

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KUANTUM ÇEKİRDEK TEMELLİ SINIFLANDIRMA

Gönül SABAĞ

YÜKSEK LİSANS TEZİ
Matematik Mühendisliği Anabilim Dalı
Matematik Mühendisliği Programı

Danışman
Dr. Öğr. Üyesi Nilgün GÜLER BAYAZIT

Temmuz, 2022

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KUANTUM ÇEKİRDEK TEMELLİ SINIFLANDIRMA

Gönül SABAH tarafından hazırlanan tez çalışması 21.07.2022 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Matematik Mühendisliği Anabilim Dalı Matematik Mühendisliği Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Dr. Öğr. Üyesi Nilgün GÜLER
BAYAZIT
Yıldız Teknik Üniversitesi
Danışman

Jüri Üyeleri

Dr. Öğr. Üyesi Nilgün GÜLER BAYAZIT, Danışman
Yıldız Teknik Üniversitesi

Doç. Dr. Salih ÇELİK, Üye
Yıldız Teknik Üniversitesi

Prof. Dr. Olcay Taner YILDIZ, Üye
Özyeğin Üniversitesi

Danışmanım Dr. Öğr. Üyesi Nilgün GÜLER BAYAZIT sorumluluğunda tarafımda hazırlanan Kuantum Çekirdek Temelli Sınıflandırma başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Gönül SABAH

İmza

Sevdiklerime...



TEŐEKKÜR

Bu tez alıőması sűreci boyunca, yeni konular hakkında bilgi sahibi olmamı saęlayan ve bakıő aımı geliőtiren, akademik ve sosyal olarak desteęini hissettięim deęerli danıőman hocam Dr. Öğr. Üyesi Nilgűn GŪLER BAYAZIT'a sonsuz teőekkűr ederim.

Gűnűl SABAHA



İÇİNDEKİLER

SİMGE LİSTESİ	vii
KISALTMA LİSTESİ	viii
ŞEKİL LİSTESİ	ix
TABLO LİSTESİ	x
ÖZET	xi
ABSTRACT	xii
1 GİRİŞ	1
1.1 Literatür Özeti	1
1.2 Amaç	2
1.3 Hipotez	2
2 YÖNTEMLER	3
2.1 Temel Kavramlar	3
2.2 Destek Vektör Makinaları	4
2.2.1 Çekirdek	4
2.2.2 Çekirdek Fonksiyonları	4
2.3 Kuantum Hesaplama	6
2.3.1 Kubit	6
2.3.2 Bloch Küresi	7
2.3.3 Süperpozisyon	8
2.3.4 Dolanıklık	9
2.3.5 Durum Evrilmesi	9
2.3.6 Gözlemlenebilirler ve Ölçümler	10
2.3.7 Yoğunluk Matrisi ve Karışık Durumlar	11
2.3.8 Kuantum Mantık Kapıları	12
2.4 Kuantum Makine Öğrenmesi	18
2.4.1 Kuantum Veri Kodlama ve Kuantum Çekirdek	20
2.4.2 Kuantum Modelin Ölçümü	27

2.5	Kuantum Çekirdek Temelli Destek Vektör Makinesi	28
3	ARAŞTIRMA YÖNTEMİ	30
3.1	Kuantum Çekirdek Tabanlı Destek Vektör Makinesi ile Parkinson Hastalığı Teşhisi	30
3.1.1	Veri Kümesi	30
3.1.2	Simülatörler ve Kuantum Bilgisayarlar	32
3.1.3	Ön İşleme Aşaması	32
3.1.4	Kuantum Çekirdek Temelli Destek Vektör Makinesi Modeli . . .	33
4	DENEYSEL ÇALIŞMALAR	35
5	SONUÇ VE ÖNERİLER	38
	KAYNAKÇA	39
	TEZDEN ÜRETİLMİŞ YAYINLAR	41

SİMGE LİSTESİ

J_ν	Bessel Deęeri
\mathcal{O}	Büyük O
\prod	Çarpım Sembolü
\dagger	Dagger Sembolü
$ \cdot\rangle$	Dirac Gösterimi
\in	Elemanındır İşareti
e	Euler Sayısı
γ	Gauss Çekirdek Katsayısı
$\langle\cdot \cdot\rangle$	Hilbert Uzayında İç Çarpım
$\langle\cdot,\cdot\rangle$	İç Çarpımı
\mathbb{C}	Kompleks Sayılar Kümesi
T	Matrisin Transpozesi
$ \cdot $	Mutlak Deęer
\mathbb{R}	Reel Sayılar Kümesi
\otimes	Tensor Çarpımı
\sum	Toplam Sembolü
$\ \cdot\ $	Vektör Boyu Gösterimi

KISALTMA LİSTESİ

CPU	Merkezi İşleme Birimi
DVM	Destek Vektör Makinaları
KDVM	Kuantum Destek Vektör Makinesi
Poly	Polynomial Çekirdek
RBF	Radial Basis Function Çekirdek
tr	Matrisin izin
QKML	Quantum Kernel Machine Learning
QPU	Kuantum İşleme Birimi

ŞEKİL LİSTESİ

Şekil 2.1	Bloch Küresi	8
Şekil 2.2	Kuantum Makine Öğrenmesi	19
Şekil 2.3	Klasik/Kuantum makine öğreniminde kullanılan dört yaklaşım [16]	19
Şekil 2.4	$x \rightarrow \phi(x)$ dönüşümü ([3]'den adapte edilmiştir.)	20
Şekil 2.5	U üniter işlemci	20
Şekil 2.6	Kuantum Devre ([4]'den adapte edilmiştir)	29
Şekil 3.1	Parkinson Veri Kümesi Korelasyon matrisi	32
Şekil 3.2	Kuantum devre ve klasik DVM	33
Şekil 4.1	Karışıklık Matrisleri	36

TABLO LİSTESİ

Tablo 2.1	Pauli Kapıları	13
Tablo 2.2	Faz Kaydırma Kapı	15
Tablo 2.3	Rotasyon Kapıları	16
Tablo 3.1	Parkinson Hastalığı Veri Kümesinin Öznitelikleri	31
Tablo 4.1	Model Performans Sonuçları	35
Tablo 4.2	Klasik DVM ile KDVM Sonuçlarının Karşılaştırılması	37

Kuantum Çekirdek Temelli Sınıflandırma

Gönül SABAH

Matematik Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Danışman: Dr. Öğr. Üyesi Nilgün GÜLER BAYAZIT

Kuantum bilgisayarların gelişimiyle, kuantum makine öğrenimi yöntemlerinin araştırılması ve geliştirilmesi daha da önem kazanmıştır. Bu tez çalışmasında klasik makine öğrenimi algoritmalarından biri olan çekirdek fonksiyonları, kuantum bilgisayarlarda uygulanması, algoritmanın yeni bir makine öğrenmesi yöntemi sunabilirliği ve sonuçları üzerinde durulmaktadır. Kauntum çekirdek temelli destek vektör makinesi ile bir model hazırlanmış ve Parkinson Hastalığı veri seti üzerinde eğitim ve testler yapılmıştır. Kuantum çekirdek için 6 kubit ve daha önceki çalışmalarda kullanılan öznitellik setleri kadar kubit ile kuantum devreler kurulup, eğitimler gerçekleştirilmiştir. Sonuçlar klasik çekirdekli destek vektör makinelere sonuçları ile karşılaştırılmıştır. Karşılaştırma sonucunda, kuantum hesaplamaların üstün özelliklerinden dolayı çekirdek tahminlerinde daha iyi performans gösterdiği görülmüştür.

Anahtar Kelimeler: Kuantum Çekirdek, Kuantum Makine Öğrenmesi, Destek Vektör Makinaları

Quantum Kernel-Based Classification

Gönül SABAH

Department of Mathematical Engineering
Master of Science Thesis

Supervisor: Assist. Prof. Dr. Nilgün GÜLER BAYAZIT

Research and development of quantum machine learning methods have become even more important with the development of quantum computers. In this thesis, kernel functions, one of the classical machine learning algorithms, its model in quantum computers, the ability of the algorithm to offer a new machine learning method and its results are emphasized.

An application was prepared with the Quantum kernel-based support vector machine and training and tests were carried out on the Parkinson's Disease dataset. Quantum circuit has been established for the quantum kernel with 6 qubits and as many qubits as the feature sets used qubits in previous studies and then trainings have been carried out. The results are compared with those of classical kernel support vector machines. As a result of the comparison, it has been seen that quantum computing performs better in kernel predictions due to its superior properties.

Keywords: Quantum Kernel, Quantum Machine Learning, Support Vector Machine

1.1 Literatür Özeti

Günümüzde artan verinin analiz edilmesi ve yorumlanması ihtiyaç haline gelmiş ve daha da önem kazanmıştır. Klasik bilgisayarlar ile artan bu verinin makine öğrenmesi yöntemleri ile işlenmesi giderek çok uzun zaman almaktadır. Bu sebeple klasik bilgisayarlara çözüm olarak, kuantum mekaniğine dayalı bilgisayarlar olacağı düşünülmektedir [1].

Makine öğrenmesi ve kuantum hesaplama, klasik bilgisayarlar ile çözülemeyen problemlerin çözümünde oldukça yüksek potansiyele sahiptir [2]. Makine öğrenme yöntemlerinden biri olan Destek Vektör Makinaları (DVM) çekirdek fonksiyonları ile örüntü tanımada oldukça yaygın kullanılmaktadır. Maria Schuld ve Nathan Killoran [3] çalışmalarında, sınıflandırma yaparken kullanılan klasik çekirdek ile kuantum çekirdek hesaplamalarının birbirine oldukça benzer olduğunu ve kuantum çekirdeğin Hilbert uzayında kuantum durumların çarpımından elde edilebileceğini göstermiştir. M. Schuld, kuantum çekirdek modellerini, kuantum ölçümü doğrusal karar sınırı olan, girdileri Hilbert öznelik uzayına eşleyen bir model olarak tanımlamıştır [4].

Vojtěch Havlíček vd. süper iletken işlemci üzerine uygulanan kuantum çekirdeklerin, kuantum öznelik setlerini sınıflandırırken tahminlerinin oldukça başarılı olduğunu göstermiştir [2].

