problémák közé tartozó bináris klasszifikációs problémára került összehasonlításra egy, a szakirodalomban korábban már publikált kvantum neurális hálózat a klasszikus SVM (Support Vector Machine) és k-NN (k-Nearest Neighbor) algoritmusokkal, valamint egy szintén klasszikus alapon működő automatizált gépi tanulás segítségével összeválogatott algoritmusegyüttessel. Az összehasonlítás során megvizsgálásra kerültek az osztályozás olyan pontossági jellemzői, mint az accuracy, precision, recall, F1, ROC görbe alatti terület (AUC), valamint a tanuláshoz és az osztályozáshoz szükséges idők.

Ebben a fejezetben először a kutatás során felhasznált eszközök ismertetéséről lesz szó, majd részletesebben bemutatásra kerülnek az ezen eszközök segítségével implementált kvantum- és a klasszikus algoritmusok, végül pedig az algoritmusok számára adatokat biztosító két adatbázis, az Iris és a Pima Indians Diabetes adatkészletek bővebb leírása következik. A modellek összehasonlításához megvizsgált pontossági jellemzőkre, valamint a tanuláshoz és osztályozáshoz szükséges időkre részletesebben az „5.1. Az összehasonlításhoz használt mérőszámok” című alfejezet tér ki.

4.1. Eszközök bemutatása

Ebben az alfejezetben azok az eszközök kerülnek bemutatásra, amelyek a jelen kutatás során a kvantum és a klasszikus gépi tanuló algoritmusok implementálása, értékelése, valamint összehasonlításakor felhasználásra kerültek. Először röviden a Python programozási nyelvről lesz szó, amelyben az egyes modellek implementálásra kerültek, majd megemlítsük néhány a megoldások során alkalmazott, Python programozási nyelvhez írt ingyenes kvantum és klasszikus gépi tanulási könyvtár: a PennyLane, a Scikit-learn és az Auto-Sklearn. Végül pedig a Google Colaboratory-ról lesz szó, amely platform fejlesztői és futtató környezetként szolgált a kutatás során, hiszen itt kerültek megírásra és futtatásra a kvantum és klasszikus gépi

tanuló algoritmusok implementálására, értékelésére, valamint összehasonlítására megírt Python kódok.

Python programozási nyelv

A Python egy általános célú, magas szintű, interpreter programozási nyelv. Az objektum orientált programozáson kívül más programozási paradigmákat is támogat, mint a procedurális és a funkcionális programozás. Ezen kívül számos modult, dinamikus típusozást, nagyon magas szintű dinamikus adattípusokat és osztályokat tartalmaz, valamint tiszta, könnyen érthető szintaxist (Python, 2023a). Könnyen tanulható nemcsak más programozási nyelveket már ismerő, tapasztalt programozók, de teljesen kezdő programozók számára is. További előnyei közé tartozik még a kiterjedt, egyszerűen hozzáférhető dokumentációja, valamint a széles és aktív fejlesztői közössége, a közhasznú Python Software Foundation által felügyelt nyílt forráskódú licence, amely lehetővé teszi a programozási nyelv ingyenes, szabad használatát és terjesztését, akár kereskedelmi célú felhasználásra is (Python, 2023b). A programnyelv egyik legnagyobb erősségének azonban a széleskörű interfész és kiegészítő könyvtárgyűjteményét tartják, amelynek köszönhetően a Python olyan területeken használható fel, mint a webfejlesztés, (természet)tudományok, adatelemzés és modellezés, gépi tanulás, (programozás) oktatás, szoftverfejlesztés, asztali grafikus felhasználói felületek építése, különböző üzleti felhasználások (vezetői döntéshozatal támogatása, vállalatirányítási rendszerek és e-kereskedelmi rendszerek) (Python, 2023c).

PennyLane könyvtár

A PennyLane egy Python nyílt forráskódú, a kvantum gépi tanulást lehetővé tevő könyvtár, amely platformfüggetlenül képes különböző backend-eken a differenciálható programozási paradigma segítségével kvantumprogramok végrehajtására és kvantum gépi tanuló algoritmusok betanítására, futtatására. Ezen könyvtárat a kanadai Xanadu fejlesztette ki, amely jelenleg az egyik legjelentősebb full-stack kvantum startup. A könyvtár olyan a gépi tanulásban elterjedt keretrendszerekkel köti össze a kvantumszámításokat, mint a NumPy autograd-ja, a JAX, a PyTorch és a TensorFlow, ezzel lehetővé téve a kvantum gépi tanulást, a kvantum áramkörök tanítását. A PennyLane tehát támogatja a hibrid kvantum és klasszikus modellek létrehozását, alkalmazását, segítségével kvantumszámítások végezhetők el, beleértve a kvantum áramkörök értékelését és gradienseinek kiszámítását. Az így kapott információt pedig továbbítja a könyvtár a klasszikus keretrendszer felé, ezzel egy működő kvantum-klasszikus pipeline-t létrehozva. A kvantum áramkörök gradienseinek kiszámítását

amennyiben van lehetőség rá, úgy paramétereltolási szabályok alapján (az áramkör futtatása kétszer vagy többször történik szándékosan eltolt paraméterek alkalmazásával) valósítja meg, egyébként pedig véges különbség módszer alapján történő becsléssel. A PennyLane könyvtár alkalmazásával különböző kvantumszámítógép szimulátorokon és valós kvantumszámítógépeken is lehetséges a kvantum áramkörök futtatása, hiszen a könyvtárba alapértelmezetten néhány kvantum-szimulátor beépítésre került, ezen kívül pedig plugin rendszer segítségével, számos külső szoftver és hardver integrálásával - például az IBM Qiskit-jével, a Google Cirq-jével, a Rigetti Forest-jével vagy a Xanadu Strawberry Field-jével - lehetőség nyílik a kvantum áramkörök valós kvantumszámítógépeken történő futtatására is. A könyvtár továbbá lehetővé teszi a kvantum áramkörök beépített automatikus differenciálását és számos optimalizálást, valamint gépi tanulást támogató eszköz felhasználását. Mivel a PennyLane a standard kvantum programozási nyelvek számos funkcionalitásának megvalósítása mellett az imént említett kvantum áramkörök automatikus differenciálással történő tanítására is lehetőséget nyújt, ezért különösen jól alkalmazható olyan területeken, mint a kvantum gépi tanulás, a kvantum optimalizálás vagy a kvantum kémia (Bergholm et al., 2018; Ganguly, 2021; PennyLane, 2023a; PennyLane, 2023b).

Scikit-learn könyvtár

A Scikit-learn szintén a Python programnyelv ingyenes, BSD licence-el rendelkező könyvtára, amely segítségével könnyen elérhetővé válik számos jelen pillanatban legelterjedtebb gépi tanuló algoritmus. A könyvtár felügyelt és felügyelet nélküli tanulási problémák megoldására alkalmas algoritmusokat is tartalmaz, segítségével pedig gyorsan, egyszerűen, kiterjedt dokumentáció használata mellett implementálhatók, értékelhetők és hasonlíthatók össze a különböző gépi tanuló algoritmusok segítségével megépített modellek. A Scikit-learn használatával előzetes tapasztalatok és komoly szakértelem nélkül, kezdőként is lehetőség nyílik számos gépi tanuló algoritmus kipróbálására, hiszen a könyvtár magas szintű nyelven íródott, rendkívül felhasználóbarát és könnyen használható (Pedregosa et al., 2012).

Auto-Sklearn könyvtár

A Python ingyenes, nyílt forráskódú Auto-Sklearn könyvtára a Scikit-learn könyvtárra épít, alkalmazása során pedig automatikusan megkeresi a számára megadott, felügyelt tanulásra alkalmas adatkészleten a probléma megoldására leginkább megfelelő tanuló algoritmus-együttest és optimalizálja a tanulási folyamat vezérlésére szolgáló hiperparamétereket, ezzel lehetővé téve, hogy a felhasználója a helyes modell megtalálása és hiperparamétereinek

optimalizálása, hangolása helyett a valós üzleti vagy egyéb problémára fókuszálhasson. Az Auto-Sklearn tehát az általános gépi tanuló keretrendszert kiegészíti egy globális optimalizálással, és ahhoz, hogy az általa javasolt modell képes legyen kellően általánosítani, a könyvtár 15 osztályozó algoritmust és 14 a megadott adatkészletben található attribútumok (features) előfeldolgozására alkalmas algoritmust tartalmaz, amelyek közül a globális optimalizálás folyamata során az összeset teszteli, és ezen tesztelt algoritmusok közül épít egy általa összeválogatott algoritmus-együttest. Továbbá az optimalizációs folyamat meggyorsítására az Auto-Sklearn úgynevezett meta-tanulást használ, hogy könnyen azonosíthasson egymáshoz hasonló adatkészleteket, és ezáltal felhasználhassa a korábbi adatkészleteken már megszerzett „tudását”. Ez a könyvtár lehetővé teszi az adatskálázást is, illetve a kategorikus paraméterek és a hiányzó értékek kezelését (Feurer et al., 2015; AutoML.org, 2023).

Egyéb felhasznált Python könyvtárak

A kutatás során a PennyLane, a Scikit-learn és az Auto-Sklearn könyvtárakon kívül még olyan egyéb Python-hoz írt könyvtárak is felhasználásra kerültek, mint az adatvizualizációra alkalmas Seaborn, Matplotlib, és Plotly könyvtárak, az adatok esztétikus, átlátható kiírására szolgáló Pprint könyvtár, az adatok feldolgozását és elemzését lehetővé tevő Pandas könyvtár, a nagyméretű, akár többdimenziós tömbök, valamint mátrixok használatát támogató NumPy könyvtár, illetve az egyes modellek betanítására és értékelésére szánt idők kiszámításához a time modul. A nagyobb, Pima Indians Diabetes adatkészlet beimportálása pedig a google.colab könyvtár files modulja, valamint az io modul segítségével történt meg.

Google Colaboratory

A Google Colaboratory, vagy röviden Colab, egy olyan eszköz, interaktív környezet, amely segítségével a böngészőben írhatók és futtathatók különböző Python programozási nyelvben írt kódok, ezen kívül lehetőség nyílik továbbá formázott szövegek, HTML kódok és képek beszúrására is. A Colab-on jegyzetfüzetek hozhatók létre, amelyekbe szövegek és kódcellák beszúrására van lehetőség, ezek a cellák pedig egymástól függetlenül, külön-külön, de akár sorban, egymás után is lefuttathatók, az eredmény pedig a futást követően a cella alatt kiírásra is kerül. A Colab jegyzetfüzetek olyan Jupyter jegyzetfüzetek, amelyek a felhasználó saját Google Drive-ján kerülnek tárolásra, így támogatva a csoportos munkát, valamint az egyszerű, gyors elérést, hiszen a jegyzetfüzetek nagyon könnyen megoszthatók másokkal és a Google fiókba történő bejelentkezés után bármilyen eszközről elérhetők. Azon felhasználók,

akkikkel megosztásra került egy adott Google Colab munkafüzet, a megfelelő jogosultság mellett szerkeszthetik a munkafüzet tartalmát, valamint megjegyzéseket is írhatnak. A Colab tehát jelentősen elősegíti a különböző adattudományos és gépi tanulási projektek kivitelezését és oktatását, hiszen előzetes konfiguráció nélkül, a felhasználó saját, lokális számítógépének teljesítményétől függetlenül, a Google felhőszerverein, nagy teljesítményű hardvereken, gyorsan és kényelmesen futtathatóvá válnak a Python kódok a böngészőn keresztül (Google, 2023).

4.2. Felhasznált algoritmusok

Ebben a fejezetben először nagyvonalakban bemutatásra kerül a kutatás során felhasznált, a szakirodalomban Santanu Ganguly 2021-es Quantum Machine Learning: An Applied Approach című könyvében korábban már publikált kvantum neurális hálózata, majd ezt követően ismertetésre kerül az összehasonlításhoz implementált három klasszikus megoldás is: a Support Vector Machine, a k-Nearest Neighbors és az automatizált gépi tanulás.

4.2.1. Kvantum neurális hálózat

A kvantum neurális hálózatok (QNN) kapcsán érdemes először a fogalom jelentését tisztázni. A kvantum neurális hálózat egy olyan mesterséges neurális hálózat, amely kvantum információ feldolgozására alkalmas. Akárcsak a klasszikus neurális hálózatok, a kvantum neurális hálózatok is az emberi agyhoz hasonlóan működnek, viszont a klasszikus változatokkal szemben képesek kihasználni a kvantumtulajdonságok nyújtotta előnyöket is. Kvantum neurális hálózatokat főként olyan problémák megoldására alkalmaznak, amelyeket klasszikus módszerekkel nem, vagy csak nagyon nehezen lehetne kivitelezni az exponenciálisan növekvő számítási és memória igényük miatt. A kvantum gépi tanulás témakörében egyre többször utalnak a kvantum neurális hálózatok a variációs áramkörökre. Variációs áramkörök alatt azon kvantum algoritmusok értendők, amelyek állítható paraméterekkel rendelkeznek, emiatt pedig optimalizálhatók. Ehhez kapcsolódóan továbbá érdemes megemlíteni a variációs kvantum osztályozókat (VQC), vagyis az olyan felügyelt gépi tanuló algoritmusokat, amelyekben a variációs áramköröket klasszifikációs feladatok elvégzésére tanították be. A kvantum neurális hálózatok kapcsán röviden érdemes még kitérni a kvantum konvolúciós neurális hálózatokra és az úgynevezett „kvanvolúciós neurális hálózatokra” (quantvolutional neural networks) is. Az előbbi fogalom olyan kvantum neurális hálózatokat takar, amelyek a klasszikus neurális hálózatok szerkezetét tükrözik. A kvantum konvolúciós neurális hálózatokat a kvantum mérések hatására összevonásra került rétegek és váltakozó konvolúciós rétegek jellemzik. Míg az utóbbi fogalom, az úgynevezett „kvanvolúciós neurális hálózatok” olyan hibrid klasszikus-

kvantum modelleket jelentenek, amelyekben variációs kvantum áramkörök rétegei fokozzák a klasszikus konvolúciós neurális hálózatokat (Ganguly, 2021).

A kvantum gépi tanulás területének újdonsága és kiforratlansága miatt számos klasszikus gépi tanuló algoritmusnak jelenleg még nem tették mindenki számára elérhetővé és felhasználhatóvá a kvantumos változatát. Azonban a jelen kutatásban implementált kvantum neurális hálózat, egy a szakirodalomban korábban már publikált, szintén az Iris adatkészlet módosított, bináris klasszifikációra alkalmas változatára írt kvantum neurális hálózat. Mindkét adatbázis, az Iris és a Pima Indians Diabetes esetén is Santanu Ganguly 2021-es Quantum Machine Learning: An Applied Approach könyvének „Chapter 8: Deep Quantum Learning” című fejezetének „Binary Classifier with PennyLane” részében leírt kvantum neurális hálózata került implementálásra. Ezen kvantum neurális hálózat a bináris klasszifikációra alkalmas Iris adatkészletre írt forráskódja bárki számára elérhető a könyv többi fejezetéhez tartozó forráskódokkal együtt a könyv kiadójának, az Apress-nek a quantum-machine-learning nevű GitHub repozitóriumában.¹

A kvantum neurális hálózat implementálásakor elsőként a PennyLane telepítésére volt szükség, ahol ahhoz, hogy a kvantum neurális hálózat forráskódjában szereplő összes PennyLane könyvtár beimportálható lehessen, nem a legfrissebb, hanem a PennyLane egyik korábbi, 0.14.1-es verziójának telepítésére került sor. A neurális hálózat rétegeinek megépítése a StronglyEntanglingLayers, a hálózat optimalizálása pedig a GradientDescentOptimizer segítségével valósult meg. A kvantum neurális hálózat a DefaultQubit függvény alkalmazásával a PennyLane alapértelmezett, szabványos kvantumbit-alapú eszközén került futtatásra, amely nagyon egyszerű szimulációját valósítja meg egy qubit-alapú kvantum áramkör architektúrájának. Továbbá a kvantum neurális hálózat implementálásával kapcsolatban még azt érdemes kiemelni, hogy a jelen kutatásban a bináris klasszifikációhoz szükséges tanító és teszt adathalmaz Ganguly eredeti kvantum neurális hálózatában található 80-20%-os aránya helyett 75-25%-os arányban került felosztásra. Ez a kis változtatás pedig azért volt szükséges, mert korábban már a kutatásban implementált klasszikus megoldások esetén is a 75-25%-os arány került alkalmazásra és a kvantum-klasszikus eredmények összehasonlítása szempontjából nagyon fontos, hogy a tanító-teszt arányok a kvantum és a klasszikus megoldások esetén

¹ GitHub repozitórium elérése: <https://github.com/Apress/quantum-machine-learning>

A repozitóriumon belül a Chapter_8_codes mappán belüli qml_iris_pl.ipynb fájl tartalmazza a jelen kutatás által is felhasznált kvantum neurális hálózat forráskódját. Fájl elérése: https://github.com/Apress/quantum-machine-learning/blob/main/Chapter_8_codes/qml_iris_pl.ipynb

megegyezzenek. Emellett a nagyobb adatkészlet, a Pima Indians Diabetes esetén az n_epochs értékének növelésére is szükség volt, ugyanis a Ganguly kvantum neurális hálózatában található 10 érték megadása esetén a kvantum neurális hálózat nem tudta elvégezni a bináris osztályozást a Pima Indians Diabetes adatkészlet esetén, hiszen minden egyedet eleve abba az osztályba sorolt, amelybe az adatbázis egyedeinek többsége tartozott. Az n_epochs paraméter azt állítja be, hogy hányszor kell az egész tanuló adatkészleten a tanuló algoritmusnak végigfutnia az összefüggések megfelelő feltárásához, és mivel egy nagyobb méretű adatbázis esetén ez többször lehet szükséges, mint egy kisebb adatkészletnél, ezért n_epochs növelésével javítható a kvantum neurális hálózat klasszifikációs képessége. Így az eredeti $n_epochs = 10$ érték mellett 100, 200, 300, 400 és 500 értékek esetén is betanításra és értékelésre került a kvantum neurális hálózat a Pima Indians Diabetes esetén, amelynek eredményeképp már $n_epochs = 100$ érték megadása esetén is nemcsak a többségi, hanem a másik osztályba is kerültek egyedek, tehát a kvantum neurális hálózat az n_epochs növelésével valóban képessé vált az osztályozásra. Ahhoz, hogy jelen kutatás konzisztens maradjon, az Iris adatbázisra alkalmazott kvantum neurális hálózat esetében is az eredetileg $n_epochs = 10$ paraméter-beállítás mellett 100, 200, 300, 400 és 500 értékek esetén is betanításra és értékelésre került a modell. Az Iris és Pima Indians Diabetes adatbázison futtatott kvantum neurális hálózatok n_epochs paraméterének megválasztását bővebben a „5.2. Kvantum neurális hálózat eredményei” című alfejezet mutatja be.

4.2.2. Klasszikus gépi tanuló algoritmusok

Ebben az alfejezetben a kutatás során az összehasonlításhoz kiválasztott három klasszikus gépi tanuló megoldás működése, valamint a hozzájuk megadott paraméter-beállítások kerülnek röviden ismertetésre. A paraméter-beállítások kapcsán bemutatásra kerül az is, hogy milyen módszer segítségével történt meg az optimális paraméterek kiválasztása az egyes modellek esetén.

Support Vector Machine (SVM)

A Support Vector Machine (SVM) a felügyelt tanulásban alkalmazott legnépszerűbb algoritmusok egyike. Az SVM-en már az 1990-es évek óta dolgoznak, matematikailag robosztus, komplex megközelítés, amely viszonylag nagy számítási kapacitást igényel. Mégis a jelenleg legelterjedtebb gépi tanuló algoritmusok közé tartozik, ugyanis kiválóan alkalmazható számos összetett adatkészlet esetén klasszifikációs problémák megoldására. Segítségével viszonylag egyszerűen lehet komplex, nagy dimenziójú, sok attribútummal rendelkező adatkészletek esetén is helyes becsléseket adni arra, hogy az adott egyed melyik

osztályba sorolható. Az SVM algoritmusnak az a fő célja, hogy a tanuló adatokat különböző osztályokra válassza szét egy olyan hipersík segítségével, amely maximalizálja az elválasztást, elkülönítést az egyes osztályok között, ezen elválasztások kapcsán pedig lineáris és nem lineáris elválasztófelületről is lehet szó. Az SVM továbbá a modell általánosító képességét is igyekszik maximalizálni. A modell általánosító képessége azt jelenti, hogy a modell mennyire képes a tanuló adatokon feltárt összefüggések alapján helyes becslést adni egy általa korábban még nem látott, ismeretlen egyed osztályára. Egy modell akkor rendelkezik jó általánosító képességgel, hogyha nem jellemző rá sem az alul- sem pedig a túltanulás. Az SVM algoritmus kapcsán végül érdemes megemlíteni, hogy számos előnye mellett felmerülhet néhány korlátja, gyengesége is a használata során. Ilyen például a hosszú betanulási idő az algoritmus komplex, nagy számítási kapacitást igénylő természete miatt, a nehézkes paraméter megválasztás, a több osztállyal rendelkező (nem bináris) klasszifikációs problémákra optimális SVM osztályozó implementálása, valamint az algoritmus rosszabb teljesítménye a kiegyensúlyozatlan (az egyik osztályba jelentősen több egyed tartozik, mint a másikba) adatbázisokon történő futtatás esetén (Ganguly, 2021; Cervantes, Garcia-Lamont, Rodríguez-Mazahua, - Lopez, 2020).

Jelen kutatásban az SVM algoritmus a Scikit-learn könyvtár SVC (`sklearn.svm.SVC`) függvénye segítségével került implementálásra. Az Iris és a Pima Indians Diabetes adatbázis esetén is az SVC (C-Support Vector Classification) optimális paramétereit előzetesen szintén a Scikit-learn könyvtár egyik modell szelekciós függvényének, a `GridSearchCV` (`sklearn.model_selection.GridSearchCV`) segítségével került kiválasztásra. A `GridSearchCV`-nek mindkét adatkészlet esetén olyan kipróbálandó paraméter beállítások kerültek megadásra, mint a C szabályzási paraméter (amely fordítottan arányos a szabályzás erősségével) 1, 10, 100 és 1000-re történő beállítása, lineáris és rbf kerneltípusok kipróbálása, valamint ezutóbbi kerneltípus esetén gamma kernel együttható értékére 0,001 és 0,0001 értékek kipróbálása. A `GridSearchCV` segítségével így kijött optimális paraméter beállítások mindkét adatkészlet esetén lineáris kernel beállítást, ebből kifolyólag a gamma kernel együttható értékére pedig az alapértelmezett 'scale'-t javasolták. C szabályzási paraméterre azonban az Iris adatkészlet esetén 1, a Pima Indians Diabetes adatkészlet esetén pedig 100 értékek beállítását ajánlották. Az SVC implementálásakor tehát az így kapott optimális paraméter beállítások kerültek megadásra mindkét adatbázison történő alkalmazás során.

k-Nearest Neighbors (k-NN)

A k-Nearest Neighbors (k-NN) algoritmus az egyik legkönnyebben implementálható klasszikus gépi tanuló algoritmus. A k-NN algoritmus strukturált, címkézett, tehát felügyelt

gépi tanulásra alkalmas adatokon dolgozik jól, és az egyes adatokat egymáshoz viszonyított elhelyezkedésük, szomszédjaiktól vett távolságuk alapján sorolja csoportokba. A távolságok kiszámolására többféle megközelítés is alkalmazható, például euklideszi vagy Mahalanobis távolság kiszámolása. A k-NN-nek szüksége van egy címkézett célváltozóval rendelkező tanuló adathalmazra, amelyben ismert, hogy mely egyed a célváltozó melyik osztályába tartozik. Így egy ismeretlen, korábban az algoritmus által még nem látott egyed esetén az adott egyedhez legközelebb eső, bizonyos számú szomszédok célváltozóinak értéke alapján becsli meg az ismeretlen egyed osztályát. Az ismeretlen egyedet ezáltal abba az osztályba sorolja, amelybe az egyed adott számú legközelebbi szomszédjainak a többsége is tartozik. Az algoritmus nevében a „k” betű arra utal, hogy hány darab legközelebb eső szomszéd kerül megfigyelésre (k = 1 esetén csak a legközelebb eső, k = 2 esetén a két legközelebb eső stb.). A k-NN alkalmazásakor így elsőként az a kérdés vetődik fel, hogy mely k a legoptimálisabb választás. Minél nagyobb számmal egyenlő k (minél több szomszéd figyelembevételére kerül sor), annál egyszerűbb lesz a modell és az algoritmus által alkalmazott, az egyes osztályokat elválasztó vonal, másnéven hipersík annál simább lesz. Kisebb k megválasztása esetén (kevesebb szomszéd figyelembevételkor) viszont komplexebb lesz a modell, az osztályokat elválasztó vonal pedig élesebb lesz, ez pedig túltanuláshoz vezethet. Kis k megválasztásakor további veszélyt jelenthet az, hogyha az ismeretlen egyed – amely osztályára a becslés irányul – pont egy másik osztályba tartozó egyed egyik kilógó (outlier) értékéhez esik közvetlen közel, hiszen így a modell könnyen rossz osztályba sorolhatja az ismeretlen egyedet, amennyiben annak csak nagyon kevés szomszédját veszi figyelembe. A k-NN algoritmus tehát nagyon érzékeny az adatokban található hibákra, a jelentősen kilógó értékek gyengébb osztályozási eredményekhez vezethetnek. A legmegfelelőbb k kiválasztására érdemes megnézni, hogy melyik k esetén lesz megfigyelhető a modellnek a tanító és a teszt adathalmazon történő futtatások során elért accuracy-jei között a legkisebb különbség. Minél kisebb a modell tanító és teszt adatokra kijött accuracy-jei közötti különbség, annál optimálisabb a modellre megválasztott k száma (Fodor, 2022a; Ganguly, 2021; Mucherino, Papajorgji, - Pardalos, 2009).

A kutatás során a k-NN algoritmust az Iris és a Pima Indians Diabetes adatkészlet esetén is a Scikit-learn könyvtárba beépített KNeighborsClassifier (sklearn.neighbors.KNeighborsClassifier) függvény segítségével került implementálásra. Az optimális k megtalálásához mindkét adatkészlet esetén egy for ciklus belsejében k-t 1-től 8-ig terjedő egész számokkal behelyettesítve kiszámolásra került minden egyes iterációban a modell tanuló és teszt adatokon végzett osztályozásaira kapott accuracy-jei, majd az előző bekezdésben említett módszert követve az így kijött eredmények alapján a legmegfelelőbb k, vagyis az az 1-

től 8-ig terjedő egész szám került kiválasztásra k értékére, amely alkalmazásakor a legalacsonyabb különbség volt megfigyelhető a tanuló és teszt adathalmazokon történő futtatáskor elért accuracy-k között. Nyolcnál nagyobb k behelyettesítésére a modell túlságos leegyszerűsítése, az alultanulás elkerülése miatt nem került sor. Az Iris adatkészlet esetén minden k -ra azonosak voltak a tanuló és teszt adatokon történő futtatásokkor kijött accuracy-k, míg a Pima Indians Diabetes adatkészlet esetén $k = 6$ -ra volt a legkisebb eltérés a tanuló és teszt adatokra kijött accuracy-k között. Így végül mindkét adatbázis esetén $k = 6$ került megválasztásra k értékére a k -NN algoritmus alkalmazásakor.

Auto Machine Learning (AutoML)

A klasszikus megoldások közül implementálásra került továbbá egy, az automatizált gépi tanulás segítségével összeválogatott algoritmus-együttes is. Az automatizált gépi tanulás egy új megközelítés a klasszikus gépi tanuláson belül, amelynek az a fő célja, hogy az egyes gépi tanuló problémák megoldására megkeresse a legalkalmasabb algoritmusokat, valamint azok hiperparamétereinek optimális megválasztását, viszonylag gyorsan és teljesen automatizáltan, komolyabb szakértelemmel rendelkező felhasználót nem igényelve. Az AutoML tehát az adott gépi tanuló probléma megoldására az általa legmegfelelőbbnek vélt algoritmusokból összeválogat egy együttest, amely elemeit a végső megoldásban az általa meghatározott sorrendben és súllyal használja fel. Az automatizált gépi tanulás számos előnnyel rendelkezik: könnyen elérhetővé teszi, segíti a gépi tanulás megértését, javítja a megoldásokat, a hatékonyságot, ezen kívül időt takarít meg, az optimális hiperparaméterek automatikus megkeresése lehetővé teszi, hogy több idő jusson az emberek számára érdekesebb feladatokra, például a kijött eredmények értelmezésére, konklúzió levonására. Az autoML használata továbbá nem igényel magas szakértelmet, ez viszont bizonyos esetekben hátrányt jelenthet, így anélkül is használhatóvá válik, hogy a felhasználó bármit is megértene belőle (AutoML.org, 2023; Fodor, 2022b).

Jelen kutatásban az automatizált gépi tanulás segítségével létrejött algoritmus együttes az Auto-sklearn könyvtár AutoSklearnClassifier megoldásával került létrehozásra (autosklearn.classification.AutoSklearnClassifier). Ehhez első lépésként a 0.15.0 verziójú Auto-sklearn könyvtár telepítésére volt szükség, ezt követően pedig importálhatóvá váltak a könyvtárban található automatizált gépi tanulást lehetővé tevő megoldások. Az AutoSklearnClassifier alkalmazásakor az Iris és a Pima Indians Diabetes adatbázisok esetén is három bemeneti paraméter került megadásra: a feladat végrehajtására szánt idő, a modell egyetlen hívására fordítható maximum időkorlát, valamint annak a mappának az elérési

útvonala, ahova a konfigurációs output és log fájlok kerülnek. A végrehajtásra szánt időre (`time_left_for_this_task`) 120 másodperc került beállításra, amely azt jelenti, hogy maximum 2 perc fordítható az adott adatkészlet számára legmegfelelőbb gépi tanuló modellek megkeresésére. A végrehajtásra szánt idő növelésével egyébként nagyobb eséllyel találhatók jobb modellek, azonban a futási idő is hosszabb lesz. A modell egyetlen hívására fordítható maximum időkorlát (`per_run_time_limit`) értékére 30 másodperc került megválasztásra, amely hatására a modell tanuló adatokra történő illesztése fél perc után megszakításra kerül. Végül a harmadik bemeneti paraméterre, a konfigurációs output és log fájlok tárolására szolgáló mappa elérési útvonalára (`tmp_folder`) a „`tmp/autosklearn_classification_example_tmp`” mappa került megadásra. Mindkét adatkészlet esetén az előbbiekben bemutatott értékek kerültek beállításra az egyes paraméterekre.