Kuantum sınıflandırıcıların, makine öğrenimi modelleri olarak eğitilebilirliği araştırılmaktadır. Seth Lloyd vd. farklı metrikler kullanarak sınıflandırma problemlerinde deneyler yapmış, kuantum modellerin eğitilebilirliği üzerine araştırmalar yapmıştır [5].

Klasik makine öğrenimi yöntemlerinin performansını belirleyen hiperparametrelerden biri de bant genişliğidir. Ruslan Shaydulin ve Stefan M. Wild, klasik çekirdek ile kuantum çekirdek arasındaki bant genişliği benzerliğini göstererek, bant genişliğini kubit sayısı ile optimize edebileceklerini göstermiştir [6].

1.2 Amaç

Günümüzde artan verinin analiz edilmesi ve yorumlanması ihtiyaç haline gelmiş ve daha da önem kazanmıştır. Klasik bilgisayarlar ile artan bu verinin makine öğrenmesi yöntemleri ile işlenmesi giderek çok uzun zaman almaktadır. Bu sebeple klasik bilgisayarlara alternatif olarak kuantum mekaniğine dayalı bilgisayarlar düşünülmektedir [1]. Gelişen kuantum bilgisayarlar sayesinde kuantum makine öğrenmesi yöntemleri daha da önem kazanmıştır. Bu amaçla bilinen klasik yöntemler kuantum makinelerde kullanılmak üzere geliştirilmektedir [7].

Kuantum makine öğrenme yöntemlerinden biri olan çekirdek metodlarının kuantum öznitelik setleri üzerinde sınıflandırma tahminlerinin oldukça başarılı olduğu bilinmektedir [2]. Bu tezin amacı kuantum tabanlı çekirdek çeşitlerinin incelenmesidir. Bu amaçla Parkinson veri kümesi üzerinde çeşitli çekirdek modeller eğitilerek performans karşılaştırması yapılmıştır.

1.3 Hipotez

Kuantum tabanlı destek vektör makine modellerinin klasik destek vektör makine yöntemlerine göre, kuantum hesaplama avantajlarından dolayı daha başarılı performans vereceğini düşünmekteyiz.

Parkinson veri kümesi üzerinde kuantum tabanlı çekirdek destek vektör makinesi ile klasik destek vektör makinesi yöntemleri kullanılarak sınıflandırma modelleri eğitilerek, bu veriye en uygun olan çekirdek çeşidi araştırılacaktır.

2.1 Temel Kavramlar

Tezde kullanılan fonksiyonel analizden birkaç terimin tanımı bu bölümde verilecektir [8, 9].

Tanım 2.1. (İç çarpım). V kümesi \mathbb{C} kompleks sayı cismi üzerinde bir lineer vektör uzayı olsun. $\langle \cdot, \cdot \rangle : V \times V \rightarrow \mathbb{C}$, $u, v \in V$ için

(i) $u \neq 0$ olduğunda $\langle u, u \rangle$ ve
 $u = 0 \Leftrightarrow \langle u, u \rangle = 0$

(ii) $\langle u, v \rangle = \overline{\langle v, u \rangle}$

(iii) $\forall \alpha, \beta \in \mathbb{C}$ ve $u_1, u_2, u_3 \in V$ için $\langle \alpha u_1 + \beta u_2, u_3 \rangle = \alpha \langle u_1, u_3 \rangle + \beta \langle u_2, u_3 \rangle$

koşulları gerçekleşiyorsa $\langle \cdot, \cdot \rangle$ fonksiyonuna iç çarpım denir.

Eğer kompleks cisim \mathbb{C} yerine \mathbb{R} reel sayılar cismi alınmış ise (ii) özelliği yerine $\langle u, v \rangle = \langle v, u \rangle$ simetri koşulu aranır.

Tanım 2.2. (İç çarpım uzayı). Eğer bir V vektör uzayı üzerinde iç çarpım tanımlanabiliyorsa bu vektör uzayına İç çarpım uzayı denir .

Tanım 2.3. (Hilbert Uzayı). Tam iç çarpım uzayına Hilbert uzayı denir.

Tanım 2.4. (Çekirdek). $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^m$ öznitelik haritası verildiğinde $x, x' \in \mathbb{R}^n$ vektörlerinin iç çarpımlarına karşılık gelen κ çekirdeği:

$$\kappa(x, x') = \phi(x)^T \phi(x') \quad (2.1)$$

şeklinde tanımlanır. Bir fonksiyonun çekirdek olabilmesi için simetrik, pozitif yarı-tanımlı bir fonksiyon olması ve Cauchy-Schwarz eşitsizliğine uyması gerek ve yeter koşulları gerekmektedir. Çekirdekler, uzayın ne olduğunu bilmeden bazı öznitelik uzaylarında çarpım hesaplamasının kolay bir yolunu verir.

Tanım 2.5. (Gram Matrisi). $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{K}$ ($\mathbb{K} = \mathbb{C}$ veya $\mathbb{K} = \mathbb{R}$) fonksiyonu verilmiş olsun. $x^m \in \mathcal{X}, m = 1, \dots, M$ elemanlarına karşılık gelen $M \times M$ boyutlu $\kappa = \kappa(x^m, x^{m'})$ matrisi gram matrisidir.

2.2 Destek Vektör Makinaları

Destek vektör makinaları (DVM), hem sınıflandırma hem de regresyon için kullanılan bir denetimli makine öğrenme algoritmasıdır. Bu algoritmanın amacı, iki sınıflı olarak etiketlenmiş veri kümesini, sınıflara ayırabilecek optimal hiperdüzlemi bulmaktır. Eğer veri kümesi doğrusal olarak ayrılabilir değil ise verinin boyutu, veri doğrusal olarak ayrılabilir oluncaya kadar büyütülür [10].

2.2.1 Çekirdek

Çekirdeğin amacı, veri kümesini daha yüksek bir uzay boyutuna taşıyarak iki girdi vektörünün arasındaki benzerliği tespit etmektir. Böylece bir düzlem ile ayıramayan veri kümesi, birbirine benzeyen gruplar halinde ayrılabilir.

2.2.2 Çekirdek Fonksiyonları

DVM çekirdek olarak, birbirine bağlı matematiksel işlemleri kullanır. Çekirdek, verileri girdi olarak alır ve onları gerekli forma dönüştürerek bir sıra işlem yapar. DVM, verinin türüne, boyutuna göre farklı türde çekirdek fonksiyonları kullanır. En çok kullanılan çekirdek fonksiyonları: doğrusal, polinom, Laplace, Gauss, Sigmoid, Anova, Bessel çekirdek fonksiyonlarıdır.

Tanım 2.6. (Doğrusal Çekirdek).

Doğrusal çekirdek, iç çarpımı ile ifade edilen en basit çekirdek fonksiyonudur. Çekirdek fonksiyonu, $x, x' \in \mathbb{R}^n$ de birer vektör olmak üzere:

$$\kappa(x, x') = x^T x' \quad (2.2)$$

şeklinde tanımlanır.

Tanım 2.7. (Polinom Çekirdek).

Polinom çekirdek, durağan olmayan çekirdek fonksiyonudur. Çekirdek fonksiyonu, $x, x' \in \mathbb{R}^n$ de birer vektör olmak üzere:

$$\kappa(x, x') = (1 + x^T x')^d \quad (2.3)$$

şeklinde tanımlanır.

Tanım 2.8. (Gauss Çekirdek).

Gauss çekirdek, radyal tabanlı çekirdek fonksiyonudur. Çekirdek fonksiyonu, $x, x' \in \mathbb{R}^n$ de birer vektör olmak üzere:

$$\kappa(x, x') = e^{-\gamma \|x-x'\|^2}, \gamma > 0 \quad (2.4)$$

şeklinde tanımlanır.

Tanım 2.9. (Laplace Çekirdek).

Laplace çekirdek, radyal tabanlı üstel çekirdek fonksiyonudur. Çekirdek fonksiyonu, $x, x' \in \mathbb{R}^n$ de birer vektör olmak üzere:

$$\kappa(x, x') = e^{-\frac{\|x-x'\|}{\sigma}} \quad (2.5)$$

şeklinde tanımlanır.

Tanım 2.10. (Anova Çekirdek).

Anova, radyal tabanlı fonksiyon çekirdek fonksiyonudur. Çekirdek fonksiyonu, $x, x' \in \mathbb{R}^n$ de birer vektör olmak üzere:

$$\kappa(x, x') = \sum_{i=1}^n e^{-\sigma(x^i - x'^i)^2} \quad (2.6)$$

şeklinde tanımlanır.

Tanım 2.11. (Sigmoid Çekirdek).

Sigmoid çekirdek fonksiyonunda $\alpha \in \mathbb{R}$ eğim ve $c \in \mathbb{R}$ kesme sabiti olmak üzere iki ayarlanabilir parametresi vardır. α değeri için $(1/N)$ kullanılabilir ve N değeri veri kümesinin boyutudur. Çekirdek fonksiyonu, $x, x' \in \mathbb{R}^n$ de birer vektör olmak üzere:

$$\kappa(x, x') = \tanh(\alpha x^T x' + c) \quad (2.7)$$

şeklinde tanımlanır.

Tanım 2.12. (Bessel Çekirdek).

$J_\nu(x)$ birinci türden ν . mertebeden Bessel fonksiyonu olmak üzere, çekirdek fonksiyonu:

$$\kappa(x, x') = \frac{J_{\nu+1}(\|x - x'\|)}{\|x - x'\|^{-n(\nu+1)}} \quad (2.8)$$

Bessel çekirdek olarak adlandırılır.

2.3 Kuantum Hesaplama

Kuantum bilgisayarda hesaplamalar yapabilmek için kuantum hesaplamasının temelleri verilmiştir [11, 12].

2.3.1 Kubit

Kuantum, atom altı parçacıkların durumlarının gözlemlenmesidir. Kuantum fiziğinde, atom altı parçacıkların kuantum durumları gözlemlenene kadar hem 0 hem de 1 olarak kabul edilir. Bu temele dayanarak kuantum bilgisayarlarda en küçük birim *kuantum bit(kubit)* kavramı kullanılır ve bir kubit dönüş(spin) yönüne bağlı olarak 0 ile 1 arasında sonsuz değer alabilir. Kubitin kuantum durumu, baz durumları $|0\rangle$ ve $|1\rangle$ ile gösterilir. Bu notasyona *ket notasyonu* denir ve Paul Dirac tarafından bulunmuştur. Bu yüzden *Dirac notasyonu* da denilmektedir. Bir kubitin baz durumları 2 boyutlu Hilbert uzayında taban vektörleri ile:

$$\begin{aligned} |0\rangle &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ |1\rangle &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{aligned} \quad (2.9)$$

şeklinde birim vektör olarak ifade edilebilir. Bu bazlar ortonormal sistemler oluşturmaktadır. Yani,

$$\begin{aligned} \langle 0|0\rangle &= \langle 1|1\rangle = 1 \\ \langle 0|1\rangle &= \langle 1|0\rangle = 0 \end{aligned} \quad (2.10)$$

şeklinde bir kubitin 0-1 durumları ile gösterilir. Kuantum durumlar ϕ, ψ, θ gibi küçük Yunan harfleriyle gösterilir. Bir kubitin *genel durumu* baz durumlarının lineer kombinasyonu olarak,

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (2.11)$$

ile ifade edilir, $\alpha, \beta \in \mathbb{C}$ için $|\alpha|^2 + |\beta|^2 = 1$ koşulu sağlanmalıdır. α, β kat sayılarına *olasılık genliği* denir. Buradan kubitin durumu iki boyutlu kompleks vektör olarak

$$|\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (2.12)$$

ile ifade edilebilir.

Kubit hakkında bilgi edinmek için *ölçüm* işlemi yapılmalıdır. $|\psi\rangle$ için $|0\rangle$ ve $|1\rangle$ baz vektörlerin, sırasıyla $\alpha\alpha^* = |\alpha|^2$, $\beta\beta^* = |\beta|^2$ ölçüm olasılıklarıdır.

Birden fazla kubit durumu tensor çarpımı ile ifade edilir. İki kubitlik bir kuantum sistemde,

$$\begin{aligned} |\psi\rangle_1 &= \alpha_1|0\rangle + \beta_1|1\rangle = 1, \quad \alpha_1, \beta_1 \in \mathbb{C} \quad |\alpha_1|^2 + |\beta_1|^2 = 1 \\ |\psi\rangle_2 &= \alpha_2|0\rangle + \beta_2|1\rangle = 1, \quad \alpha_2, \beta_2 \in \mathbb{C} \quad |\alpha_2|^2 + |\beta_2|^2 = 1 \end{aligned} \quad (2.13)$$

$|\psi\rangle_1$ ve $|\psi\rangle_2$ tensor çarpımı:

$$\begin{aligned} |\psi\rangle^{\otimes 2} &= |\psi\rangle_1 \otimes |\psi\rangle_2 \\ &= (\alpha_1|0\rangle + \beta_1|1\rangle) \otimes (\alpha_2|0\rangle + \beta_2|1\rangle) \\ &= \alpha_1|0\rangle \otimes \alpha_2|0\rangle + \alpha_1|0\rangle \otimes \beta_2|1\rangle + \beta_1|1\rangle \otimes \alpha_2|0\rangle + \beta_1|1\rangle \otimes \beta_2|1\rangle \\ &= \alpha_1\alpha_2|00\rangle + \alpha_1\beta_2|01\rangle + \beta_1\alpha_2|10\rangle + \beta_1\beta_2|11\rangle \end{aligned} \quad (2.14)$$

şeklindedir. Burada $|\alpha_1\alpha_2|^2 + |\alpha_1\beta_2|^2 + |\beta_1\alpha_2|^2 + |\beta_1\beta_2|^2 = 1$ dir.

n -kubitlik kuantum durum:

$$|\psi\rangle^{\otimes n} = |\psi\rangle_1 \otimes \dots \otimes |\psi\rangle_{n-1} \otimes |\psi\rangle_n = \alpha_1|00\dots 0\rangle + \beta_1|00\dots 1\rangle + \dots + \alpha_n|1\dots 1\rangle + \beta_n|1\dots 1\rangle \quad (2.15)$$

ifade edilir. Olasılık genlikleri $|\alpha_1|^2 + |\alpha_2|^2 + |\beta_1|^2 + |\beta_2|^2 + \dots + |\alpha_n|^2 + |\beta_n|^2 = 1$ dir.

2.3.2 Bloch Küresi

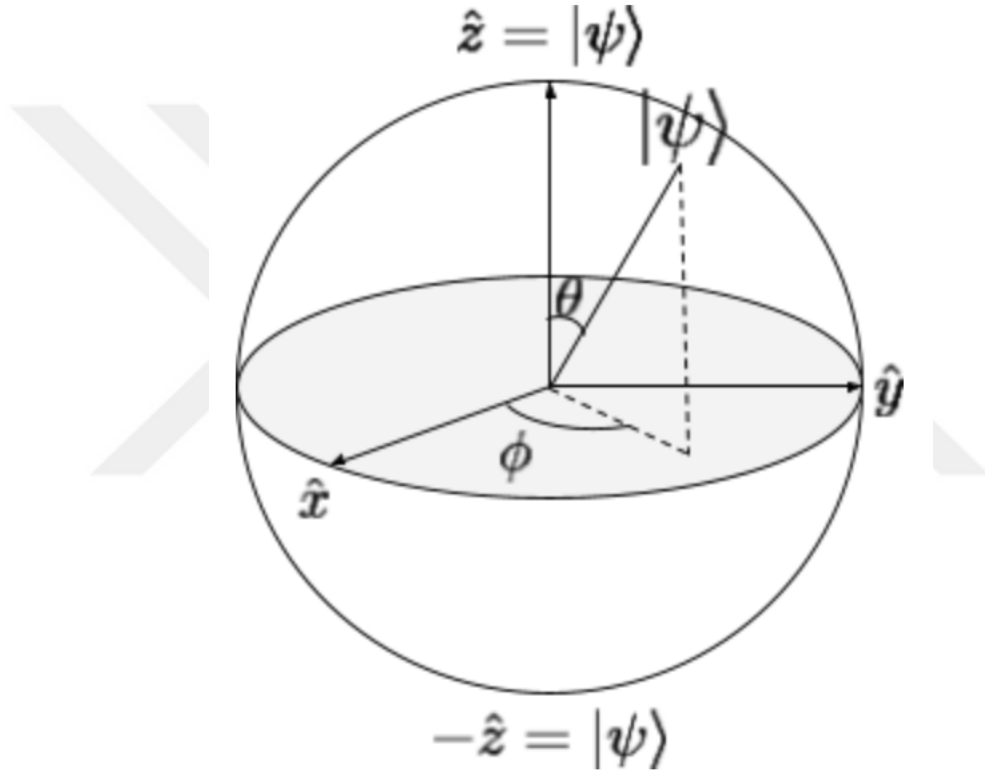
Bir kubitin geometrik olarak polar koordinatlarıyla Bloch küresi üzerinde gösterimidir.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad \alpha, \beta \in \mathbb{C} \quad |\alpha|^2 + |\beta|^2 = 1 \quad (2.16)$$

Kuantum durum için, α ve β 'yı θ ve ϕ açıları cinsinden,

$$\begin{aligned}
 0 &\leq \theta \leq \pi, \quad 0 \leq \phi < 2\pi \\
 \alpha &= \cos\left(\frac{\theta}{2}\right) \\
 \beta &= e^{i\phi} \sin\left(\frac{\theta}{2}\right), \quad i^2 = -1, \quad i \in \mathbb{C} \\
 |\psi\rangle &= \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi} \sin\left(\frac{\theta}{2}\right)|1\rangle
 \end{aligned}
 \tag{2.17}$$

şeklinde Bloch küresi üzerinde gösterilir.



Şekil 2.1 Bloch Küresi

2.3.3 Süperpozisyon

Atomaltı parçacıkların kuantum sistem gözlemlenmediği sürece, kendilerine öz kuantum mekaniksel biçimde davranarak aynı anda birçok yerde bulunma özelliği göstermesine *süperpozisyon* denir. Yani klasik bilgisayarlarda 0 ve 1 bit olabiliyorken, kuantum bilgisayarlar kubitin 0, 1 ve bunların tüm olası durumlarını hesaplar. Örneğin, elektronların hareketi olan spin, hem yukarı hem de aşağı yönelimlerin hesaplanmasıdır. Bu şekilde aynı anda tüm olasılıkların hesaplanması, birçok olası sonucu değerlendirme olanağı sağlamaktadır. n-kubit kuantum sistemin süperpozisyon durumu, 2^n farklı şekilde ifade edilebilir.

2.3.4 Dolanıklık

Kuantum bilgisayarların özelliklerinden biri de dolanıklık kavramıdır. Çoklu kubit sistemlerde kendinden daha küçük boyutta, tensör çarpımı ile ifade edilemeyen durumlara *dolanık durum* denir. Örneğin, $|\psi\rangle = \frac{|00\rangle+|11\rangle}{\sqrt{3}}$ durumu dolanık olsun.

$$\begin{aligned} \frac{1}{\sqrt{3}}(|00\rangle + |11\rangle) &= (\alpha_0|0\rangle + \beta_0|1\rangle) \otimes (\alpha_1|0\rangle + \beta_1|1\rangle) \\ &= \alpha_0\alpha_1|00\rangle + \alpha_0\beta_1|01\rangle + \beta_0\alpha_1|10\rangle + \beta_0\beta_1|11\rangle \end{aligned} \quad (2.18)$$

olurdu. Buradan,

$$\begin{aligned} \alpha_0\alpha_1 &= \frac{1}{\sqrt{3}}, \\ \alpha_0\beta_1 &= 0, \\ \beta_0\alpha_1 &= 0, \\ \beta_0\beta_1 &= \frac{1}{\sqrt{3}} \end{aligned}$$

olmalı. Bu durumda $\alpha_0\alpha_1$ ve $\beta_0\beta_1$ sifira eşit değildir. $|00\rangle$ ve $|11\rangle$ gözlemlenme olasılığı $1/3$ dir. Önemli nokta: iki kubitlik kuantum sistemde, ilk kubitin $|0\rangle$ olma olasılığını $|00\rangle$ verir ve $|\alpha_0\alpha_1|^2 = 1/3$ dir, ikinci kubit de aynı sonucu verir. Aynı şekilde ikinci kubitin $|1\rangle$ olma olasılığını da $|11\rangle$ verir ve $|\beta_0\beta_1|^2 = 1/3$ dir. Burada ikinci kubit gözlemlenmeden, sadece birinci kubit gözlemleyerek ikinci kubitin durumu bilinebilir. Bu dolanıklık sayesinde mümkündür. Bir çeşit bilgi transferidir. Dolanık durumda olan iki kubitin birini sonsuz uzaklıkta olsa dahi, birinin üzerinde ölçüm yapıldığında diğer kubitin de durumunun ölçüm bilgisini elde edilebilir. Kuantum bilgisayarların klasik bilgisayarlara göre üstünlük kazanmasını sağlayan bir özellik dolanıklıktır.