4.3. Felhasznált adatok

Ezen fejezetben a kutatás során felhasznált két adatkészlet, az Iris és a Pima Indians Diabetes adatkészlet kerül részletes bemutatásra. Először a két adatkészlet beimportálásának, valamint általános bemutatásának leírására kerül sor, végül az adatkészleteken elvégzett előzetes műveletekről, adattisztításról, valamint az Iris adatkészlet esetén a bináris klasszifikációra alkalmas változat megalkotásáról lesz szó.

Iris adatbázis

Az Iris adatbázis az egyik legismertebb és legelterjedtebb adatkészlet a gépi tanulás területén, amely kiváló adathalmazként szolgál klasszifikációs problémák megoldására betanítani kívánt modellek számára. Mivel az adatkészlet rendkívül széles körben ismert és gyakran alkalmazott, ezért a Python programnyelv egyik gépi tanuló könyvtárából, a Scikit-learn-ből is könnyen beimportálható, hiszen az Iris is egyike a Scikit-learn-be beépített népszerű adatkészleteknek. Ezen kívül az adatkészlet az interneten számos oldalon gyorsan megtalálható és letölthető, valamint a Python Pandas könyvtárában található csv fájlok beolvasására alkalmas függvény (`read_csv()`) segítségével URL megadásával is egyszerűen beimportálható.

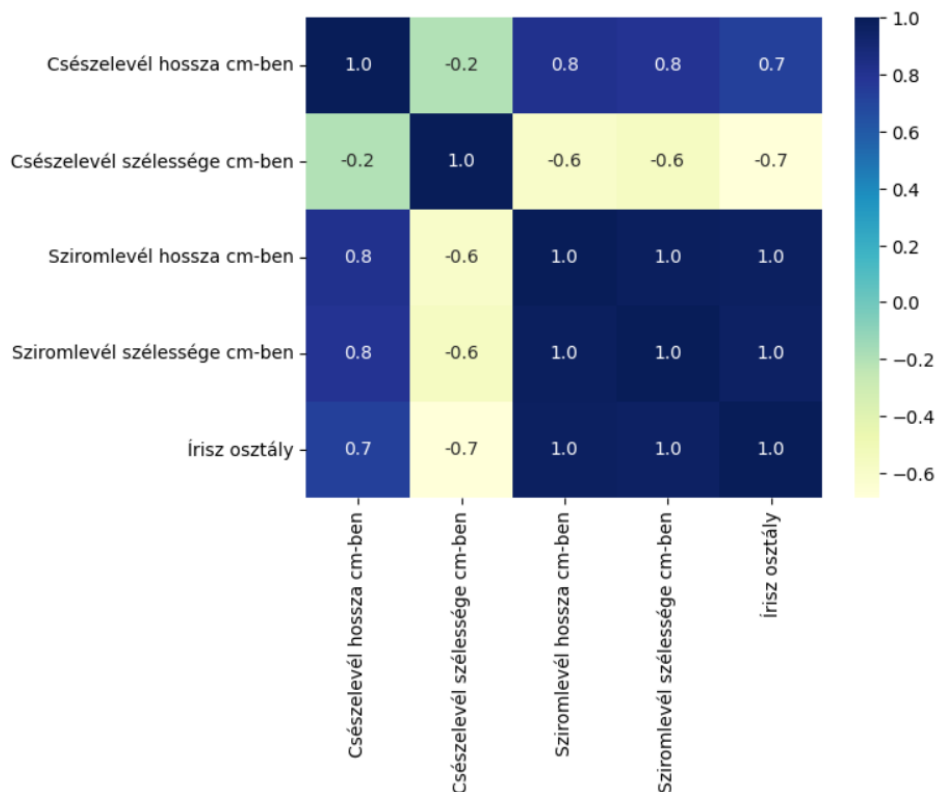
Az Iris adatkészletet egyébként 1936-ban Ronald Fisher brit statisztikus és biológus hozta létre. Az adatkészlet három különböző osztályt tartalmaz, amelyből mindegyik az írisz, másnéven nőszirom virág valamelyik típusába sorolja az egyedeket: Iris setosa, Iris versicolor vagy Iris virginica. Mindegyik osztályba 50 egyed tartozik, így összesen 150 egyed található az egész adatkészletben. Az Iris négy numerikus attribútumból áll: a csészelevél hossza centiméterben, a csészelevél szélessége centiméterben, a szíromlevél hossza centiméterben, és

a sziromlevél szélessége centiméterben. Az adatkészlet utolsó oszlopában pedig kategorikus változóként az adott egyed osztálya szerepel, tehát az, hogy melyik írisz típusba tartozik a három közül. Fontos megjegyezni továbbá, hogy az adatkészletben nem szerepelnek hiányzó adatok, valamint, hogy amíg a csésze- és sziromlevelek hosszával és szélességével kapcsolatos tulajdonságok nem választhatók el lineárisan egymástól, addig egy írisz típus osztálya lineárisan elválasztható a másik kettő osztálytól. Ezutóbbi azért jelentős, mert a kutatás során bináris klasszifikációs problémára megoldást nyújtó kvantum és klasszikus modellek implementálására, értékelésére, és összehasonlítására került sor, tehát szükség volt az eredeti Iris adatkészlet átalakítására azért, hogy egy módosított, bináris klasszifikációra alkalmas változata kerülhessen felhasználásra a modellek betanításához és teszteléséhez. Ez az átalakítás úgy valósult meg, hogy a három különböző írisz fajtából csak kettőre vonatkozó adatok kerültek megtartásra, tehát a típusokat jelölő osztályok közül egy, az Iris virginica és ennek értelmében az összes ezen osztályba tartozó egyed eltávolításra került az adatkészletből. Erre pedig azért nyílt egyszerűen lehetőség, mert ahogy korábban már említésre került, egy írisz típus osztálya lineárisan elválasztható a másik kettő osztálytól (UC Irvine Machine Learning Repository, 2023). Mivel mindhárom osztályba 50 egyed tartozott, ezért a módosított, bináris klasszifikációra alkalmas Iris adatkészlet 100 egyedből áll, hiszen az eredeti 150 egyedből kiesett az 50 Iris virginica osztályba tartozó egyed, így azt eredményezve, hogy 50 Iris setosa és 50 Iris versicolor osztályba tartozó egyed maradjon az adatok között. Az így megkapott módosított, bináris klasszifikációra alkalmas Iris adatkészlet összefoglaló statisztikája az 1. táblázatban látható:

	Csészelevél hossza centiméterben	Csészelevél szélessége centiméterben	Sziromlevél hossza centiméterben	Sziromlevél szélessége centiméterben
Darabszám	100	100	100	100
Átlag	5,471	3,094	2,862	0,785
Szórás	0,642	0,4761	1,449	0,566
Minimum	4,300	2,000	1,000	0,100
25. percentilis	5,000	2,800	1,500	0,200
50. percentilis (medián)	5,400	3,050	2,450	0,800
75. percentilis	5,900	3,400	4,325	1,300
Maximum	7,000	4,400	5,100	1,800

1. táblázat: Az Iris adatkészlet módosított, bináris klasszifikációra alkalmas változatának összefoglaló statisztikája (Forrás: saját szerkesztés)

Az Iris adatkészlet összefoglaló statisztikáján kívül még az is megvizsgálásra került, hogy az egyes attribútumok milyen mértékben korrelálnak egymással, ezt pedig a 3. ábrán látható Python nyelv Matplotlib és Seaborn könyvtárainak segítségével létrehozott korrelációs hődiagram szemlélteti:



3. ábra: Korreláció a módosított, bináris klasszifikációra alkalmas Iris adatkészletben
(Forrás: saját szerkesztés)

A 3. ábrán található korrelációs hődiagram alapján látható, hogy számos attribútum erősen korrelál egymással, mégsem érdemes redukálni a számukat, hiszen ahogy már korábban említésre került, a csésze- és sziromlevelek hosszával és szélességével kapcsolatos tulajdonságok nem választhatók el lineárisan egymástól, valamint a kiinduló adatkészletben eleve kevés attribútum található, így a dimenzió csökkentés nem jelentene előnyt a kutatás során alkalmazott gépi tanuló algoritmusok betanításánál.

Pima Indians Diabetes adatbázis

A másik, nagyobb adatkészlet, amely jelen kutatás során felhasználásra került, a 768 egyedet tartalmazó Pima Indians Diabetes nevű adatkészlet, amely az adattudománnyal és gépi tanulással foglalkozó szakemberek körében széleskörűen elterjedt Kaggle weboldaláról került letöltésre csv formátumban. Az adatkészlet eredetileg az Országos Diabétesz és Emésztő- és Vesebetegségek Intézetétől (National Institute of Diabetes and Digestive and Kidney Diseases)

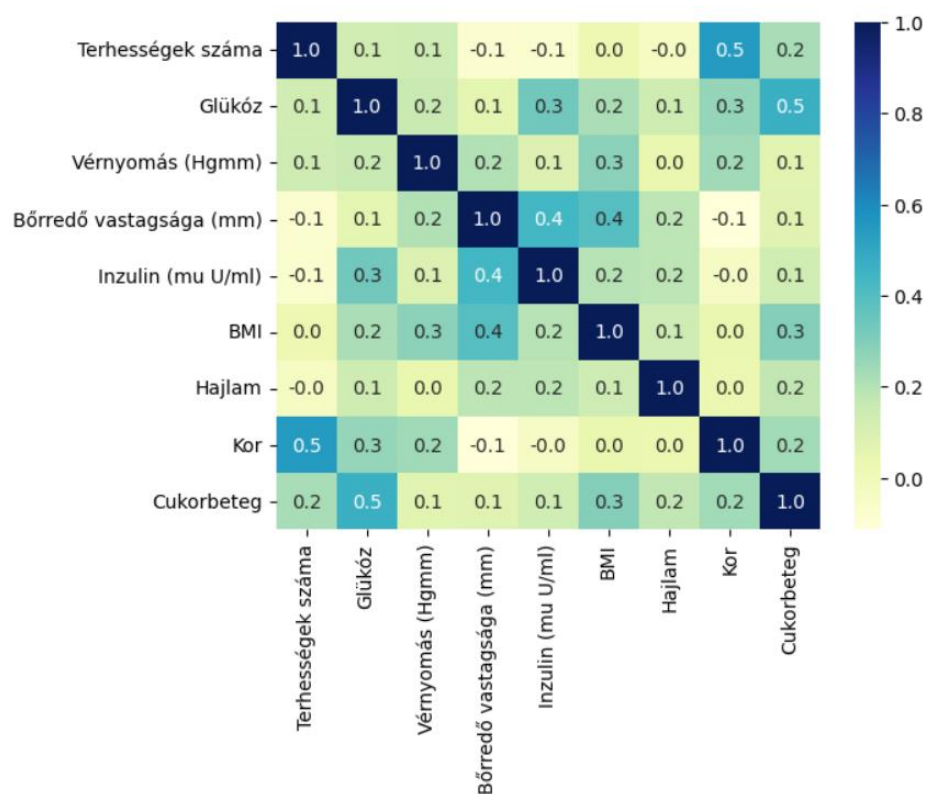
származik, amely az Amerikai Egyesült Államok Nemzeti Egészségügyi Intézetének a része. A Pima Indians Diabetes fő célja, hogy a benne tárolt tulajdonságok, diagnosztikai mérések alapján előre lehessen jelezni egy adott egyedről, hogy cukorbetegségben szenved-e vagy sem. Az adatkészlet bináris klasszifikációra alkalmas, ugyanis célváltozója csak a korábban említett két értéket veheti fel: cukorbeteg-e az adott egyed vagy egészséges (Kaggle, 2023).

A Pima Indians Diabetes az Iris-hez hasonlóan szintén egy ismert, széles körben elterjedt adatkészlet, amelyet gyakran használnak benchmark adatkészletként különböző gépi tanulási projektekben. Az adatkészletben pima indián örökségű 21 éves vagy annál idősebb nőkre vonatkozó adatok találhatók. A pima indiánok egy amerikai őslakos indián csoport, akik jellemzően Arizona államban és Mexikóban élnek, körükben pedig magas arányban fordul elő a diabetes mellitus, vagyis a cukorbetegség (Chang, Bailey, Xu, - Sun, 2022). A Pima Indians Diabetes nyolc attribútumból áll: a terhességek számából (amennyiben nem volt várandós az adott egyed, úgy ez az érték nulla), a plazma glükóz koncentrációból, amely egy két órán át tartó orális glükóz tolerancia teszt eredményeképp került előállításra (a 2. táblázatban Glükózként került feltüntetésre ez az attribútum), a diasztolés vérnyomás értékéből (Hgmm-ben mérve), a tricepsz bőrredő vastagságából (mm-ben mérve, a 2. táblázatban Bőrredő vastagsága (mm) névvel feltüntetve), a két órás inzulin vizsgálat eredményéből (mu U/ml-ben mérve a 2. táblázatban Inzulin (mu U/ml) névvel feltüntetve), a testtömegindexből (kilogrammban megadott testtömeg osztva a méterben mért testmagasság négyzetével, a lenti táblázatban BMI-ként feltüntetve), az úgynevezett Diabetes Pedigree Function-ből, amely a családi kortörténet alapján számolja ki a cukorbetegség megjelenésének valószínűségét (a 2. táblázatban Hajlamként feltüntetve), valamint az életkorból (években mérve). Az adatkészlet utolsó oszlopában pedig maga a diagnózis, tehát a célváltozó található, vagyis az, hogy az adott egyed a cukorbeteg vagy az egészséges osztályba került-e. Az adatkészletben 268 cukorbeteg, és 500 egészséges osztályba sorolt egyed található, így összességében elmondható, hogy nem egyenlően oszlik el a beteg és az egészséges egyedek száma (Kaggle, 2023). Mivel az adatkészletben nem szerepelnek hiányzó adatok, ezért az összes egyed felhasználható a kvantum- és a klasszikus gépi tanuló modellek betanítására és értékelésére. A 2. táblázatban látható a Pima Indians Diabetes adatkészlet összefoglaló statisztikája:

	Terhességek száma	Glükóz	Vérnyomás (Hgmm)	Bőrredő vastagsága (mm)	Inzulin (mu U/ml)	BMI	Hajlam	Kor
Darabszám	768	768	768	768	768	768	768	768
Átlag	3,845	120,895	69,105	20,536	79,799	31,993	0,472	33,241
Szórás	3,369	31,973	19,356	15,952	115,244	7,884	0,331	11,760
Minimum	0	0	0	0	0	0	0,078	21
25. percentilis	1	99	62	0	0	27,300	0,244	24
50. percentilis (medián)	3	117	72	23	30,500	32	0,373	29
75. percentilis	6	140,250	80	32	127,250	36,600	0,626	41
Maximum	17	199	122	99	846	67,100	2,420	81

2. táblázat: a Pima Indians Diabetes adatkészlet összefoglaló statisztikája (Forrás: saját szerkesztés)

A Pima Indians Diabetes adatkészlet esetében is az összefoglaló statisztika mellett még az egyes attribútumok közötti korrelációk is megvizsgálásra kerültek a Python nyelv Matplotlib és Seaborn könyvtárainak segítségével. Az attribútumok közötti korrelációt a 4. ábrán látható korrelációs hődiagram szemlélteti:



4. ábra: Korreláció a Pima Indians Diabetes adatkészletben (Forrás: saját szerkesztés)

A 4. ábrán található korrelációs hődiagramról leolvasható, hogy az egyes attribútumok önmagukon kívül semelyik másik attribútummal sem korrelálnak jelentős mértékben, hiszen legfeljebb 0,5 és legalább -0,1 a korreláció, amelyek közül egyik sem számít sem a pozitív, sem pedig a negatív irányú kapcsolatok közül erősnek. Így tehát a Pima Indians Diabetes adatkészlet esetében sincs szükség dimenziócsökkentésre a kutatás során alkalmazott gépi tanuló algoritmusok betanításának megkönnyítésére.