2.3.5 Durum Evrilmesi

Bir kuantum sisteminin durumlarının evrilmesi, üniter işlemcilerin durumlara uygulanmasıyla gerçekleşir. Kuantum hesaplamalarda tüm dönüşümler üniter olmalıdır. Hesaplamanın üniter olmasının ilk sebebi, izole bir kuantum sistemin *tersinir* olmasıdır. Üniter işlemciler her zaman tersinir işlemcilerdir. Üniter matrislerin tersi, kompleks eşleniğinin transpozesinin tersine eşittir(adjoint). Eğer U üniter işlemci ve v bir vektör ise:

$$\begin{aligned} U^\dagger &= (U^T)^* \\ U^\dagger Uv &= Iv = v \end{aligned} \quad (2.19)$$

dir. Üniter işlemcilerin kullanılmasının ikinci sebebi ise bu işlemciler, uzayın iç çarpımını ve vektör uzunluğunu korumasıdır. Her kuantum durum evrilmesi için bir U üniter işlemci vardır. $|\psi_1\rangle$ durum evrilmesi,

$$|\psi_2\rangle = U|\psi_1\rangle \quad (2.20)$$

şeklinde olur.

2.3.6 Gözlemlenebilirler ve Ölçümler

Klasik makinelerde işlem sonuçları hemen alınabiliyorken, kuantum sistemde kuantum durumlar doğrudan gözlemlenemediğinden, klasik bit düzeyinde bilgi alabilmek için bir ölçüm işlemi yapılır. Kuantum durumun ölçümü, olası baz durumlardan birisidir. Kuantum durum $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ için, $|\alpha_0|^2 + |\alpha_1|^2 = 1$ olmak üzere, ölçüm, $i \in \{0, 1\}$, durum vektörünün $|\alpha_i|^2$ olasılıkla $|i\rangle$ baz durumlardan birinin izdüşümünü verir. Kuantum durum $|\psi\rangle$, herhangi bir $|\phi\rangle$ kuantum duruma yansıtmak için, bir P izdüşüm matrisi tanımlamak gerekir. P izdüşüm matrisi,

$$P = |\phi\rangle\langle\phi| \quad (2.21)$$

olarak tanımlanır. Kuantum durum $|\psi\rangle$ üzerindeki izdüşümü,

$$P|\psi\rangle = |\phi\rangle\langle\phi||\psi\rangle \quad (2.22)$$

denklemini sağlar.

Bir kuantum sistemde ölçüm, kuantum devrenin gözlemlenebilir sonuçların hesaplanmasıdır. Kuantum makinelerin klasik makinelerden farkı, klasik sistemin gözlemlenmenin sistemin durumunu değiştirmemesidir. Kuantum sistemler gözlemlendiğinde kuantum mekaniksel özellikler ortadan kaybolur ve parçacıklar klasik fizik kurallarına göre davranmaya devam eder. Bu yüzden kuantum sistemin gözlemlenebilmesi ve daha sonrasında sistemin kuantum durumunun ne olduğunun anlaşılabilmesi için Hermisyen matrisler kullanılır.

\mathcal{M} , $n \in N$ için özdeğerleri λ_n ve birbirine dik olan özvektörleri $|\phi_n\rangle$ olan bir Hermisyen matris olsun. Bu halde $|\phi_n\rangle$ izdüşümlerine karşılık gelen P_n izdüşüm matrisi (dış çarpım ile elde edilir),

$$P_n = |\phi_n\rangle\langle\phi_n| \quad (2.23)$$

olur. \mathcal{M} gözlemlenebilir matrisi ise,

$$\mathcal{M} = \sum \lambda_n |\phi_n\rangle\langle\phi_n| \quad (2.24)$$

şeklinde ifade edilir. Ölçüm sonucu λ_n özdeğerini elde etme olasılığı $Pr(\lambda_n)$ denklemi:

$$Pr(\lambda_n) = \langle\psi|P_n|\psi\rangle = tr[P_n|\psi\rangle\langle\psi|] \quad (2.25)$$

Kuantum durum olasılık genliği $\alpha = \langle\phi_n|\psi\rangle$ ölçüm sonrası $Pr(\lambda_n)$ olasılığı,

$$Pr(\lambda_n) = \frac{|\langle\phi_n|\psi\rangle|^2}{\langle\psi|\psi\rangle} = \frac{|\alpha_n|^2}{\langle\psi|\psi\rangle} \quad (2.26)$$

Burada $\langle\psi|\psi\rangle = 1$ dir. Ölçüm sonucu, kuantum durumu çöker, sistem ψ_n durumda kalır. Sistemin son durumu izdüşüm matrisi $P_n = |\phi_n\rangle\langle\phi_n|$ cinsinden,

$$|\psi\rangle \rightarrow \frac{P_n|\psi\rangle}{\sqrt{\langle\psi|P_n|\psi\rangle}} \quad (2.27)$$

2.3.7 Yoğunluk Matrisi ve Karışık Durumlar

Kuantum devrede, saf kuantum durum, $\alpha_i \in \mathbb{C}$ ve $\sum_i |\alpha_i|^2 = 1$ olmak üzere,

$$|\psi\rangle = \sum_i \alpha_i |i\rangle \quad (2.28)$$

formundadır. Bir kuantum devrede, çevresel müdahalelerden ya da ön görülemeyen nedenlerden dolayı kuantum durum korunamaz ve durumun göreceli faz değeri ölçülemez. Bu sebeple kuantum durumun istatistiksel bir ortalaması alınır ve *karışık durum* denir. Karışık durumlar, *yoğunluk matrisi* denilen matris ile ifade edilir. Bir kuantum devrenin $|\psi_i\rangle$ saf durumları verilsin ve $|\psi_i\rangle$ durumunda olma olasılığı p_i olsun.

p_i için $0 \leq p_i \leq 1$ ve $\sum_i p_i = 1$ dir. Bu saf durumları olasılık dağılımı:

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i| \quad (2.29)$$

yoğunluk matrisidir. Yoğunluk matrisinin bazı özellikleri:

- (i) $\rho^\dagger = \rho$.
- (ii) $\text{tr}(\rho) = 1$.
- (iii) Eğer λ sayısı ρ matrisinin bir özdeğeri ise, $\lambda \geq 0$.

Bir \mathbf{A} gözlemlenebilirin beklenen değeri:

$$\begin{aligned}\langle \mathbf{A} \rangle &= \sum_j p_j \langle \psi_j | \mathbf{A} | \psi_j \rangle \\ &= \sum_j p_j \text{tr}(|\psi_j\rangle \langle \psi_j | \mathbf{A}) \\ &= \sum_j \text{tr}(p_j |\psi_j\rangle \langle \psi_j | \mathbf{A}) \\ &= \text{tr}\left(\sum_j p_j |\psi_j\rangle \langle \psi_j | \mathbf{A}\right) \\ &= \text{tr}(\rho \mathbf{A})\end{aligned}\tag{2.30}$$

olarak tanımlanır.

2.3.8 Kuantum Mantık Kapıları

Kuantum devrelerde, kubitlere kuantum mantık kapıları uygulanarak durum vektörlerinin yönelimleri değiştirilebilir. Kuantum kapıları kuantum operatörleridir ve üniter matrisler ile temsil edilirler. Bu yüzden bir birimdir ve terslenebilir işlemler yaparlar. Bir, iki ve üç kubitte etki eden kapılar açıklanmıştır [11, 12].

2.3.8.1 Birim Kapı (I)

Kimlik operatörüdür. Tek kubit için birim kapısı şöyle tanımlanır:

$$\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = |0\rangle\langle 0| + |1\rangle\langle 1|\tag{2.31}$$

$$\begin{aligned}\mathbf{I}|0\rangle &= |0\rangle \\ \mathbf{I}|1\rangle &= |1\rangle\end{aligned}\tag{2.32}$$

Birim kapı $|\psi\rangle$ kuantum durum üzerindeki etkisi,

$$\begin{aligned}
 |\psi\rangle &= \alpha|0\rangle + \beta|1\rangle \\
 \mathbf{I}|\psi\rangle &= (|0\rangle\langle 0| + |1\rangle\langle 1|)(\alpha|0\rangle + \beta|1\rangle) \\
 &= (\alpha|0\rangle|0\rangle\langle 0| + \alpha|0\rangle|1\rangle\langle 1|)(\beta|1\rangle|0\rangle\langle 0| + \beta|1\rangle|1\rangle\langle 1|) \\
 &= \alpha|0\rangle + \beta|1\rangle
 \end{aligned} \tag{2.33}$$

şeklinde ifade edilir.

2.3.8.2 Pauli Kapıları (X, Y, Z)

Pauli kapıları tek bir kübit üzerine etki eder. Üç farklı Pauli kapısı vardır ve bunlar X, Y ve Z sembolleriyle gösterilir. Bloch küre üzerinde, sırasıyla x, y ve z eksenini etrafında 180 derece dönüştürür.