5. Eredmények bemutatása

A kvantum neurális hálózat és a kiválasztott klasszikus Support Vector Machine, k-Nearest Neighbors algoritmusok, valamint az automatizált gépi tanulás segítségével összeválogatott algoritmus-együttes bináris klasszifikációs probléma megoldására kijött eredményeire a kisebb méretű Iris, és a nagyobb méretű Pima Indians Diabetes adatkészlet esetén is megvizsgálásra került az osztályozás számos pontossági jellemzője, illetve a tanuláshoz és osztályozáshoz szükséges idők. Mindkét adatbázis esetén a kutatásban felhasznált összes gépi tanuló algoritmusra (mind a kvantum, mind pedig a klasszikus változatokra) külön-külön, az adott modell betanítása és a betanított modell értékelése összesen hússzor került elvégzésre a tanítás során a tanuló, az értékelés során pedig a tesztadatok felhasználásával. Mindegyik iteráció során 0-tól 19-ig más szám került megválasztásra a tanító és teszt adatok felosztása előtt az adatok összekeverését vezérlő paraméter (`random_state`) értékére (a tanító és teszt adatok szétválasztása mindegyik modell esetén 75%-25%-os arányban történt meg). Ezt követően minden iteráció során betanításra került az adott modell a tanító adatok segítségével, majd értékelésre a tesztadatok felhasználásával, végül pedig a pontossági jellemzők, valamint a tanuláshoz és az osztályozáshoz szükséges idők kerültek kiszámolásra. Így minden modell esetén adatbázisonként húsz különböző érték állt rendelkezésre a pontossági jellemzőkre, valamint a tanuláshoz és az osztályozáshoz szükséges időkre vonatkozóan. A modellek mérőszámokként húsz különböző értékeinek átlagai modellenként és mérőszámokként külön-külön kiszámolásra kerültek, majd a kvantum és klasszikus megoldások összehasonlításakor az egyes modellek így kijött átlagos pontossági jellemzői, valamint átlagos tanulási és osztályozási idői kerültek összehasonlításra egymással, hogy megválaszolhatóvá váljon jelen kutatás elsőszámú kérdése, mely szerint valóban megfigyelhető-e a kvantumfölény a mindenki számára szabadon elérhető Python alapú, a klasszikus gépi tanulást a kvantumossággal ötvöző szofver-megoldások esetén.

A fejezetben először röviden bemutatásra kerülnek az összehasonlításakor felhasznált pontossági jellemzők, valamint a betanuláshoz és az osztályozáshoz szükséges idők. Ezt

követően a kvantum neurális hálózat, majd a kiválasztott klasszikus megoldások eredményei kerülnek ismertetésre az Iris és a Pima Indians Diabetes adatkészletek esetén külön-külön. Végül adatbázisonként összehasonlításra és kiértékelésre kerülnek a kvantum és a klasszikus megoldások átlagos mérőszámai, valamint megvilágításra kerül, hogy jelen kutatás alapján mennyire figyelhető meg a kvantumfőlény hibrid megoldások alkalmazásakor, mennyire tud igazi megoldás lenni a kvantumosság hozzáadása a klasszikus gépi tanuláshoz.

5.1. Az összehasonlításhoz használt mérőszámok

A kvantum neurális hálózat és a kiválasztott klasszikus megoldások összehasonlításakor olyan pontossági jellemzők kerültek megvizsgálásra, mint az accuracy, a precision, a recall, az F1-érték, valamint a ROC görbe alatti terület (AUC). A következőkben a pontossági jellemzők, valamint a tanuláshoz és az osztályozáshoz szükséges idők bemutatásáról lesz szó. A felhasznált mérőszámok bemutatásához azonban érdemes először röviden kitérni a klasszifikációs modelleknél rendkívül gyakran használt konfúziós mátrixra. A konfúziós mátrixban feltüntetettek alapján kerültek kiszámolásra a továbbiakban bemutatott, a kutatás során felhasznált pontossági jellemzők. Az 5. ábra a konfúziós mátrixot szemlélteti:

		Modell érték	
		pozitív	negatív
Tényleges érték	pozitív	TP	FN
	negatív	FP	TN

5. ábra: Konfúziós mátrix (Forrás: saját szerkesztés)

Az 5. ábrán látható konfúziós mátrixban a pozitív értékek a célosztályba való tartozást jelentik, míg a negatívak a nem a célosztályba tartozó egyedeket képviselik. A True Positive (TP) a modell által sikeresen a célosztályba sorolt egyedek számát, míg a True Negative (TN) a modell által sikeresen nem a célosztályba sorolt egyedek számát jelenti. Ennek megfelelően a False Positive (FP) a modell által hibásan a célosztályba sorolt egyedek számát, a False Negative (FN) pedig a modell által hibásan nem a célosztályba sorolt egyedek számát takarja (Czakov, 2023; Ganguly, 2021; Handelman et al., 2019).

Accuracy

Az accuracy azt méri, hogy a teszt adatkészlet egyedei közül (mind a pozitív, mind pedig a negatív osztályba soroltakat is figyelembe véve) hányat sikerült a modellnek helyesen osztályoznia, vagyis valóban abba az osztályba sorolnia, amelybe az egyed a célváltozójának

címkéje alapján ténylegesen tartozik. Az accuracy kiszámolásához a helyes osztályba sorolt teszt egyedek száma kerül elosztásra az összes teszt egyed számával.

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{FP} + \text{TN} + \text{FN})}$$

Minél közelebb esik a modell accuracy-je a 100%-hoz, vélhetően annál pontosabb osztályzás elvégzésére lesz képes kiegyensúlyozott adatkészlet esetén. Az accuracy kapcsán fontos azonban megjegyezni, hogy amennyiben erősen kiegyensúlyozatlan az adatkészlet - tehát az egyik osztályba tartozik az adatok jelentős többsége, míg a másik osztályba alig sorolható egyed - például egy rendkívül ritka betegség esetén, ahol 1:999 a beteg és az egészséges egyedek aránya, ott nem érdemes az accuracy-ra hagyatkozni a modell pontosságának vizsgálatakor, hiszen ebben az esetben ha nem is végez osztályozást a modell, hanem minden egyedet egészségesnek sorol, akkor is 99,9 %-os accuracy állapítható meg. Így tehát hiába esik nagyon közel a modell 99,9%-os accuracy-je a 100%-hoz, akkor is haszontalan a modell, hiszen valójában nem végez klasszifikációt. Kiegyensúlyozatlan adatkészlet esetén ezért más pontossági mérőszámok eredményeit érdemes figyelembe venni, viszont kiegyenlített adatkészlet esetén, valamint abban az esetben, amikor az adatkészlet célváltozójának összes osztálya egyformán fontos, érdemes az accuracy-t is kiszámolni és felhasználni a modell pontosságának meghatározására (Czakon, 2023; Ganguly, 2021).

Precision

A precision azt mutatja meg, hogy a valóban pozitív egyedeket mennyire sorolja be helyesen a modell a célosztályba, vagyis mennyire helyesen végzi el az osztályozást, milyen a modell valós pozitív előrejelzési képessége. A precision kiszámolásához a modell által helyesen a pozitív osztályba sorolt egyedek száma kerül elosztásra a modell által a pozitív osztályba sorolt összes egyed számával.

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})}$$

Minél nagyobb értéke lesz az így kiszámolt precision-nek, annál helyesebben képes a modell a pozitív osztályba sorolni egyedeket. Minél alacsonyabb viszont a precision, annál többször sorolt a modell hibásan a pozitív osztályba egyedet. Internetes keresés esetén a precision az összes releváns találat aránya a keresőmotor által a beírt kulcsszavakra visszaadott teljes találati halmazban (Czakon, 2023; Ganguly, 2021; Handelman et al., 2019).

Recall

A recall vagy szenzitivitás azt mutatja meg, hogy az összes pozitív osztályba tartozó egyed közül hányat sorolt be a modell helyesen a pozitív osztályba.

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})}$$

Ha recall értéke magas, akkor a modell sok, a pozitív osztályba tartozó elemet helyesen sorol be. Az előbbi, internetes keresés példájánál maradva, a recall a keresőmotor által kiadott, releváns találatok aránya az összes létező (amit a keresőmotor feldobott és az is, amit nem) releváns találat számához képest, vagyis az eredmény teljességének a mértéke (Czakon, 2023; Ganguly, 2021; Handelman et al., 2019).

F1-érték

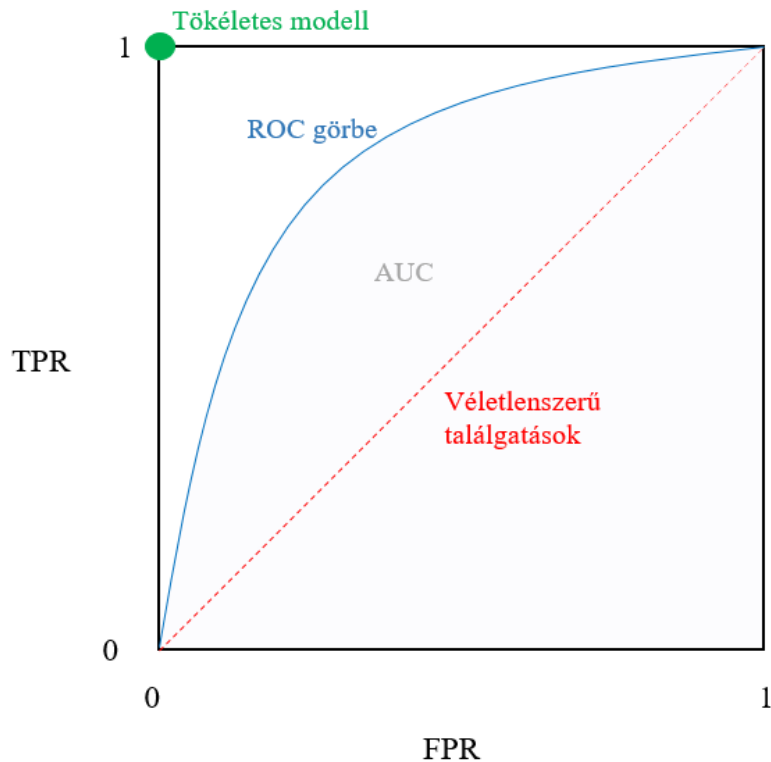
Az F1-érték a precision és recall mérőszámokat egyesíti egyetlen mérőszámmá, hiszen F1-érték ezen két mérőszám harmonikus átlaga.

$$F1 = 2 * \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}$$

Ha F1 értéke 1, akkor a modell tökéletesen osztályoz, ha F1 értéke 0, akkor pedig nem osztályoz, tehát a modell használhatatlan. Az F1-érték kiszámolása hasznos lehet minden bináris klasszifikációs feladatnál, főként azoknál, ahol inkább a pozitív osztályra kerül a fókusz (Czakon, 2023; Ganguly, 2021).

ROC görbe alatti terület (AUC)

A ROC (Receiver Operating Characteristics) görbe alatti terület, vagyis az Area Under the Curve (AUC) azt mutatja meg, hogy a modell mennyire képes szétválogatni a pozitív és negatív egyedeket az osztályozás elvégzése során. A ROC görbe minden pontja ugyanis egy lehetséges küszöbhez tartozó szenzitivitás – fals pozitív arány (TPR – FPR) értékpárt ábrázol. Minél magasabban emelkedik, minél inkább a bal felső sarokhoz illeszkedik, azaz nagyobb területet zár be az így ábrázolt görbe, annál jobb a modell, hiszen annál több küszöb esetén magas a szenzitivitás és alacsony a fals pozitív arány. A következő oldalon található 6. ábrán a ROC görbe és az AUC ábrázolása látható:



6. ábra: ROC görbe és AUC ábrázolása (Forrás: saját szerkesztés)

A görbe alatti terület (AUC) segítségével tehát egyetlen számmá tömöríthető a modell jósága anélkül, hogy küszöbértéket kellene választani. Az AUC pedig a következőképpen értelmezhető: egy tetszőleges pozitív és negatív esetet választva a modell az esetek hányad részében pontozza magasabbra a pozitív esetet, mint a negatívát. Ha az AUC értéke 1, akkor tökéletes a modell, ha viszont 0,5 az értéke, akkor a véletlenszerű találgatások hatékonyságával működik a modell, tehát nem jó. Továbbá érdemes kiemelni, hogy ezen mérőszám sem alkalmazható erősen kiegyenlített adatkészleteknél, hiszen ezen adatkészletek esetén a fals pozitív arány rendkívül alacsony lenne a valóban a negatív osztályba tartozó egyedek nagy száma miatt (Czakon, 2023; Handelman et al., 2019).

Tanuláshoz és osztályozáshoz szükséges idők

A tanuláshoz (illesztéshez) szükséges idő alatt azt az időt értjük, amelyre a modellnek szüksége van ahhoz, hogy a tanító adatbázisban olyan mintázatokot, összefüggéseket tárjon fel, amelyek alapján később, egy általa ismeretlen egyedre képes lesz a célváltozó megfelelő osztályába besorolni. Az osztályozáshoz szükséges idő pedig a korábban már a tanító adatokon betanított modell teszt adatokra történő illesztésének idejét takarja (Ganguly, 2021).

5.2. Kvantum neurális hálózat eredményei

A kvantum neurális hálózat eredményeinek bemutatása előtt fontos kiemelni, hogy Santanu Ganguly 2021-es Quantum Machine Learning: An Applied Approach könyvének „Chapter 8: Deep Quantum Learning” című fejezetének „Binary Classifier with PennyLane” részében szereplő kvantum neurális hálózat esetén az `n_epochs` nevű paraméter - amely azt állítja be, hogy hányszor szükséges az egész tanuló adatkészleten a tanuló algoritmusnak végigfutnia az összefüggések megfelelő feltárásához – értéke 10 volt, amely a bináris klasszifikációra alkalmas Iris adatkészlet esetén erős osztályozóképeséssel rendelkező modellt eredményezett, hiszen ahogy a 3. táblázat is mutatja a modell pontossági mutatói mind 0,9 (azaz 90%) felettiak voltak. A nagyobb méretű Pima Indians Diabetes adatkészlet esetén azonban már nem állítható ugyanez, ugyanis ahogyan már korábban a „4.2.1. Kvantum neurális hálózat” című alfejezetben is említésre került, a Pima Indians Diabetes esetén a(z) `n_epochs = 10` beállítás mellett a kvantum neurális hálózat nem tudta elvégezni a bináris osztályozást, minden egyedet eleve abba az osztályba sorolt, amelybe az adatbázis egyedeinek többsége (egészséges) tartozott. Ezt a 4. táblázat első sora is szemlélteti. Mivel a Pima Indians Diabetes adatbázis a bináris klasszifikációra alkalmas Iris adatbázisnál több mint hétszer nagyobb egyedszámmal rendelkezik, ezért érdemes `n_epochs` számának megnövelése mellett betanítani a kvantum neurális hálózatot a Pima Indians Diabetes esetén, hiszen egy nagyobb méretű adatbázis felhasználásakor előfordulhat, hogy többször szükséges a tanuló algoritmusnak végigfutnia az összefüggések megfelelő feltárásához az egész tanuló adatkészleten. Így az eredeti `n_epochs = 10` beállítás mellett 100, 200, 300, 400 és 500 értékek beállítása esetén is betanításra és értékelésre került a kvantum neurális hálózat, `n_epochs` beállításonként egyetlen alkalommal, a betanulást és értékelést megelőzően a tanító és teszt adatok felosztásakor az adatok összekeverését vezérlő paraméter (`random_state`) 0-ra állítása mellett. Ahhoz, hogy a kutatás konzisztens maradjon, ugyanezen beállítások mellett az Iris adatkészleten is betanításra és értékelésre került a kvantum neurális hálózat különböző `n_epochs` beállítások mellett. A 3. táblázat az Iris adatkészletet felhasználó kvantum neurális hálózat pontossági jellemzőinek, valamint a tanuláshoz és osztályozáshoz szükséges időinek alakulását mutatja különböző `n_epochs` paraméter-beállítások mellett. A modell betanítása és értékelése `n_epochs` beállításonként egyszer, a tanuló és teszt adatok felosztásakor `random_state = 0` paraméter-beállítás mellett történt meg.

n_epochs	Tanulási idő (mp)	Osztályozási idők (mp)	Accuracy	Precision	Recall	F1	AUC
10	14,135	0,205	0,96	1	0,933	0,966	0,967
100	174,179	0,410	1	1	1	1	1
200	294,446	0,277	1	1	1	1	1
300	540,884	0,252	1	1	1	1	1
400	679,039	0,249	1	1	1	1	1
500	686,737	0,254	1	1	1	1	1

3. táblázat: Az Iris adatbázison futtatott kvantum neurális hálózat betanulási ideje, osztályozási ideje és pontossági jellemzői különböző n_epochs paraméter-beállítások mellett (Forrás: saját szerkesztés)

A 3. táblázatból leolvasható, hogy n_epochs = 100 vagy az annál nagyobb kiválasztott értékek beállítása mellett a pontossági jellemzők értékeiben nincs különbség, az összes a maximálisan elérhető 1 értéket veszi fel, vagyis n_epochs = 10 paraméter-beállítás kivételével az összes többi kiválasztott n_epochs paraméter-beállítás mellett tökéletes a modell. A 100 vagy annál nagyobb különböző n_epochs paraméter megválasztások esetén mindössze a tanuláshoz és az osztályozáshoz szükséges idők különböznek egymástól, ez a különbség azonban csak a tanuláshoz szükséges idő esetén jelentős. Ebben az esetben ugyanis egyenes arányosság figyelhető meg az n_epochs paraméterre megadott érték és a tanuláshoz szükséges idő alakulása között. Ez nem meglepő, hiszen minél nagyobb érték kerül megadásra n_epochs paraméterre, annál többször fut végig a tanuló algoritmus az egész tanuló adatkészleten az összefüggések feltárásához, ezáltal pedig annál több időre lesz szüksége a modellnek a betanuláshoz. Mivel n_epochs = 100 vagy az annál nagyobb kiválasztott értékek beállítása mellett az Iris adatbázison implementált kvantum neurális hálózat tökéletes modellt eredményez, ezért az Iris adatbázis szempontjából teljesen lényegtelen, hogy a 100, 200, 300, 400 és 500 értékek közül a kutatás során a továbbiakban melyik érték kerül az n_epochs paraméter értékének megválasztására.

A 4. táblázat a Pima Indians Diabetes adatkészletet felhasználó kvantum neurális hálózat pontossági jellemzőinek, valamint a tanuláshoz és osztályozáshoz szükséges időinek alakulását mutatja különböző n_epochs paraméter-beállítások mellett. Az Iris adatkészlethez hasonlóan, a Pima Indians Diabetes esetén is a modell betanítása és értékelése n_epochs beállításonként egyszer, a tanuló és teszt adatok felosztásakor random_state = 0 paraméter-beállítás mellett történt meg.

n_epochs	Tanulási idő (mp)	Osztályozási idők (mp)	Accuracy	Precision	Recall	F1	AUC
10	312,248	4,554	0,645833	0	0	0	0,5
100	2789,613	4,840	0,708	0,658	0,368	0,472	0,631
200	5327,371	3,615	0,7083	0,667	0,353	0,462	0,628
300	8744,847	4,159	0,7083	0,667	0,353	0,462	0,628
400	10022,986	4,240	0,7083	0,667	0,353	0,462	0,628
500	13371,565	3,587	0,7083	0,667	0,353	0,462	0,628

4. táblázat: A Pima Indians Diabetes adatbázison futtatott kvantum neurális hálózat betanulási ideje, osztályozási ideje és pontossági jellemzői különböző n_epoch paraméter-beállítások mellett (Forrás: saját szerkesztés)

A 4. táblázat mutatja, hogy n_epochs = 10 paraméter-beállítás mellett még nem volt képes a teszt adatok osztályozására a Pima Indians Diabetes esetén a kvantum neurális hálózat, hiszen az egyszeri, a tanuló és teszt adatok felosztásakor random_state = 0 paraméter-beállítás melletti betanulást követően a modell kiértékelésekor körülbelül 0,65 (65%) jött ki az accuracy-re, ez pedig megegyezik az adatbázisban az egészséges osztályba tartozó egyedek arányával (500 egyed egészséges az összes 768-ból), továbbá 0,5 lett a ROC görbe alatti terület (AUC), amely a véletlenszerű találgatások hatékonyságával egyenlő. Ez alapján, valamint az így betanított modell a teszt adatok osztályba sorolásakor elvégzett predikcióinak kiíratását, megvizsgálását követően megállapíthatóvá vált, hogy n_epochs = 10 beállítás esetén a modell minden egyedet az egészséges osztályba sorolt, nem tudta az osztályozást elvégezni. Az n_epochs paraméterre megadott érték növelésével azonban javítható volt a modell osztályozási képessége, a 4. táblázat alapján látható, hogy már n_epochs = 100 paraméter-beállítás esetén is javultak a modell pontossági jellemzői, ugyanakkor ezen javulás a tanuláshoz szükséges idő jelentős megnövekedésével járt. Ahogy az Iris adatkészlet esetén is megfigyelhető volt, úgy a Pima Indians Diabetes-nél is látszik, hogy n_epochs növelésével a tanuláshoz szükséges idők is növekednek, hiszen ezen paraméter felelős annak meghatározásáért, hogy a tanulás során az összefüggések feltárásához hányszor fusson végig a tanuló algoritmus az egész tanuló adatkészleten, a többszöri futás pedig hosszabb tanulási időt eredményez, különösen nagyobb méretű adatbázisok esetén. Amennyiben n_epochs paraméterre nagyobb értékek kerültek megadásra, a kvantum neurális hálózat tanuláshoz szükséges ideje több órát (akár 5-6 órát) is igénybe vett egyszeri betanítás esetén is (ezt kiválóan szemléltetik például a 4. táblázat utolsó és utolsóelőtti sorai), viszont a pontossági jellemzők terén nem volt megfigyelhető javulás n_epochs növelésével n_epochs = 100 paraméter-beállításhoz képest a Pima Indians Diabetes esetén. Ennek egyik oka az lehet, hogy nagyobb adatbázis esetén az eredeti, Ganguly Quantum Machine Learning: An Applied Approach könyvében publikált kvantum neurális hálózatának

paraméter-beállításai közül nem csak az `n_epochs` paraméter megváltoztatása lehet szükséges, hanem például a kvantum neurális háló rétegeinek számát beállító `n_layers` nevű paraméter megnövelése is javíthat a pontossági mutatók alakulásán. Jelen kutatás fókuszában azonban nem a felhasznált kvantum neurális hálózat nagyobb adatkészletekre történő paraméter-optimalizálására irányult, így a kutatás során az `n_layers` paraméter eredeti 3-ra történő beállításán nem történt változás, csupán a korábban már említett `n_epochs` paraméter értéke változott meg az eredeti 10-ről 100-ra. A kutatás következő lépésében a megfelelő `n_epochs` paraméter-beállítás mellett egy húsz iterációs ciklusban kerül a kvantum neurális hálózat iterációnként különböző - a tanító és teszt adatok felosztása előtt az adatok összekeverését vezérlő - `random_state` paraméterek megadása mellett betanításra, illetve értékelésre az Iris és a Pima Indians Diabetes adatkészletek esetén is, ezért nem érdemes olyan értéket megválasztani `n_epochs`-ra, amely hatására a modell a pontossági jellemzők terén nem érne el javulásokat, viszont iterációnként több órán át végezné el a betanulást. Ennek hatására a kutatás során az Iris és a Pima Indians Diabetes esetén is `n_epochs = 100` paraméter-beállítás került alkalmazásra a kvantum neurális hálózatban a jelen „5.2. Kvantum neurális hálózat eredményei” alfejezet folytatásában bemutatott további betanulások és értékelések elvégzésekor.

Eredmények az Iris adatbázisra vonatkozóan

Az 5. táblázat az `n_epochs = 100` paraméter-beállítás mellett mutatja, hogy hogyan alakultak a húsz iterációs cikluson belül `random_state` paraméterre iterációnként más érték beállítása mellett a kvantum neurális hálózat betanuláshoz és osztályozáshoz szükséges idői, valamint a pontossági jellemzői az Iris adatbázis esetén.

<code>random_state</code>	Tanulási idő (mp)	Osztályozási idő (mp)	Accuracy	Precision	Recall	F1	AUC
0	174,180	0,410	1	1	1	1	1
1	175,493	0,243	1	1	1	1	1
2	175,086	0,283	1	1	1	1	1
3	174,876	0,253	1	1	1	1	1
4	174,250	0,239	1	1	1	1	1
5	174,598	0,236	1	1	1	1	1
6	174,978	0,243	1	1	1	1	1
7	174,979	0,404	1	1	1	1	1
8	173,927	0,257	1	1	1	1	1
9	175,091	0,245	1	1	1	1	1
10	174,772	0,273	1	1	1	1	1
11	175,182	0,244	1	1	1	1	1
12	174,139	0,247	1	1	1	1	1
13	174,828	0,242	1	1	1	1	1
14	172,413	0,246	1	1	1	1	1

15	172,976	0,242	1	1	1	1	1
16	172,499	0,243	1	1	1	1	1
17	173,318	0,236	1	1	1	1	1
18	172,096	0,395	1	1	1	1	1
19	172,361	0,265	1	1	1	1	1

5. táblázat: Az Iris adatbázison futtatott kvantum neurális hálózat betanulási ideje, osztályozási ideje és pontossági jellemzői $n_epoch = 100$ paraméter-beállítás mellett (Forrás: saját szerkesztés)

Az 5. táblázatban látható, hogy az Iris adatbázis esetén a kvantum neurális hálózat pontossági jellemzői mindahúsz, különböző – a betanulást és értékelést megelőzően a tanító és teszt adatok felosztásakor az adatok összekeverését vezérlő – `random_state` paraméter-beállítás melletti modell betanítás és értékelés esetén ugyanúgy a maximálisan elérhető 1-et érik el, tehát a modell minden `random_state` paraméter-beállítás mellett tökéletesnek mondható a pontossági jellemzők alapján. A tanulási idők pedig iterációnként körülbelül 2, majdnem 3 percet vettek igénybe. Így a kvantum neurális hálózat tanuláshoz szükséges ideje már a kisebb méretű Iris adatkészlet esetén is lényegesen hosszabbnak bizonyult a klasszikus SVM és k-NN modellekhez képest, összehasonlításukra részletesebben azonban a „5.4. Kiértékelés – Kvantum és klasszikus változatok összehasonlítása” című alfejezetben kerül sor.

Eredmények a Pima Indians Diabetes adatbázisra vonatkozóan

A 6. táblázat azt ábrázolja, hogy az $n_epochs = 100$ paraméter-beállítás mellett, hogyan alakultak a húsz iterációs cikluson belül `random_state` paraméterre iterációnként más érték beállítása mellett a kvantum neurális hálózat betanuláshoz és osztályozáshoz szükséges idői valamint a pontossági jellemzői a Pima Indians Diabetes adatbázis esetén.

<code>random_state</code>	Tanulási idő (mp)	Osztályozási idő (mp)	Accuracy	Precision	Recall	F1	AUC
0	2789,613	4,840	0,708	0,658	0,368	0,472	0,631
1	2817,366	3,732	0,714	0,705	0,425	0,530	0,658
2	2796,469	3,764	0,708	0,588	0,462	0,517	0,648
3	2793,310	4,880	0,745	0,783	0,480	0,595	0,697
4	2791,611	3,834	0,740	0,656	0,350	0,457	0,633
5	2797,285	3,716	0,823	0,717	0,611	0,660	0,758
6	2783,149	3,696	0,703	0,809	0,442	0,571	0,678
7	2298,605	3,193	0,760	0,757	0,431	0,549	0,680
8	2315,471	3,396	0,750	0,784	0,420	0,547	0,678
9	2245,589	3,179	0,786	0,745	0,547	0,631	0,727
10	2241,728	3,128	0,797	0,795	0,500	0,614	0,719
11	2242,130	3,577	0,745	0,667	0,419	0,515	0,660
12	2243,103	3,142	0,776	0,833	0,493	0,619	0,718
13	2241,013	3,327	0,724	0,767	0,434	0,555	0,674
14	2248,164	3,172	0,740	0,657	0,377	0,479	0,643
15	2245,363	3,112	0,719	0,542	0,448	0,491	0,642
16	2240,404	3,121	0,750	0,604	0,500	0,547	0,679

17	3007,787	4,288	0,766	0,667	0,448	0,536	0,676
18	3012,633	4,100	0,688	0,667	0,442	0,531	0,647
19	3027,768	4,039	0,750	0,791	0,466	0,586	0,695

6. táblázat: A Pima Indians Diabetes adatbázison futtatott kvantum neurális hálózat betanulási ideje, osztályozási ideje és pontossági jellemzői $n_epoch = 100$ paraméter-beállítások mellett (Forrás: saját szerkesztés)

A 6. táblázatról leolvasható, hogy a kvantum neurális hálózat az Iris-nél jelentősen nagyobb, több egyedet és attribútumot tartalmazó Pima Indians Diabetes adatbázis esetén $n_epochs = 100$ paraméter-beállítás mellett hússzor, különböző $random_state$ paraméter-értékek beállításával történő betanítás és értékelés során nem eredményez tökéletes modelleket, hiszen az egyes accuracy-k jellemzően 0,7 – 0,8 (70-80%) között, a precision-ök 0,6 - 0,8 (60-80%), a recall-ok 0,3- 0,5, az F1-értékek 0,4-0,6, a AUC pedig 0,6 és 0,7 között alakulnak. Az így kijött pontossági jellemzők bár az Iris adatbázis felhasználásával betanított és értékelt kvantum neurális hálózat pontossági jellemzőihez képest gyengébbek, a Pima Indians Diabetes adatbázison betanított és értékelt klasszikus megoldásokkal szemben azonban nem marad le, sőt bizonyos pontossági jellemzők esetén meg is előz néhány klasszikus modellt. Erről bővebben a „5.4. Kiértékelés – Kvantum és klasszikus változatok összehasonlítása” című alfejezetben lesz szó. Ami a Pima Indians Diabetes-en futtatott kvantum neurális hálózat tanuláshoz és osztályozáshoz szükséges időinek alakulását illeti, iterációnként körülbelül 35-50 percet vett igénybe a modell betanulása és 3-4 másodpercet az értékelése. Mindkét idő esetén jelentősen alulmarad a kvantum neurális hálózat a Pima Indians Diabetes-en alkalmazott klasszikus SVM, k-NN és automatizált gépi tanulás segítségével összeválogatott algoritmus-együttes tanuláshoz és osztályozáshoz szükséges időivel szemben. Különösen a tanuláshoz szükséges idők esetén teljesít jelentősen gyengébben a kvantum neurális hálózat a klasszikus megoldásokkal szemben, ugyanis lényegesen hosszabb időbe telik a kvantummodell betanulása mindahusz megvizsgált esetben a klasszikus modellekénél. Ez szintén a „5.4. Kiértékelés – Kvantum és klasszikus változatok összehasonlítása” című alfejezetben kerül részletezésre.