Tablo 2.1 Pauli Kapıları

Kapı	Sembol	Matris Gösterimi
Pauli X	σ_x	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli Y	σ_y	$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli Z	σ_z	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ kuantum durum üzerindeki Pauli kapılarının etkileri:

2.3.8.3 Pauli X Kapısı

X kapısı kuantum durum vektörü, Bloch küresi üzerinde x eksenini etrafında π kadar döndürmektedir. X kapısının $|\psi\rangle$ kuantum durumuna etkisi aşağıdaki gibidir:

$$\mathbf{X}|\psi\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} 0\alpha + 1\beta \\ 1\alpha + 0\beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix} \tag{2.34}$$

Klasik bilgisayarlardaki NOT kapısıyla aynı olduğu görülmektedir. X kapısının $|0\rangle$ ve $|1\rangle$ baz durumlarına etkisi aşağıdaki gibidir:

$$\begin{aligned} X|0\rangle &= |1\rangle \\ X|1\rangle &= |0\rangle \end{aligned} \quad (2.35)$$

$|0\rangle$ ve $|1\rangle$ vektörlerinin tersini almaktadır.

2.3.8.4 Pauli Y Kapısı

Y kapısı kuantum durum vektörü, Bloch küresi üzerinde y eksenini etrafında π kadar döndürmektedir. Y kapısının $|\psi\rangle$ kuantum durumuna etkisi aşağıdaki gibidir:

$$Y|\psi\rangle = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} 0\alpha - i\beta \\ i\alpha + 0\beta \end{bmatrix} = \begin{bmatrix} -i\beta \\ i\alpha \end{bmatrix} \quad (2.36)$$

Y kapısının $|0\rangle$ ve $|1\rangle$ baz durumlarına etkisi aşağıdaki gibidir:

$$\begin{aligned} Y|0\rangle &= i|1\rangle \\ Y|1\rangle &= -i|0\rangle \end{aligned} \quad (2.37)$$

2.3.8.5 Pauli Z Kapısı

Z kapısı kuantum durum vektörü, Bloch küresi üzerinde z eksenini etrafında π kadar döndürmektedir. Z kapısının $|\psi\rangle$ kuantum durumuna etkisi aşağıdaki gibidir:

$$Z|\psi\rangle = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} 1\alpha + 0\beta \\ 0\alpha - 1\beta \end{bmatrix} = \begin{bmatrix} \alpha \\ -\beta \end{bmatrix} \quad (2.38)$$

Z kapısının $|0\rangle$ ve $|1\rangle$ baz durumlarına etkisi aşağıdaki gibidir:

$$\begin{aligned} Z|0\rangle &= |1\rangle \\ Z|1\rangle &= -|1\rangle \end{aligned} \quad (2.39)$$

2.3.8.6 Faz Kaydırma Kapısı($P(\theta)$)

$P(\theta)$ kapısı, Bloch küresi üzerinde belli bir θ açısı kadar döndürmek için kullanılır. Bu kapı uygulandıktan sonra $|0\rangle$ ve $|1\rangle$ baz durumlarının ölçme olasılığı değişmez, ancak

kuantum durumunun fazını deęiřtirir. $\mathbf{P}(\theta)$,

$$\mathbf{P}(\theta) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix} = |0\rangle\langle 0| + e^{i\theta}|1\rangle\langle 1| \quad (2.40)$$

ile tanımlanır. $\mathbf{P}(\theta)$ kapısının $|0\rangle$ ve $|1\rangle$ baz durumlarına etkisi ařaęıdaki gibidir:

$$\begin{aligned} \mathbf{P}(\theta)|0\rangle &= |0\rangle \\ \mathbf{P}(\theta)|1\rangle &= e^{i\theta}|1\rangle \end{aligned} \quad (2.41)$$

θ açısı için faz kapıları ařaęıdaki gibidir:

Tablo 2.2 Faz Kaydırma Kapı

θ	Kapı	Matris Gösterimi
$\theta = \pi$	Z	$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix}$
$\theta = \pi/2$	S	$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}$
$\theta = \pi/4$	T	$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta/4} \end{bmatrix}$

2.3.8.7 Hadamard Kapısı(H)

En önemli kapılardan biri *Hadamard dönüşümü* olarak bilinir. Hadamard dönüşümü aslında baz durumlardan oluşan ve eşit genlik deęerlerine sahip bir üst üste gelme durumu türetir. Tek kubit için Hadamard dönüşümü:

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \quad (2.42)$$

řeklindedir.

\mathbf{H} kapısının $|0\rangle$ ve $|1\rangle$ baz durumlarına etkisi ařaęıdaki gibidir:

$$\mathbf{H}|0\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \quad (2.43)$$

$$\mathbf{H}|1\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$$

2.3.8.8 Rotasyon(Döndürme) Kapıları

Kuantum durum vektörlerini Bloch küresi üzerinde x, y ve z eksenleri etrafında θ açısı kadar döndürmek için kullanılır.

Tablo 2.3 Rotasyon Kapıları

Kapı	Matris Gösterimi
$R_x(\theta)$	$\begin{bmatrix} \cos(\frac{\theta}{2}) & -i \sin(\frac{\theta}{2}) \\ -i \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{bmatrix}$
$R_y(\theta)$	$\begin{bmatrix} \cos(\frac{\theta}{2}) & -i \sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{bmatrix}$
$R_z(\theta)$	$\begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix}$

2.3.8.9 Kontrollü kapılar

Kontrollü kapılar iki veya daha fazla kubitte etki eder, bir veya daha fazla kubit kontrol işlevi görevi görür. Örneğin kontrollü NOT(CNOT veya CX) kapısı iki kubit üzerinde etki eder, kontrol kubit $|0\rangle$ ise hedef kubitte değişiklik olmaz, eğer kontrol kubit $|1\rangle$ ise hedef kubitte **X** kapısı uygulanır. **CNOT** aşağıdaki şekilde gösterilir:

$$\mathbf{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (2.44)$$

CNOT kapısının $|00\rangle$, $|01\rangle$, $|10\rangle$ ve $|11\rangle$ baz durumlarına etkisi aşağıdaki gibidir:

$$\begin{aligned} \mathbf{CNOT}|00\rangle &= |00\rangle \\ \mathbf{CNOT}|01\rangle &= |01\rangle \\ \mathbf{CNOT}|10\rangle &= |11\rangle \\ \mathbf{CNOT}|11\rangle &= |10\rangle \end{aligned} \quad (2.45)$$

Tek kubit üzerinde çalışan kapı **U** matrisi olsun ve gösterimi:

$$\mathbf{U} = \begin{bmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{bmatrix}$$

olsun. Bu kapıya kontrollü **U** kapısı denir, iki kubit üzerinde çalışırken ilk kubit kontrol görevi görür. Genel matris gösterimi:

$$\mathbf{CU} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{bmatrix} \quad (2.46)$$

Pauli Kapılarla çalışırken, kontrollü-**X**(CX), kontrollü-**Y**(CY) ve kontrollü-**Z**(CZ) diye adlandırılır.

2.3.8.10 SWAP kapısı

SWAP kapısı kuantum durumundaki kubitlerin yerini değiştirir. Matris gösterimi:

$$\mathbf{SWAP} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.47)$$

şeklinde. **SWAP** kapısı $|00\rangle$, $|01\rangle$, $|10\rangle$ ve $|11\rangle$ baz durumlarına etkisi aşağıdaki gibidir:

$$\mathbf{SWAP}(\alpha_0|00\rangle + \alpha_1|01\rangle + \alpha_2|10\rangle + \alpha_3|11\rangle) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} \alpha_0 \\ \alpha_2 \\ \alpha_1 \\ \alpha_3 \end{bmatrix} \quad (2.48)$$

2.3.8.11 Toffoli (CCNOT) Kapısı

CCNOT kapısı kuantum durumundaki üç kubit üzerinde çalışır. Kontrol kubitler ilk iki kubit $|11\rangle$ durumunda ise hedef kubitte **X** kapısı etkisi gösterir. Matris gösterimi:

$$\text{CCNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.49)$$

2.3.8.12 Fredkin (CSWAP) Kapısı

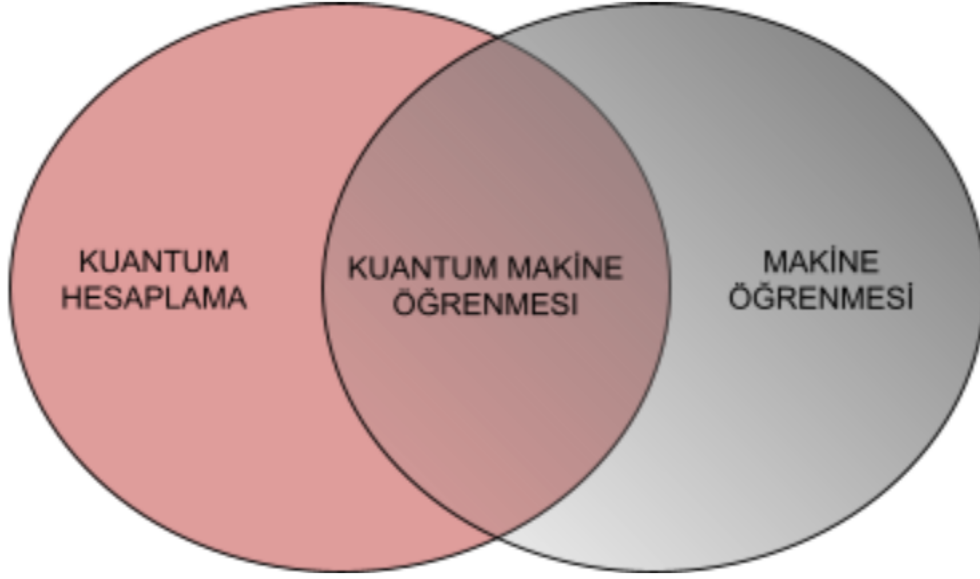
Fredkin kapısı kuantum durumundaki üç kubit üzerinde çalışır ve kontrollü-yer değiştirme kapısı olarak da bilinir. Kontrol kubitler ilk iki kubit $|11\rangle$ durumunda ise hedef kubite X kapısı etkisi gösterir. Matris gösterimi:

$$\text{CSWAP} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.50)$$

2.4 Kuantum Makine Öğrenmesi

Günümüzde hala birçok klasik makine öğrenmesi yöntemleri kullanılmakta ve geliştirilmeye devam etmektedir. Ancak beraberinde hem zaman hem de maliyet gibi büyük zorluklar meydana getirmektedir. Bu zorluklarla başedebilmek için sunulan alternatiflerden biri de kuantum bilgisayarlardır [7]. Makine öğrenmesi yöntemlerini klasik bilgisayarlar yerine kuantum bilgisayarlarda uygulanmasına Kuantum Makine Öğrenmesi denir. Şekil 2.2'deki gibi Kuantum Makine Öğrenmesi ifade edilebilir.

Kuantum bilgisayarların en büyük avantajı, atom altı parçacıklarının aynı anda birçok durumda (hem 0 hem 1) olabilme özelliği ile, klasik bilgisayarların yapamadığı birçok olası durumların hesaplamasını yapabilir ve karmaşıklık açısından da önemli ölçüde hızlanmasını sağlayabilir olmasıdır [13]. Diğer bir önemli avantajlarından biri de kuantum bilgisayarların sahip olduğu özellikler (dolanıklık, süperpozisyon gibi.)



Şekil 2.2 Kuantum Makine Öğrenmesi

sayesinde anlık, sezgisel durumlar üretebilmesidir [14].

Yakın gelecekte kuantum bilgisayarların daha da geliştirilmesiyle kuantum makine öğrenmesi disiplininde araştırmaların ve deneylerin sayısının artması beklenmektedir [7]. Verinin kuantum duruma getirilmesi, bu kuantum durumlara uygulanacak olan yöntemlerin, uygulanan algoritmanın yeni bir makine öğrenmesi yöntemi sunabilirliği ve sonuçları gibi konular üzerinde durulmaktadır [15].

Klasik ve kuantum sistemlerde kullanılan makine öğrenmesi yöntemi olarak dört yaklaşım vardır (Şekil 2.3). Bu yaklaşımlar verilerin hangi sistem tarafında üretilip ve hangi sistem tarafından hesaplandığına bağlı olarak birleştirilir. Veriler klasik makinelerde üretilip ve hesaplanıyorsa geleneksel makine öğrenimidir (CC). Diğer yaklaşımlarda, veriler kuantum makinelerde üretilip, hesaplamalar klasik makinelerde yapılır (QC); veriler klasik makinelerde üretilip ve hesaplamalar, analizler kuantum makinelerde yapılır (CQ). Son olarak verileri kuantum makinelerde üretilip, hesaplamalar kuantum makinelerde olur (QQ) [16].

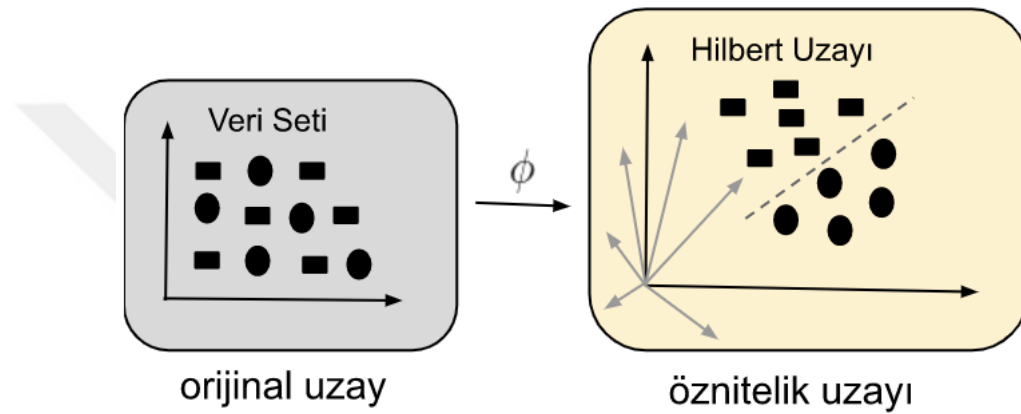
Veri Tipi	Algoritma Tipi	
	Classic	Quantum
Classic	CC	CQ
Quantum	QC	QQ

Şekil 2.3 Klasik/Kuantum makine öğreniminde kullanılan dört yaklaşım [16]

Bugün mevcut olan kuantum bilgisayarlarda kısıtlı çok az fiziksel kubit vardır [17]. Bu noktada en önemli unsur, kuantum durumlarda klasik verilerin nasıl temsil edileceğidir. Literatürde farklı veri kodlama türü tanımlanmıştır [5]. Uygulanan veri kodlama türü performansı direkt etkilediği için veri kümesine en uygun veri kodlama türünün seçimi önem arz etmektedir.

2.4.1 Kuantum Veri Kodlama ve Kuantum Çekirdek

Bu kısımda [3], [4] ve [5] çalışmalarından yararlanılarak kuantum veri kodlama ve çekirdek çeşitleri verilmiştir.



Şekil 2.4 $x \rightarrow \phi(x)$ dönüşümü ([3]'den adapte edilmiştir.)

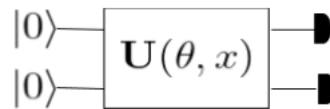
Kuantum veri kodlama, gerçel bir x değeri Hilbert uzayında $|\phi(x)\rangle$ kuantum durum vektörü ile temsil edilmesidir. $x \rightarrow |\phi(x)\rangle$ dönüşüm (Şekil 2.4 verinin orijinal uzaydan daha büyük boyutlu olan öznitelik uzayına haritalanmasıdır.) adımları aşağıdaki gibidir,

$$x \in \mathcal{X},$$

Hilbert uzayında, $U(x)$ üniter işlemci ve $|\psi\rangle = |0\rangle$ kuantum durum vektörü olsun.

$$U(x)|\psi\rangle = |\phi(x)\rangle$$

$$U(x)|0\rangle = |\phi(x)\rangle$$



Şekil 2.5 U üniter işlemci

Burada $U(x)$ öznitelik kodlama devresi, kubitin $|\psi\rangle$ kuantum durum vektörü üzerinde çalışır. $|\phi(x)\rangle$ kuantum durum vektörü Hilbert uzayında, $U(x)$ 'in $|\psi\rangle$ kuantum

durum vektörüne uygulanmasıyla elde edilir. Hilbert uzayında, "öznitelik-gömülü devre(feature-embedding circuit)"kodlaması yapılmıştır.

$x \rightarrow |\phi(x)\rangle$ dönüşümü bazı durumlarda doğrusal olmayan fonksiyonlar olabilir [3]. Bunu iyileştirmek için, kuantum hesaplamanın sağlamış olduğu yoğunluk matrisi özelliği kullanılarak, $x \rightarrow \rho(x)$ dönüşümü yapılır [4]. Bu dönüşüm "veri-öznitelik haritası kodlama" olarak adlandırılır.

Tanım 2.13. (Veri-Öznitelik Haritası Kodlama(Data-encoding feature map)).

$|\psi\rangle$ vektörü, n kubitlik kuantum sistemi verilsin. \mathcal{F} , $2^n \times 2^n$ boyutlu kompleks değerli matrislerin uzayı olmak üzere $\rho, \sigma \in \mathcal{F}$ için $\langle \rho, \sigma \rangle_{\mathcal{F}} = \text{tr}\{\rho^\dagger \sigma\}$ olarak Hilbert Chemit ile donatılmış olsun. Veri-Öznitelik Haritası Kodlama:

$$\phi : \mathcal{X} \rightarrow \mathcal{F}, \quad (2.51)$$

$$\phi(x) = |\phi(x)\rangle\langle\phi(x)| = \rho(x) \quad (2.52)$$

dönüşümü olarak tanımlanır.

Kubitin $\rho(x)$ yoğunluk matrisi \mathcal{F} uzayında ölçüm yapabilmeyi sağlar.

Tanım 2.14. (Kuantum Çekirdek). ϕ , \mathcal{X} tanım kümesi üzerinde bir veri-öznitelik haritası kodlaması olsun. Bir kuantum çekirdek, $x, x' \in \mathcal{X}$ olmak üzere $\rho(x)$ ve $\rho(x')$ veri-öznitelik haritası kodlaması vektörleri arasındaki iç çarpımdır:

$$\kappa(x, x') = \text{tr}[\rho(x')\rho(x)] = |\langle\phi(x')|\phi(x)\rangle|^2 \quad (2.53)$$

Kuantum çekirdeğin pozitif tanımlı bir fonksiyon olduğu aşağıda gösterilmiştir [4].

Çekirdek, $\kappa_c(x, x') = \langle\phi(x')|\phi(x)\rangle$ karmaşık sayılardan oluşur ve $\kappa_c(x, x')^* = \langle\phi(x')|\phi(x)\rangle^*$ karmaşık sayıların eşlenik özelliğinden, çekirdeğin eşleniği de bir çekirdektir.

$x^m \in \mathcal{X}, m = 1, \dots, M$ ve $c_m \in \mathbb{C}$

$$\sum_{m, m'} c_m c_{m'}^* (\kappa_c(x^m, x^{m'}))^* = \sum_{m, m'} c_m c_{m'}^* \langle\phi(m)|\phi(m')\rangle \quad (2.54)$$

$$= \left\| \sum_m c_m^* |\phi(m')\rangle \right\|^2 \quad (2.55)$$

$$\geq 0 \quad (2.56)$$

Burada eşlenik $\kappa_c(x, x')^*$ pozitif olduğu görülür.

Yukarıda çekirdek ve eşleniği verilmiştir. İki çekirdeğin iç çarpımı ile elde edilen çekirdeğin mutlak karesi, kuantum çekirdeğin pozitif tanımlı fonksiyonudur. Her $x^m \in \mathcal{X}$ için pozitif tanımlı fonksiyonlar elde edilir ve bu fonksiyonlar $M \times M$ boyutlu bir matristir. $M \times M$ matrisi çekirdeğin Gram matrisi olarak ifade edilebilir.

Bu bölümde genel olarak verinin kuantum durumlarda kodlanması ve çekirdek oluşturulması verilmiştir. Aşağıda kuantum veri kodlama çeşitleri ve öznitelik haritası kodlamasının oluşturduğu kuantum çekirdek çeşitleri [4]'den yararlanılarak verilecektir.