5.3. Kiválasztott klasszikus gépi tanuló algoritmusok eredményei

Ebben az alfejezetben a jelen kutatás során kiválasztott és implementált klasszikus SVM, k-NN és az automatizált gépi tanulás segítségével összeválogatott algoritmus-együttes pontossági jellemzőire, valamint a tanuláshoz és az osztályozáshoz szükséges időire kijött eredmények kerülnek bemutatásra, először az Iris adatbázis, majd a Pima Indians Diabetes esetén.

Eredmények az Iris adatbázisra vonatkozóan

A 7. táblázat az Iris adatbázisra felhasznált Support Vector Machine algoritmus összesen húsz alkalommal, különböző random_state paraméter-beállítások mellett történő betanítása és értékelése során kijött pontossági jellemzőinek, valamint a tanuláshoz és osztályozáshoz szükséges időinek alakulását szemlélteti. A modell mindahúsz betanulása és értékelése során lineáris kernel beállítás került alkalmazásra, ebből kifolyólag gamma együttható értékére pedig az alapértelmezett 'scale' került megválasztása. C szabályzási paraméter-beállítás minden esetben 1 volt.

random_state	Tanulási idő (mp)	Osztályozási idő (mp)	Accuracy	Precision	Recall	F1	AUC
0	0,003	0,002	1	1	1	1	1
1	0,002	0,001	1	1	1	1	1
2	0,002	0,001	1	1	1	1	1
3	0,002	0,001	1	1	1	1	1
4	0,002	0,001	1	1	1	1	1
5	0,002	0,001	1	1	1	1	1
6	0,002	0,001	1	1	1	1	1
7	0,002	0,001	1	1	1	1	1
8	0,002	0,001	1	1	1	1	1
9	0,002	0,001	1	1	1	1	1
10	0,002	0,001	1	1	1	1	1
11	0,002	0,001	1	1	1	1	1
12	0,002	0,002	1	1	1	1	1
13	0,003	0,001	1	1	1	1	1
14	0,002	0,002	1	1	1	1	1
15	0,002	0,001	1	1	1	1	1
16	0,002	0,001	1	1	1	1	1
17	0,003	0,001	1	1	1	1	1
18	0,002	0,001	1	1	1	1	1
19	0,002	0,001	1	1	1	1	1

7. táblázat: A klasszikus SVM pontossági jellemzői, valamint a tanuláshoz és az osztályozáshoz szükséges idők az Iris adatbázis esetén (Forrás: saját szerkesztés)

A 7. táblázaton látható, hogy az Iris adatbázis esetén az SVM mindahúsz betanított és kiértékelt modellje a pontossági jellemzők alapján tökéletesnek mondható, hiszen az accuracy, a precision, a recall, az F1-érték és az AUC is a maximálisan elérhető 1 értéket veszi fel minden alkalommal. Továbbá leolvasható az is, hogy az SVM tanuláshoz és osztályozáshoz szükséges idői ezen 100 egyeddel rendelkező Iris adatkészlet esetén rendkívül alacsonyak, az egy másodperc töredékét érik el csupán.

A 8. táblázat az Iris adatbázisra alkalmazott k-Nearest Neighbors algoritmus húsz alkalommal történő betanulása és értékelése során megfigyelhető pontossági jellemzőket, illetve a tanuláshoz és osztályozáshoz szükséges időket mutatja be. A k-NN algoritmus

implementálásakor az optimális k paraméter-érték megkeresése után, mindahusz esetben k értékére 6 került megválasztásra, így mindig az adott egyedhez 6 legközelebb eső szomszéd osztályai alapján történt meg az egyed valamelyik osztályba sorolása.

random_state	Tanulási idő (mp)	Osztályozási idő (mp)	Accuracy	Precision	Recall	F1	AUC
0	0,006	0,006	1	1	1	1	1
1	0,002	0,004	1	1	1	1	1
2	0,002	0,004	1	1	1	1	1
3	0,001	0,006	1	1	1	1	1
4	0,002	0,003	1	1	1	1	1
5	0,002	0,003	1	1	1	1	1
6	0,002	0,003	1	1	1	1	1
7	0,002	0,003	1	1	1	1	1
8	0,002	0,006	1	1	1	1	1
9	0,002	0,003	1	1	1	1	1
10	0,002	0,003	1	1	1	1	1
11	0,004	0,005	1	1	1	1	1
12	0,002	0,003	1	1	1	1	1
13	0,002	0,003	1	1	1	1	1
14	0,002	0,004	1	1	1	1	1
15	0,002	0,004	1	1	1	1	1
16	0,002	0,003	1	1	1	1	1
17	0,002	0,004	1	1	1	1	1
18	0,002	0,005	1	1	1	1	1
19	0,002	0,003	1	1	1	1	1

8. táblázat: A klasszikus k -NN pontossági jellemzői, valamint a tanuláshoz és az osztályozáshoz szükséges idők az Iris adatbázis esetén (Forrás: saját szerkesztés)

A 8. táblázat szemlélteti, hogy az Iris adatbázis esetén az SVM-hez hasonlóan, a k -NN mindahusz betanított és kiértékelt modellje is a pontossági jellemzők alapján tökéletesnek mondható, hiszen minden esetben mindegyik pontossági jellemző 1 értéket vett fel. Az Iris adatbázis esetén az SVM-hez hasonlóan, a k -NN algoritmus alkalmazásakor is rendkívül rövid idő szükséges a modell tanulására és értékelésére mindahusz alkalommal.

A 9. táblázaton az Iris adatbázis felhasználása mellett húsz alkalommal betanított és értékelte automatizált gépi tanulás segítségével összeválogatott algoritmus-együttes pontossági jellemzőinek, valamint tanuláshoz és osztályozáshoz szükséges időinek alakulása látható. Az automatizált gépi tanulás segítségével összeválogatott gépi tanuló-algoritmus együttes mindahusz esetben a „4.2.2. Klasszikus gépi tanuló algoritmusok” című alfejezet „Auto Machine Learning (AutoML)” részében bemutatott paraméter-beállítások megadása mellett került implementálásra.

random_state	Tanulási idő (mp)	Osztályozási idő (mp)	Accuracy	Precision	Recall	F1	AUC
0	127,159	0,717	1	1	1	1	1
1	118,551	0,625	1	1	1	1	1
2	115,703	3,729	1	1	1	1	1
3	114,723	0,807	1	1	1	1	1
4	116,923	0,935	1	1	1	1	1
5	118,708	0,696	1	1	1	1	1
6	116,794	0,679	1	1	1	1	1
7	119,536	0,626	1	1	1	1	1
8	115,949	0,788	1	1	1	1	1
9	116,930	0,619	1	1	1	1	1
10	114,523	0,814	1	1	1	1	1
11	118,797	0,574	1	1	1	1	1
12	115,347	0,710	1	1	1	1	1
13	122,118	1,006	1	1	1	1	1
14	116,043	0,715	1	1	1	1	1
15	124,702	0,800	1	1	1	1	1
16	120,853	2,854	1	1	1	1	1
17	116,152	1,509	1	1	1	1	1
18	116,361	0,979	1	1	1	1	1
19	117,831	0,661	1	1	1	1	1

9. táblázat: Az automatikus gépi tanulás segítségével létrehozott algoritmus-együttes pontossági jellemzői, valamint a tanuláshoz és az osztályozáshoz szükséges idők az Iris adatbázis esetén (Forrás: saját szerkesztés)

A 9. táblázat bemutatja, hogy az Iris adatbázis esetén az SVM és k-NN megoldásokhoz hasonlóan az automatizált gépi tanulás segítségével összeválogatott algoritmus-együttes (AutoML) is tökéletes osztályzásra képes modelleket eredményez mindahusz esetben a pontossági jellemzőkre kijött értékek alapján. Észrevehető azonban, hogy a tanuláshoz és az osztályozáshoz több időre volt szüksége az automatizált gépi tanulás segítségével összeválogatott algoritmus-együttesnek az SVM és k-NN modellekhez képest. Különösen hosszabb idő volt szükséges a betanuláshoz, ennek pedig az az oka, hogy az AutoML paraméter-beállítások megadásakor a végrehajtásra szánt időre (time_left_for_this_task) 120 másodperc került beállításra, amely azt jelenti, hogy maximum 2 perc fordítható az Iris adatbázis felhasználása esetén a legmegfelelőbb gépi tanuló algoritmus-együttes megkeresésére mindahusz betanulás során. Az AutoML esetén tehát nincs értelme figyelembe venni a tanuláshoz szükséges időkre kijött eredményeket, és ezen mérőszám mentén összehasonlítani az AutoML teljesítményét akár a klasszikus SVM és k-NN megoldásokéval, akár a kvantum neurális hálózatéval, hiszen az AutoML esetén szinte előre beállításra került, hogy mennyi idő szükséges a betanuláshoz.

Eredmények a Pima Indians Diabetes adatbázisra vonatkozóan

A 10. táblázat a Pima Indians Diabetes adatbázis felhasználása mellett implementált Support Vector Machine algoritmus összesen húsz alkalommal, különböző random_state paraméter-beállítások mellett történő betanítása és értékelése során kijött pontossági jellemzőinek, valamint a tanuláshoz és osztályozáshoz szükséges időinek alakulását szemlélteti. Akárcsak az Iris adatbázis esetén, a Pima Indians Diabetes felhasználásakor is a modell mindahúsz betanulása és értékelése során lineáris kernel beállítás került alkalmazásra, ebből kifolyólag gamma együttható értékére pedig az alapértelmezett 'scale' került megválasztásra. C szabályzási paraméter-beállítás minden esetben 100 volt.

random_state	Tanulási idő (mp)	Osztályozási idő (mp)	Accuracy	Precision	Recall	F1	AUC
0	102,256	0,004	0,781	0,685	0,597	0,638	0,733
1	81,389	0,008	0,771	0,719	0,594	0,651	0,732
2	140,474	0,004	0,766	0,644	0,500	0,563	0,690
3	98,646	0,006	0,734	0,738	0,563	0,638	0,710
4	81,303	0,006	0,823	0,767	0,697	0,730	0,793
5	199,709	0,005	0,745	0,575	0,700	0,632	0,733
6	54,911	0,004	0,771	0,691	0,585	0,633	0,725
7	59,764	0,003	0,786	0,738	0,643	0,687	0,756
8	134,721	0,004	0,781	0,741	0,614	0,672	0,746
9	125,452	0,004	0,750	0,662	0,623	0,642	0,722
10	70,366	0,005	0,719	0,660	0,493	0,565	0,672
11	173,084	0,004	0,750	0,714	0,455	0,556	0,680
12	76,656	0,004	0,781	0,746	0,620	0,677	0,748
13	76,101	0,004	0,755	0,725	0,529	0,612	0,707
14	71,496	0,003	0,734	0,643	0,429	0,514	0,656
15	97,241	0,004	0,724	0,593	0,508	0,547	0,669
16	103,168	0,006	0,797	0,759	0,612	0,678	0,754
17	204,742	0,004	0,776	0,750	0,592	0,661	0,738
18	78,468	0,003	0,719	0,667	0,457	0,542	0,663
19	56,674	0,004	0,786	0,704	0,603	0,650	0,740

10. táblázat: A klasszikus SVM pontossági jellemzői, valamint a tanuláshoz és az osztályozáshoz szükséges idők a Pima Indians Diabetes adatbázis esetén (Forrás: saját szerkesztés)

Ahogy a 10. táblázat is ábrázolja, a Pima Indians Diabetes adatbázis felhasználásakor az SVM húsz betanított és kiértékelt modelljeire kijött pontossági jellemzői alapján egyszer sem figyelhető meg tökéletes modell, hiszen az accuracy 0,7-0,8 (70-80%) között, a precision 0,6-0,7 (60-70%) között, a recall többnyire 0,5-0,7 között (néhány kivétellel), az F1-érték 0,5-0,7 között, az AUC pedig 0,65-0,8 között alakult. Ezen kívül a 10. táblázatról leolvasható az is, hogy az SVM tanuláshoz szükséges idői ezen 768 egyeddel és 8 attribútummal rendelkező Pima Indians Diabetes adatkészlet esetén lényegesen hosszabbak voltak, mint az Iris adatbázis (7. táblázat) esetén. Nagyobb méretű adatbázis felhasználásakor tehát gyengébb pontossági

jellemzők és lényegesen hosszabb tanuláshoz szükséges idők voltak megfigyelhetők a klasszikus SVM esetén.

A 11. táblázat a Pima Indians Diabetes adatbázisra alkalmazott k-Nearest Neighbors algoritmus húsz alkalommal történő betanulása és értékelése során megfigyelhető pontossági jellemzőket, illetve a tanuláshoz és osztályozáshoz szükséges időket mutatja be. Ugyanúgy ahogy az Iris adatbázisnál is, a Pima Indians Diabetes esetén is a k-NN algoritmus implementálásakor az optimális k paraméter-érték megkeresése után, mindahúsz esetben k értékére 6 került megválasztásra, így mindig az adott egyedhez 6 legközelebb eső szomszéd osztályai alapján történt meg az egyed valamelyik osztályba sorolása.

random_state	Tanulási idő (mp)	Osztályozási idő (mp)	Accuracy	Precision	Recall	F1	AUC
0	0,003	0,009	0,760	0,667	0,516	0,582	0,697
1	0,002	0,012	0,760	0,745	0,507	0,603	0,705
2	0,002	0,010	0,771	0,684	0,448	0,542	0,679
3	0,002	0,008	0,651	0,667	0,325	0,437	0,604
4	0,002	0,008	0,760	0,692	0,545	0,610	0,709
5	0,002	0,009	0,734	0,596	0,467	0,523	0,661
6	0,002	0,011	0,760	0,702	0,508	0,589	0,699
7	0,003	0,008	0,724	0,707	0,414	0,523	0,658
8	0,002	0,008	0,729	0,705	0,443	0,544	0,668
9	0,002	0,007	0,698	0,622	0,406	0,491	0,634
10	0,002	0,014	0,703	0,652	0,423	0,513	0,645
11	0,002	0,008	0,729	0,675	0,409	0,509	0,653
12	0,002	0,008	0,682	0,600	0,423	0,496	0,629
13	0,002	0,008	0,724	0,681	0,457	0,547	0,667
14	0,002	0,013	0,729	0,690	0,317	0,435	0,624
15	0,002	0,012	0,745	0,667	0,444	0,533	0,668
16	0,002	0,007	0,755	0,778	0,418	0,544	0,677
17	0,002	0,013	0,745	0,789	0,423	0,550	0,678
18	0,002	0,008	0,734	0,744	0,414	0,532	0,666
19	0,002	0,008	0,776	0,717	0,524	0,606	0,712

11. táblázat: A klasszikus k-NN pontossági jellemzői, valamint a tanuláshoz és az osztályozáshoz szükséges idők a Pima Indians Diabetes adatbázis esetén (Forrás: saját szerkesztés)

A 11. táblázatban látható pontossági jellemzők alapján megállapítható, hogy a Pima Indians Diabetes adatbázis esetén a k-NN algoritmus sem tudott tökéletes modellt létrehozni a húsz megvizsgált eset közül egyszer sem. Pontossági jellemzői hasonlóan alakultak a 10. táblázatban bemutatott SVM pontossági jellemzőihez, a tanuláshoz szükséges idői azonban jelentősen rövidebbek voltak, mint az SVM-é. Megállapítható tehát, hogy az Iris adatkészlethez képest sokkal több egyeddel és attribútummal rendelkező Pima Indians Diabetes esetén a k-NN algoritmus a pontossági jellemzők terén jelentősen gyengébben teljesít, mint a kisebb Iris felhasználásakor, a tanuláshoz szükséges idők esetén azonban nem állítható ugyanez.