2.4.1.1 Temel kodlama

Klasik bit düzeyine karşılık gelen kubit sistemidir. Örneğin, $x=1001$ ikili ifadesi 4 kubit ile kuantum durum $|1001\rangle$ ile temsil edilir. \mathcal{X} veri kümesi M satır örnek, her bir örneği de N boyutlu olsun.

$$\mathcal{X} = \{x^1, \dots, x^m, \dots, x^M\}$$

Her bir örnek N -bit ikili dizi olmalı;

$$x^m = (b_1, \dots, b_N), b_i \in \{0, 1\}, i = 1, \dots, N.$$

x^m girdisi, kuantum durum vektörü $|x^m\rangle$ ile eşleşir. Tüm veri temel kodlamada:

$$|\mathcal{X}\rangle = \frac{1}{\sqrt{M}} \sum_{m=1}^M |x^m\rangle \quad (2.57)$$

şeklinde ifade edilir.

Örneğin $x^{(1)} = 01$ ve $x^{(2)} = 11$ ikili diziyeye dönüştürülmüş olsun, kuantum sistemde $|x^{(1)}\rangle = |01\rangle$ ve $|x^{(2)}\rangle = |11\rangle$ şeklinde temsil edilir. İkili kubit sistem gereklidir. Tüm verinin temel kodlaması ise:

$$|\mathcal{X}\rangle = \frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|11\rangle \quad (2.58)$$

.

Temel kodlama ile kuantum çekirdeğin elde edilmesi:

x , \mathcal{X} kümesinin elemanı ve N uzunluğunda ikili dizi olsun. Her ikili dizinin, tekil

tamsayı değeri i_x aşağıdaki şekilde elde edilir:

$$i_x = \sum_{k=0}^{N-1} 2^k x_k \quad (2.59)$$

ve i_x için kuantum durum vektörü $|i_x\rangle$ ile gösterilir. Buradan ikili dizinin, öznitelik haritası kodlaması ile eşleştirilmesi:

$$\phi : x \rightarrow |i_x\rangle\langle i_x| \quad (2.60)$$

şeklinde olur. Kuantum çekirdek, öznitelik harita kodlamasının iç çarpımından Kronecker deltasından elde edilir.

$$\kappa(x, x') = |\langle i_{x'} | j_x \rangle|^2 = \delta_{x, x'} \quad (2.61)$$

Temel kodlama için $\mathcal{O}(N)$ adet kubit gereklidir.

2.4.1.2 Genlik kodlaması

M örnekli \mathcal{X} veri kümesinde, herbir veri $\mathbf{x} \in \mathcal{X}$ kuantum durum vektörü $|\psi_{\mathbf{x}}\rangle$:

$$|\psi_{\mathbf{x}}\rangle = \sum_{i=1}^N x_i |i\rangle \quad (2.62)$$

ile tanımlanır. Burada $N = 2^n$ genlik sayısı, x_i için \mathbf{x} vektörün i . bileşeni, $|i\rangle$ vektörü de x_i bileşeninin temel kodlanmış kuantum durum vektörüdür. Genlik kodlaması uygularken ilk adım olarak \mathbf{x} vektörü normalize edilir:

$$|\psi_{\mathbf{x}_{norm}}\rangle = \frac{\sum_{i=1}^N x_i |i\rangle}{\|\mathbf{x}\|} = 1 \quad (2.63)$$

Tüm veri kümesi \mathcal{X} için kuantum durum:

$$\alpha = \mathcal{X}_{norm} \{x_1^{(1)} \dots x_N^{(1)}, \dots, x_1^{(M)} \dots x_N^{(M)}\}, |\alpha|^2 = 1$$

$$|\mathcal{X}\rangle = \sum_{i=1}^{2^n} \alpha_i |i\rangle \quad (2.64)$$

şeklinde ifade edilir.

Örnek, $\mathbf{x} = (1.1, 0.0, 1.2, 5.5)$ olsun. \mathbf{x} vektörünün normalize edilmesi ile

$\mathbf{x}_{norm} = \frac{1}{\sqrt{32.9}}(1.1, 0.0, 1.2, 5.5)$ elde edilir. Kuantum durum vektörü:

$$|\mathbf{x}_{norm}\rangle = \frac{1}{\sqrt{32.9}}[1.1|00\rangle + 1.2|10\rangle + 5.5|11\rangle]$$

olur.

Veri kümesi \mathcal{X} için kodlanmış genlik sayısı $N \times M$ kadar olur. Eğer bir kuantum sistem n kubitten oluşursa, n için $n \geq \log_2(NM)$ olmalıdır [18]. Hesaplama karmaşıklığı $\mathcal{O}(2^n)$ dir.

Genlik kodlaması ile kuantum çekirdeğin elde edilmesi:

$\mathbf{x} \in \mathcal{X}$ ve N boyutlu olsun. \mathbf{x} için genlik kodlama ile öznitelik haritası kodlaması:

$$\|\mathbf{x}\|^2 = \sum_i^N |x_i|^2 = 1 \quad (2.65)$$

$$\phi : \mathbf{x} \rightarrow |\mathbf{x}\rangle\langle\mathbf{x}| = \sum_{i,j=1}^N x_i x_j^* |i\rangle\langle j| \quad (2.66)$$

şeklinde ifade edilir. Öznitelik haritalama kodlamalarının iç çarpımından elde edilen kuantum çekirdek, lineer çekirdeğin mutlak karesidir:

$$\kappa(\mathbf{x}, \mathbf{x}') = |\langle\mathbf{x}'|\mathbf{x}\rangle|^2 = |\mathbf{x}^\dagger \mathbf{x}'|^2 \quad (2.67)$$

Genlik kodlama için $\mathcal{O}(N)$ adet kubit gereklidir.

2.4.1.3 Tekrarlı Genlik kodlaması

Genlik kodlamasının r defa tekrarlandığı durumlardır. \mathbf{x} için tekrarlı genlik kodlama ile öznitelik haritası kodlaması:

$$\phi : \mathbf{x} \rightarrow |\mathbf{x}\rangle\langle\mathbf{x}| \otimes \dots \otimes |\mathbf{x}\rangle\langle\mathbf{x}| \quad (2.68)$$

şeklinde ifade edilir.

Tekrarlı genlik kodlaması ile kuantum çekirdeğin elde edilmesi:

$$\kappa(\mathbf{x}, \mathbf{x}') = (|\langle \mathbf{x}' | \mathbf{x} \rangle|^2) = (|(\mathbf{x}')^\dagger \mathbf{x}|^2)^r \quad (2.69)$$

şeklinde dir. Tekrarlı genlik kodlama için $\mathcal{O}(rN)$ adet kubit gereklidir.

2.4.1.4 Rotasyon kodlaması

Veri, kubitlerin dönüş açısına göre kodlanmıştır. Veri kümesi \mathcal{X} 'den \mathbf{x} girdisinin i . bileşeni, i . kubitte Pauli rotasyon kuantum kapıları ile kodlanır. Örneğin, Pauli Y rotasyon kapısı uygulanmış kubitin kuantum durumu:

$$|q_i(x_i)\rangle = \cos(x_i)|0\rangle + \sin(x_i)|1\rangle \quad (2.70)$$

ile ifade edilir.

Rotasyon kodlama ile öznitelik haritası kodlaması:

$$|\mathbf{x}\rangle = \sum_{q_1, \dots, q_N=0}^1 \prod_{k=1}^N \cos(x_k)^{q_k} \sin(x_k)^{1-q_k} |q_1, \dots, q_N\rangle \quad (2.71)$$

$$\phi : \mathbf{x} \rightarrow |\mathbf{x}\rangle \langle \mathbf{x}| \quad (2.72)$$

şeklinde ifade edilir.

Rotasyon kodlaması ile kuantum çekirdeğin elde edilmesi:

Öznitelik haritası kodlamasının iç çarpımından hesaplanan kuantum çekirdek, kosinüs çekirdek olarak ifade edilir. Kuantum çekirdek:

$$\kappa(\mathbf{x}, \mathbf{x}') = \prod_{k=1}^N |\sin x_k \sin x'_k + \cos x_k \cos x'_k|^2 = \prod_{k=1}^N |\cos(x_k - x'_k)|^2 \quad (2.73)$$

şeklinde ifade edilir.

Rotasyon kodlama için $\mathcal{O}(N)$ adet kubit gereklidir.

2.4.1.5 Uyumlu durum kodlaması

Uyumlu durumlar, kuantum optik alanında ışık modlarını tanımlamak için kullanılır. Kubitin kuantum durumu, $\{|0\rangle, |1\rangle, |2\rangle, \dots\}$ kümesinin ayırık baz durumlardan oluşur. Uyumlu durum formu:

$\alpha \in \mathbb{C}$ olmak üzere,

$$|\alpha\rangle = e^{-\frac{|\alpha|^2}{2}} \sum_{k=0}^{\infty} \frac{\alpha^k}{\sqrt{k!}} |k\rangle \quad (2.74)$$

şeklinde tanımlanır.

Herhangi bir $\mathbf{x} \in \mathcal{X}$ için i . bileşeni x_i 'in uyumlu durumu formu $|\alpha_{x_i}\rangle$ olarak ifade edilir. x_i için uyumlu durum kodlama ile öznitelik haritası kodlaması:

$$|\alpha_{x_i}\rangle = e^{-\frac{|x_i|^2}{2}} \sum_{k=0}^{\infty} \frac{x_i^k}{\sqrt{k!}} |k\rangle \quad (2.75)$$

$$\phi : x_i \rightarrow |\alpha_{x_i}\rangle \langle \alpha_{x_i}| \quad (2.76)$$

olur.

$\mathbf{x} = (x_1, x_2, \dots, x_N)$ için uyumlu durum ile öznitelik haritası kodlaması:

$$|\alpha_{\mathbf{x}}\rangle \langle \alpha_{\mathbf{x}}| = |\alpha_{x_1}\rangle \langle \alpha_{x_1}| \otimes |\alpha_{x_2}\rangle \langle \alpha_{x_2}| \otimes \dots \otimes |\alpha_{x_N}\rangle \langle \alpha_{x_N}| \quad (2.77)$$

şeklinde ifade edilir.