A 12. táblázat a Pima Indians Diabetes adatbázis felhasználása mellett húsz alkalommal betanított és értékelt automatizált gépi tanulás segítségével összeválogatott algoritmus-együttes pontossági jellemzőinek, valamint tanuláshoz és osztályozáshoz szükséges időinek alakulását ábrázolja. Az automatizált gépi tanulás segítségével összeválogatott gépi tanuló-algoritmus együttes a Pima Indians Diabetes felhasználásakor is mindahúsz esetben a „4.2.2. Klasszikus gépi tanuló algoritmusok” című alfejezet „Auto Machine Learning (AutoML)” részében bemutatott paraméter-beállítások megadása mellett került implementálásra.

random_state	Tanulási idő (mp)	Osztályozási idő (mp)	Accuracy	Precision	Recall	F1	AUC
0	114,956	0,854	0,781	0,700	0,565	0,625	0,725
1	114,646	1,054	0,786	0,741	0,623	0,677	0,751
2	119,549	1,143	0,776	0,667	0,517	0,583	0,703
3	119,636	0,714	0,740	0,734	0,588	0,653	0,718
4	117,939	0,193	0,802	0,726	0,682	0,703	0,773
5	117,965	0,523	0,734	0,571	0,600	0,585	0,698
6	115,021	0,561	0,766	0,679	0,585	0,628	0,721
7	118,651	0,743	0,802	0,786	0,629	0,698	0,765
8	116,030	0,610	0,734	0,661	0,557	0,605	0,697
9	117,517	0,892	0,724	0,633	0,551	0,589	0,686
10	114,518	0,869	0,734	0,708	0,479	0,571	0,682
11	115,697	0,443	0,766	0,706	0,545	0,615	0,713
12	119,269	0,816	0,760	0,778	0,493	0,603	0,705
13	114,414	0,612	0,766	0,705	0,614	0,656	0,733
14	118,613	0,629	0,755	0,690	0,460	0,552	0,680
15	118,267	0,631	0,776	0,679	0,603	0,639	0,732
16	115,464	0,391	0,766	0,720	0,537	0,615	0,713
17	117,306	0,824	0,766	0,750	0,549	0,634	0,721
18	123,955	0,634	0,750	0,789	0,429	0,556	0,681
19	115,608	0,844	0,792	0,756	0,540	0,630	0,727

12. táblázat: Az automatikus gépi tanulás segítségével létrehozott algoritmus-együttes pontossági jellemzői, valamint a tanuláshoz és az osztályozáshoz szükséges idők a Pima Indians Diabetes adatbázis esetén (Forrás: saját szerkesztés)

A 12. táblázat alapján látható, hogy a Pima Indians Diabetes adatbázis esetén az SVM és k-NN megoldásokhoz hasonló értékeket eredményez az egyes pontossági jellemzőkre az automatizált gépi tanulás segítségével összeválogatott algoritmus-együttes (AutoML) is. A tanuláshoz szükséges időkben azonban a Pima Indians Diabetes esetén is lényegesen hosszabb időre volt szüksége az AutoML-nek az SVM és k-NN algoritmusokkal szemben. Ennek oka az AutoML betanulásakor megadott, korábban már említett „time_left_for_this_task” paraméter-beállítás, amely a végrehajtásra szánt időre 120 másodpercet állít be mindahúsz esetben, ezáltal biztosítva, hogy maximum 2 perc legyen a Pima Indians Diabetes adatbázis felhasználása esetén is a legmegfelelőbb gépi tanuló algoritmus-együttes megkeresésére fordítható idő. Az

AutoML esetén tehát a Pima Indians Diabetes adatbázis felhasználásakor sincs értelme figyelembe venni a tanuláshoz szükséges időkre kijött eredményeket, és ezen mérőszám mentén elvégezni az egyéb megoldásokkal történő összehasonlításokat, hiszen ahogy korábban már kiemelésre került az Iris adatbázis kapcsán is, az AutoML esetén szinte előre beállításra került, hogy mennyi idő szükséges a betanuláshoz.

5.4. Kiértékelés - Kvantum és klasszikus változatok összehasonlítása

Ez az alfejezet az előzőekben bemutatott kvantum neurális hálózat és a klasszikus SVM, k-NN, valamint az automatizált gépi tanulás segítségével összeválogatott algoritmus-együttes húsz alkalommal történő betanulása és értékelése során kiszámolt pontossági jellemzők, illetve a tanuláshoz és az osztályozáshoz szükséges idők átlagos értékei alapján hasonlítja össze a kvantum és klasszikus megoldások pontosságát, valamint teljesítményét az Iris és a Pima Indians Diabetes adatbázisokra külön-külön.

Eredmények összehasonlítása az Iris adatbázisra vonatkozóan

A 13. táblázaton a kvantum és a klasszikus modellek húsz betanulás és értékelés során kiszámolt tanuláshoz és osztályozáshoz szükséges időinek, valamint pontossági jellemzőinek az átlaga látható mindegyik modellre külön bemutatva, az Iris adatbázis felhasználása mellett.

Modell	Tanulási idő (mp)	Osztályozási idő (mp)	Accuracy	Precision	Recall	F1	AUC
Kvantum neurális hálózat	174,102	0,272	1	1	1	1	1
Klasszikus SVM	0,002	0,001	1	1	1	1	1
Klasszikus k-NN	0,002	0,004	1	1	1	1	1
Klasszikus AutoML	118,185	1,042	1	1	1	1	1

13. táblázat: Az Iris adatbázisra vonatkozóan az egyes modellek átlagos tanulási és osztályozási idői, valamint pontossági jellemzői (Forrás: saját szerkesztés)

A 13. táblázatról leolvasható, hogy az átlagos pontossági jellemzők alapján a kvantum neurális hálózat egyáltalán nem maradt le a klasszikus modellekkel szemben, ugyanúgy tökéletes osztályozásra képes modellt eredményezett a bináris klasszifikációs probléma megoldására az Iris adatbázis esetén, mint a klasszikus SVM, k-NN és AutoML modellek. A kvantum neurális hálózat átlagos tanuláshoz szükséges ideje azonban lényegesen hosszabb volt már ezen kisebb, 100 egyeddel rendelkező Iris adatbázis esetén is, mint a klasszikus SVM és k-NN modelleké. Az „5.3. Kiválasztott klasszikus gépi tanuló algoritmusok eredményei” című

fejezet AutoML részében kifejtett okok miatt a modellek átlagos tanuláshoz szükséges időinek összehasonlításakor az automatizált gépi tanulás segítségével létrejött algoritmus-együttes átlagos tanuláshoz szükséges idejét nem érdemes figyelembe venni. Míg a klasszikus SVM és k-NN modellek átlagosan egy másodpercnél is jóval kevesebb időt igényeltek a betanuláshoz, addig a kvantum neurális hálózathoz átlagosan több mint 2, majdnem 3 perc volt szükséges ugyanehhez.

Így már a kisebb méretű Iris adatbázis esetén is megállapítható, hogy a jelen kutatás során alkalmazott kvantum neurális hálózat, amely egy hibrid megoldás – hiszen implementálásakor a klasszikus gépi tanuláshoz került hozzáadásra a kvantumosság – nem figyelhető meg a technológiailag megjósolt kvantumfölény. Bár a pontossági jellemzők terén az Iris adatbázis esetén elérte a kvantum neurális hálózat a klasszikus megoldások pontosságait, a betanítási ideje azonban jelentősen hosszabb volt, mint a klasszikusoké.

Eredmények összehasonlítása a Pima Indians Diabetes adatbázisra vonatkozóan

A 14. táblázat a nagyobb méretű, Pima Indians Diabetes adatbázis felhasználásakor a kvantum és klasszikus modellek húsz betanulás és értékelés során kiszámolt tanuláshoz és osztályozáshoz szükséges időinek, valamint pontossági jellemzőinek átlagait mutatja, mindegyik modellre külön-külön.

Modell	Tanulási idő (mp)	Osztályozási idő (mp)	Accuracy	Precision	Recall	F1	AUC
Kvantum neurális hálózat	2558,928	3,662	0,745	0,709	0,453	0,550	0,677
Klasszikus SVM	104,331	0,004	0,763	0,696	0,571	0,624	0,718
Klasszikus k-NN	0,002	0,009	0,734	0,689	0,442	0,535	0,667
Klasszikus AutoML	117,251	0,699	0,764	0,709	0,557	0,621	0,716

14. táblázat: A Pima Indians Diabetes adatbázisra vonatkozóan az egyes modellek átlagos tanulási és osztályozási idői, valamint pontossági jellemzői (Forrás: saját szerkesztés)

A 14. táblázat alapján megállapítható, hogy az átlagos pontossági jellemzők terén a kvantum neurális hálózat megközelítette a klasszikus modelleket a Pima Indians Diabetes esetén is, sőt a klasszikus k-NN-nél erősebb pontossági jellemzőket eredményezett. A kvantum neurális hálózat átlagos tanuláshoz szükséges ideje azonban lényegesen hosszabb volt a Pima Indians Diabetes adatbázis esetén is, mint a klasszikus SVM és k-NN modelleké. Itt is fontos kiemelni, hogy az „5.3. Kiválasztott klasszikus gépi tanuló algoritmusok eredményei” című fejezet AutoML részében kifejtett okok miatt a modellek átlagos tanuláshoz szükséges időinek

összehasonlításakor az automatizált gépi tanulás segítségével létrejött algoritmus-együttes átlagos tanuláshoz szükséges idejét nem érdemes figyelembe venni. A Pima Indians Diabetes esetén az átlagos tanuláshoz szükséges időket vizsgálva megállapítható, hogy a k-NN algoritmusnak jelentősen rövidebb volt a betanítási ideje, mint a klasszikus SVM-é vagy a kvantum neurális hálózaté. A kvantum neurális hálózat pedig ezen nagyobb méretű, az Iris adatbázisnál több egyeddel és attribútummal rendelkező Pima Indians Diabetes adatbázis esetén lényegesen, több mint hússzor hosszabb időt igényelt a tanuláshoz, mint a klasszikus SVM, hiszen míg a klasszikus SVM-nek átlagosan körülbelül 104 másodpercre (kevesebb, mint 2 percre) volt szüksége a tanuláshoz, addig a kvantum neurális hálózatnak átlagosan körülbelül 2559 másodperc, azaz 42-43 perc volt szükséges ugyanehhez.

A modellek a nagyobb méretű, Pima Indians Diabetes adatbázisra történő felhasználásakor kiszámolt átlagos pontossági jellemzői, valamint tanuláshoz és osztályozáshoz szükséges idői alapján -ahogy korábban már a kisebb méretű Iris adatbázis esetén is megállapítható volt, úgy nagyobb adatbázis felhasználása mellett - még erősebben alátámaszthatóvá vált, hogy a jelen kutatás során alkalmazott kvantum neurális hálózat, amely egy hibrid megoldás (implementálásakor a klasszikus gépi tanuláshoz került hozzáadásra a kvantumosság) nem tud igazi megoldás lenni a technológiailag megjósolt kvantumfölény eléréséhez a gépi tanulás területén. Bár a pontossági jellemzők terén az Iris és a Pima Indians Diabetes adatbázisok esetén is megközelítette vagy néhol el is érte a kvantum neurális hálózat a klasszikus megoldások pontosságait, a tanuláshoz szükséges idő terén – különösen nagyobb adatbázison történő alkalmazáskor – azonban lényegesen gyengébben teljesített, mint a klasszikus algoritmusok.

A jelen kutatás során kapott eredmények nem támasztják alá a gépi tanuló kvantumalgoritmus fölényét a klasszikusokkal szemben, helyette azonban aláhúzzák a szakirodalomból ismert véleményt, miszerint a kutatásban is vizsgált hibrid megoldás nem lehet valódi megoldás a kvantumfölény eléréséhez a gépi tanulás területén. A valóban eredményes, a kvantumfölény elérésére képes megoldásokhoz olyan új algoritmusok fejlesztése szükséges, amelyek képesek a kvantumtulajdonságok biztosította előnyök kihasználása mellett hatékonyan elemezni is az adatokat.

6. Konklúzió

Jelen dolgozat arra a kérdésre kereste a választ, hogy valóban megfigyelhetők-e a Python alapú szoftver-megoldások esetén a kvantum gépi tanulás előnyei a klasszikussal szemben, azaz mi mondható el a kvantum és a hagyományos gépi tanuló algoritmusok pontosságáról és teljesítményéről. Ezen kérdés megválaszolására két különböző méretű, publikusan elérhető adatbázis felhasználásával, bináris klasszifikációs problémára került összehasonlításra egy klasszikus Support Vector Machine (SVM), k-Nearest Neighbors (k-NN) algoritmus, valamint egy automatizált gépi tanulás segítségével összeválogatott algoritmus-együttes, egy a szakirodalomban Santanu Ganguly 2021-es Quantum Machine Learning: An Applied Approach című könyvében publikált kvantum neurális hálózattal. Az összehasonlítás olyan pontossági jellemzők alapján történt, mint az accuracy, a precision, a recall, az F1-érték, valamint a ROC görbe alatti terület (AUC), ezeken kívül továbbá a tanuláshoz és az osztályozáshoz szükséges idők is megvizsgálásra kerültek.² A vizsgálatok eredményei nem igazolták a gépi tanuló kvantumalgoritmus fölényét. Sem a pontossági jellemzőkben, sem pedig a tanuláshoz vagy osztályozáshoz szükséges időkben nem múlta felül a kvantum neurális hálózat a klasszikus megoldásokat, sőt a kvantum modell betanításának ideje – különösen a nagyobb méretű adatbázis esetén - lényegesen hosszabb volt, mint a klasszikusoké.

A dolgozat eredményei alátámasztják a szakirodalomból ismert véleményt, hogy a jelen kutatás során is felhasznált nem tisztán kvantum, hanem hibrid megoldások -vagyis a kvantumosság hozzáadása a klasszikus gépi tanuláshoz - nem tud igazi megoldás lenni a technológiailag megjósolt kvantumfölény eléréséhez. A valóban célravezető és eredményes megoldásokhoz olyan új algoritmusok megalkotása szükséges, amelyek hatékonyan elemzik az adatokat, miközben kihasználják a kvantumtulajdonságok biztosította előnyöket.

A jelen kutatásban is felhasznált, a kvantumosság klasszikus gépi tanuláshoz való hozzáadásával létrejött hibrid megoldások bár a kvantumfölény elérésére nem képesek, mégis kutatásuk és alkalmazásuk fontos szerepet tölthet be a kvantum gépi tanulásban, hiszen segítségükkel bárki számára egyszerűen elérhetővé, kipróbálhatóvá, megérthetővé és fejleszthetővé válhat a technológia, ezzel további előrelépésekhez vezetve a kvantum gépi tanulás területén.

Végül az ezen dolgozat fő célját képző kvantum és klasszikus gépi tanuló algoritmusok pontosságának és teljesítményének összehasonlításával, a kvantumfölény megvalósulásával

² A kutatás során az egyes modellek betanításához, értékeléséhez és az összehasonlításhoz szükséges mérőszámok kiszámolásához írt forráskódokat igény esetén a szerző szívesen megosztja.

kapcsolatban érdemes lehet olyan egyéb kutatásokat elvégezni, mint a tisztán kvantum, azaz nem kvantum-szimulátoron, hanem valós kvantumszámítógépen, kvantumfejlesztői környezetben írt, kvantumadatokon futtatott megoldások pontosságának és teljesítményének összehasonlítása a klasszikus megoldásokkal. Illetve szintén érdekes lehet megvizsgálni azt is, hogy hogyan teljesítenek a klasszikus és a jelen kutatásban is felhasznált hibrid megoldások olyan adatkészlet esetén, amelyben az adatok valamilyen kvantummechanikai folyamat során jöttek létre (pl. kémiai, biológiai kísérletek).

Irodalomjegyzék

- Aaronson, S. (2007). The learnability of quantum states. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 463(2088), 3089–3114. <https://doi.org/10.1098/rspa.2007.0113>
- Aharonov, D., Cotler, J., & Qi, X.-L. (2022). Quantum algorithmic measurement. *Nature Communications*, 13(1), 887. <https://doi.org/10.1038/s41467-021-27922-0>
- Aïmeur, E., Brassard, G., & Gambs, S. (2006). Machine Learning in a Quantum World. In A. Y. Tawfik & S. D. Goodwin (Eds.), *Advances in Artificial Intelligence* (Vol. 3060, pp. 431–442). Springer Berlin Heidelberg. https://doi.org/10.1007/11766247_37
- Alchieri, L., Badalotti, D., Bonardi, P., & Bianco, S. (2021). An introduction to quantum machine learning: From quantum logic to quantum deep learning. *Quantum Machine Intelligence*, 3(2), 28. <https://doi.org/10.1007/s42484-021-00056-8>
- Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J. C., Barends, R., Biswas, R., Boixo, S., Brandao, F. G. S. L., Buell, D. A., Burkett, B., Chen, Y., Chen, Z., Chiaro, B., Collins, R., Courtney, W., Dunsworth, A., Farhi, E., Foxen, B., ... Martinis, J. M. (2019). Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779), 505–510. <https://doi.org/10.1038/s41586-019-1666-5>
- AutoML.org. (2023). *Auto-Sklearn*, <https://www.automl.org/automl/auto-sklearn/>
- Bala, P. C. (2020. október 16). *Introduction to Quantum Computing with Python and Qiskit, Towards AI*. <https://towardsai.net/p/programming/introduction-to-quantum-computing-with-python-and-qiskit-215f32e2e222>
- Bergholm, V., Izaac, J., Schuld, M., Gogolin, C., Ahmed, S., Ajith, V., Alam, M. S., Alonso-Linaje, G., AkashNarayanan, B., Asadi, A., Arrazola, J. M., Azad, U., Banning, S., Blank, C., Bromley, T. R., Cordier, B. A., Ceroni, J., Delgado, A., Di Matteo, O., ... Killoran, N. (2018). *PennyLane: Automatic differentiation of hybrid quantum-classical computations*. <https://doi.org/10.48550/ARXIV.1811.04968>
- Bhatia, V., & Ramkumar, K. R. (2020). An efficient Quantum Computing technique for cracking RSA using Shor's Algorithm. *2020 IEEE 5th International Conference on Computing*

- Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., & Lloyd, S. (2017). Quantum machine learning. *Nature*, 549(7671), 195–202. <https://doi.org/10.1038/nature23474>
- Boixo, S., Isakov, S. V., Smelyanskiy, V. N., Babbush, R., Ding, N., Jiang, Z., Bremner, M. J., Martinis, J. M., & Neven, H. (2018). Characterizing quantum supremacy in near-term devices. *Nature Physics*, 14(6), 595–600. <https://doi.org/10.1038/s41567-018-0124-x>
- Cerezo, M., Verdon, G., Huang, H.-Y., Cincio, L., & Coles, P. J. (2022). Challenges and opportunities in quantum machine learning. *Nature Computational Science*, 2(9), 567–576. <https://doi.org/10.1038/s43588-022-00311-3>
- Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, 408, 189–215. <https://doi.org/10.1016/j.neucom.2019.10.118>
- Chang, V., Bailey, J., Xu, Q. A., & Sun, Z. (2022). Pima Indians diabetes mellitus classification based on machine learning (ML) algorithms. *Neural computing & applications*, 1–17. Advance online publication. <https://doi.org/10.1007/s00521-022-07049-z>
- Chen, S., Cotler, J., Huang, H.-Y., & Li, J. (2021). *A Hierarchy for Replica Quantum Advantage*. <https://doi.org/10.48550/ARXIV.2111.05874>
- Chen, S., Cotler, J., Huang, H.-Y., & Li, J. (2022). Exponential Separations Between Learning With and Without Quantum Memory. *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, 574–585. <https://doi.org/10.1109/FOCS52979.2021.00063>
- Cotler, J., Huang, H.-Y., & McClean, J. R. (2021). *Revisiting dequantization and quantum advantage in learning tasks*. <https://doi.org/10.48550/ARXIV.2112.00811>
- Cross, A. W., Smith, G., & Smolin, J. A. (2015). Quantum learning robust against noise. *Physical Review A*, 92(1), 012327. <https://doi.org/10.1103/PhysRevA.92.012327>
- Czakon, J. (2023. március 21). *24 Evaluation Metrics for Binary Classification (And When to Use Them)*, neptune.ai. <https://neptune.ai/blog/evaluation-metrics-binary-classification>

Dilmegani, C. (2022a, január 10). *State of Quantum Computing in 2022 EOY for Business Leaders*, AI Multiple. <https://research.aimultiple.com/quantum-computing/>

Dilmegani, C. (2022b, február 11). *Top 20 Quantum Computing Use Cases in 2022*, AI Multiple. <https://research.aimultiple.com/quantum-computing-applications/>

Feurer, M., Klein, A., Eggenberger, K., Springenberg, J., Blum, M., & Hutter, F. (2015). Efficient and robust automated machine learning. *Advances in neural information processing systems*, 28.

Fodor, Sz. E. (2022a). *Felügyelt gépi tanítás – Klasszifikáció II*. A mesterséges intelligencia alapjai tárgy 5. gyakorlatának diasora. Budapest: Budapesti Corvinus Egyetem

Fodor, Sz. E. (2022b). *Automatizált gépi tanulás (autoML)*. A mesterséges intelligencia alapjai tárgy 11. gyakorlatának diasora. Budapest: Budapesti Corvinus Egyetem

Ganguly, S. (2021). *Quantum Machine Learning: An Applied Approach*. Ashford: Apress Berkeley CA.

Ghahramani, Z. (2004). Unsupervised Learning. In O. Bousquet, U. von Luxburg, & G. Rätsch (Eds.), *Advanced Lectures on Machine Learning* (Vol. 3176, pp. 72–112). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-28650-9_5

Google. (2023). *Üdvözljük a Colab webhelyén!* <https://colab.research.google.com/>

Handelman, G. S., Kok, H. K., Chandra, R. V., Razavi, A. H., Huang, S., Brooks, M., Lee, M. J., & Asadi, H. (2019). Peering Into the Black Box of Artificial Intelligence: Evaluation Metrics of Machine Learning Methods. *American Journal of Roentgenology*, 212(1), 38–43. <https://doi.org/10.2214/AJR.18.20224>

Horodecki, R., Horodecki, P., Horodecki, M., & Horodecki, K. (2009). Quantum entanglement. *Reviews of Modern Physics*, 81(2), 865–942. <https://doi.org/10.1103/RevModPhys.81.865>

Huang, H.-Y., Kueng, R., & Preskill, J. (2021). Information-Theoretic Bounds on Quantum Advantage in Machine Learning. *Physical Review Letters*, 126(19), 190505. <https://doi.org/10.1103/PhysRevLett.126.190505>

Huang, H.-Y., Broughton, M., Cotler, J., Chen, S., Li, J., Mohseni, M., Neven, H., Babbush, R., Kueng, R., Preskill, J., & McClean, J. R. (2022). Quantum advantage in learning from experiments. *Science*, 376(6598), 1182–1186. <https://doi.org/10.1126/science.abn7293>

- Kaggle. (2023). *Pima Indians Diabetes Database*, <https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database>
- Khan, T. M., & Robles-Kelly, A. (2020). Machine Learning: Quantum vs Classical. *IEEE Access*, 8, 219275-219294. <https://doi.org/10.1109/ACCESS.2020.3041719>
- Knill, E. (2010). Quantum computing. *Nature*, 463(7280), 441–443. <https://doi.org/10.1038/463441a>
- Ménard, A., Ostojic, I., Patel, M., & Volz, D. (2020, February 6). A game plan for quantum computing. *McKinsey Quarterly*. <https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/a-game-plan-for-quantum-computing>
- Microsoft. (2023a). *Mi az a qubit?* <https://azure.microsoft.com/hu-hu/overview/what-is-a-qubit/#introduction>
- Microsoft. (2023b). *Mi a gépi tanulás?* <https://azure.microsoft.com/hu-hu/overview/what-is-machine-learning-platform/#benefits>
- Microsoft. (2022. január 4). *Understanding quantum computing*, <https://docs.microsoft.com/en-us/azure/quantum/overview-understanding-quantum-computing>
- Mishra, N., Kapil, M., Rakesh, H., Anand, A., Mishra, N., Warke, A., Sarkar, S., Dutta, S., Gupta, S., Dash, A. P., Gharat, R., Chatterjee, Y., Roy, S., Raj, S., Jain, V. K., Bagaria, S., Chaudhary, S., Singh, V., Maji, R., Dalei, P., Behera, B. K., Mukhopadhyay, S., & Panigrahi, P. K. (2021). Quantum Machine Learning: A Review and Current Status. In N. Sharma, A. Chakrabarti, V. E. Balas, & J. Martinovic (Eds.), *Data Management, Analytics and Innovation* (Vol. 1175, pp. 101–145). Springer Singapore. https://doi.org/10.1007/978-981-15-5619-7_8
- Monroe, C., Meekhof, D. M., King, B. E., Itano, W. M., & Wineland, D. J. (1995). Demonstration of a Fundamental Quantum Logic Gate. *Physical Review Letters*, 75(25), 4714–4717. <https://doi.org/10.1103/PhysRevLett.75.4714>
- Monroe, C., Meekhof, D. M., King, B. E., & Wineland, D. J. (1996). A “Schrödinger Cat” Superposition State of an Atom. *Science*, 272(5265), 1131–1136. <https://doi.org/10.1126/science.272.5265.1131>
- Mosca, M. (2018). Cybersecurity in an era with Quantum Computers: Will we be ready?. *IEEE Security & Privacy*, 16(5), 38-41. <https://doi.org/10.1109/MSP.2018.3761723>

- Mucherino, A., Papajorgji, P. J., & Pardalos, P. M. (2009). K-Nearest Neighbor Classification. In A. Mucherino, P. J. Papajorgji, & P. M. Pardalos, *Data Mining in Agriculture* (Köt. 34, o. 83–106). Springer New York. https://doi.org/10.1007/978-0-387-88615-2_4
- Partida, D. (2022. janúar 24). *Top Quantum Computing Companies for 2022*, CIOinsight. <https://www.cioinsight.com/innovation/quantum-computing-companies/>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2012). *Scikit-learn: Machine Learning in Python*. <https://doi.org/10.48550/ARXIV.1201.0490>
- PennyLane. (2023a). *What is PennyLane?* <https://docs.pennylane.ai/en/stable/introduction/pennylane.html>
- PennyLane. (2023b). *Frequently Asked Questions*, <https://pennylane.ai/faq.html>
- Python. (2023a). *General Python FAQ*, <https://docs.python.org/3/faq/general.html>
- Python. (2023b). *About*, <https://www.python.org/about/>
- Python. (2023c). *Applications for Python*, <https://www.python.org/about/apps/>
- Roell, J. (2018. február 26). *Demystifying Quantum Gates – One Qubit At A Time*, Towards Data Science. <https://towardsdatascience.com/demystifying-quantum-gates-one-qubit-at-a-time-54404ed80640>
- Salemian, S., & Mohammadne, S. (2008). Quantum Hadamard Gate Implementation Using Planar Lightwave Circuit and Photonic Crystal Structures. *American Journal of Applied Sciences*, 5(9), 1144–1148. <https://doi.org/10.3844/ajassp.2008.1144.1148>
- Schuld, M., & Petruccione, F. (2018). *Supervised learning with quantum computers* (Vol. 17). Berlin: Springer. <https://doi.org/10.1007/978-3-319-96424-9>
- Schuld, M., Sinayskiy, I., & Petruccione, F. (2015). An introduction to quantum machine learning. *Contemporary Physics*, 56(2), 172-185. <https://doi.org/10.1080/00107514.2014.964942>

Servedio, R. A., & Gortler, S. J. (2004). Equivalences and Separations Between Quantum and Classical Learnability. *SIAM Journal on Computing*, 33(5), 1067–1092.

<https://doi.org/10.1137/S0097539704412910>

Srivastava, R., Choi, I., Cook, T., & Team, N. U. E. (2016). The commercial prospects for quantum computing. *Networked Quantum Information Technologies*.

<https://nqit.ox.ac.uk/sites/www.nqit.ox.ac.uk/files/2018-10/Commercial%20Prospects%20for%20Quantum%20Computing%20Dec%202016.pdf>

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

UC Irvine Machine Learning Repository. (2023). *Iris Data Set*,

<https://archive.ics.uci.edu/ml/datasets/iris>

Vas, R. F. (2020a). *Gépi tanulás I.* Üzleti intelligencia tárgy 8. előadás. Budapest: Budapesti Corvinus Egyetem

Vas, R. F. (2020b). *Gépi tanulás II.* Üzleti intelligencia tárgy 9. előadás. Budapest: Budapesti Corvinus Egyetem

Williams, C. P. (2011). *Explorations in Quantum Computing* (2nd ed). Springer London.

<https://doi.org/10.1007/978-1-84628-887-6>

Witteck, P. (2014). *Quantum machine learning: What quantum computing means to data mining* (1st. edition). Elsevier AP, Academic Press.