Uyumlu durum kodlaması kuantum çekirdeğin elde edilmesi:

Uyumlu durum kodlama ile Gauss çekirdeği elde edilir [19]. Kuantum çekirdek:

$$\kappa(x, x') = |e^{-\left(\frac{|x|^2}{2} + \frac{|x'|^2}{2} - x^T x'\right)}|^2 \quad (2.78)$$

şeklinde ifade edilir.

2.4.1.6 Hilbert Uzayında Çekirdeğin Gösterimi

Tanım 2.15. (Doğuran Çekirdekli Hilbert Uzayları). $\mathcal{X} \neq \emptyset$ ve $\mathcal{H}; f : \mathcal{X} \rightarrow \mathbb{R}$ tanımlı fonksiyonların oluşturduğu Hilbert uzayı olsun. Eğer aşağıdaki özellikleri sağlayan $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ fonksiyonu mevcut ise $\mathcal{H} \langle \cdot, \cdot \rangle$ iç çarpımı ($\|f\| = \sqrt{\langle f, f \rangle}$) ile donatılmış doğuran çekirdek Hilbert uzayı olarak adlandırılır [9].

(i) $\forall f \in \mathcal{H}$ için $\langle f, \kappa(x, \cdot) \rangle = f(x)$
 sonuç olarak $\langle \kappa(x, \cdot), \kappa(x', \cdot) \rangle = \kappa(x, x')$

(ii) κ fonksiyonları \mathcal{H} Hilbert uzayını gerer.

2.4.2 Kuantum Modelin Ölçümü

Kuantum sistemlerde en önemli adımlardan biri de ölçümdür. Bu bölümde, verinin kuantum durumlarının, öznitelik haritası kodlaması üzerinden ölçümü verilecektir. Kuantum ölçüm, fiziksel olarak gözlemlenebilir bir \mathcal{M} Hermisyen operatörün özdeğerleridir. Özdeğerler, bir durum üzerindeki gözlem sonucunda gözlemin alabileceği olası tek değerlerdir [11].

Tanım 2.16. (Kuantum Modelin Ölçümü) \mathcal{X} tanım kümesi üzerinde x 'in $|\phi\rangle$ kuantum durum vektörü ve \mathcal{M} Hermisyen operatör olsun. x elemanı için beklenen kuantum ölçüm fonksiyonu:

$$\rho(x) = |\phi(x)\rangle\langle\phi(x)| \quad (2.79)$$

$$f(x) = \text{tr}[\rho(x)\mathcal{M}] \quad (2.80)$$

şeklinde ifade edilir. Tüm \mathcal{X} veri kümesi için:

$$f : \mathcal{X} \rightarrow \mathbb{R} \quad (2.81)$$

gösterilir.

$x \in \mathcal{X} \subseteq \mathbb{R}$ için $\mathbf{U}(x)$ üniter işlemci ve $|\psi\rangle$ kuantum durum vektörü olsun. $\mathbf{U}(x)$ üniter işlemcinin $|\psi\rangle$ kuantum durum vektörünün üzerinde çalışmasıyla $x \rightarrow |\phi(x)\rangle$ dönüşümü:

$$\mathbf{U}(x)|\psi\rangle = |\phi(x)\rangle \quad (2.82)$$

ile gösterilir.

x reel verisi için $x \rightarrow \rho$ dönüşümü:

$$\rho(x) = |\phi(x)\rangle\langle\phi(x)| \quad (2.83)$$

ile gösterilir.

\mathcal{X} reel veri kümesi için, \mathcal{F} uzayında, $\phi : x \in \mathcal{X} \rightarrow \rho(x) \in \mathcal{F}$ dönüşümü gerçekleşir.

x için beklenen kuantum ölçümü hesaplamak için f fonksiyonu:

$$f(x) = \text{tr}[\rho(x)\mathcal{M}] = \langle \phi(x) | \mathcal{M} | \phi(x) \rangle \quad (2.84)$$

şeklindedir.

Örnek, $x \in \mathcal{X} \subseteq \mathbb{R}$ reel girdisi ve tek kubit üzerinde Pauli-X rotasyon kodlamasıyla kuantum durumu $|\phi(x)\rangle = R_x(x)|0\rangle$ olsun.

Ölçüm \mathcal{M} bir Hermisyen matris ve boyutu 2×2 , keyfi bir $R(\theta_1, \theta_2, \theta_3)$ rotasyon operatörü ve ölçüm için Pauli-Z baz operatörü olsun.

$$R(\theta_1, \theta_2, \theta_3) = \begin{pmatrix} e^{i(-\frac{\theta_1}{2} - \frac{\theta_3}{2})} \cos(\frac{\theta_2}{2}) & -e^{i(-\frac{\theta_1}{2} + \frac{\theta_3}{2})} \sin(\frac{\theta_2}{2}) \\ e^{i(\frac{\theta_2}{2} - \frac{\theta_3}{2})} \sin(\frac{\theta_2}{2}) & e^{i(\frac{\theta_1}{2} + \frac{\theta_3}{2})} \cos(\frac{\theta_2}{2}) \end{pmatrix}$$

$$\mathcal{M}(\theta_1, \theta_2, \theta_3) = R^\dagger(\theta_1, \theta_2, \theta_3) \sigma_z R(\theta_1, \theta_2, \theta_3)$$

Kuantum modelde ölçüm fonksiyonu:

$$f(x) = \text{tr}[\rho(x)\mathcal{M}(\theta_1, \theta_2, \theta_3)] \quad (2.85)$$

$$= \langle \phi(x) | \mathcal{M}(\theta_1, \theta_2, \theta_3) | \phi(x) \rangle \quad (2.86)$$

formülünden, x için ölçüm:

$$f(x) = \cos(\theta_2) \cos(x) - \sin(\theta_1) \sin(\theta_2) \sin(x) \quad (2.87)$$

olur.

2.5 Kuantum Çekirdek Temelli Destek Vektör Makinesi

Son yıllarda kuantum makine öğrenimi araştırmaları ile, makine öğrenimi modelleri kuantum devreler ile nasıl eğitebilir ve yorumlanabilir konusunda pek çok çalışma yapılmıştır [7]. Makine öğrenimi modellerinden DVM'nin, kuantum çekirdek yöntemleri ile yakın ilişkili olduğu gösterilmiştir.[19].

Klasik bilgisayarlarda DVM ancak belirli bir boyuta kadar gerçekleştirilebilir. Belirli bir boyuttan sonra klasik bilgisayarların yeterli işlem gücü olmadığı için zor olacaktır. Kuantum hesaplamaların sağlamış olduğu kuantum durum, kuantum kapıları, veriyi kuantum durumlarına kodlama ve ölçüm ilkelerini kullanarak kuantum çekirdek temelli DVM algoritmasının daha verimli çalışmasını ve daha hızlı sonuç üretmesini

sağlar.

Kuantum çekirdek devresi temelde Şekil 2.6'da gösterildiği üzere üç adımdan oluşur:

- **Veriyi kodlama:**

x reel girdi, uygun öznitelik kodlama devresi ile $|\phi(x)\rangle$ kuantum durumuna hazırlanır.

- **İşlem:**

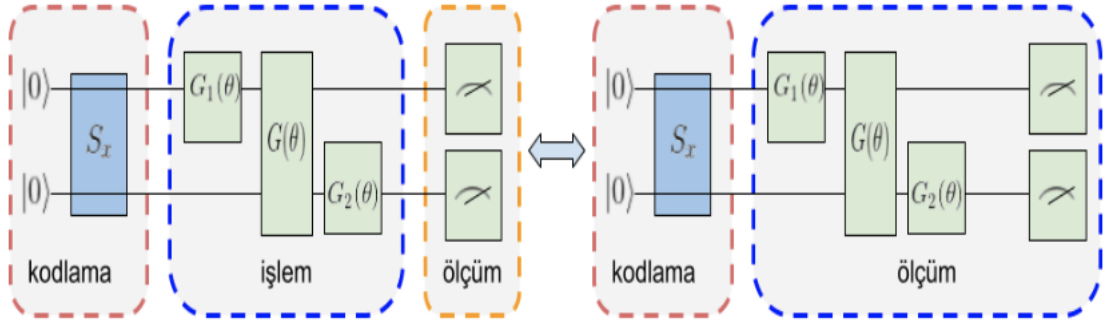
\mathcal{M} , Hilbert uzayında keyfi gözlemlenebilir matristir. Bu matris bir Hermisyen matris ve $2^n \times 2^n$ boyutlu olsun. $\mathbf{G}(\theta)$ uygulayıcı devre, uygulayıcı devre üzerinde gözlemlenebilir Pauli-Z σ_z^0 matristir. Gözlemlenebilir matris $\mathcal{M}(\theta)$:

$$\mathcal{M}(\theta) = \mathbf{G}(\theta)^\dagger \sigma_z^0 \mathbf{G}(\theta)$$

şeklindedir.

- **Ölçüm:**

Gözlenebilir $\mathcal{M}(\theta)$ matrisinin klasik bilgisayardaki değeri hesaplanır.



Şekil 2.6 Kuantum Devre ([4]'den adapte edilmiştir)

Klasik makine öğrenimindeki çekirdek yaklaşımında karar fonksiyonu f yi eğitmek yerine, bu çekirdeğe eşdeğer kuantum devrede yürütülen çekirdek ile model eğitilebilir. Bu kuantum çekirdek, iki girdi için kodlanan kuantum durumlarına karşılık gelen değerdir:

$$\kappa(x, x') = |\langle \phi(x') | \phi(x) \rangle|^2$$

Kuantum çekirdek ile veri eğitilirken, ortak kuantum durum işleme ve ölçümleme yapılmaz, sadece devreye giren girdinin kuantum durum kodlanmasına üzerinde çalışılır. Bu yüzden tüm veri kümesi için kuantum çekirdek $\kappa : \mathbf{X} \times \mathbf{X} \rightarrow \mathbb{K}, \mathbb{K} \subseteq \mathbb{C}$ olur.

3.1 Kuantum Çekirdek Tabanlı Destek Vektör Makinesi ile Parkinson Hastalığı Teşhisi

Son yıllarda Parkinsonlu hasta sayısının büyük artış göstermesi hastalığın teşhis ve tedavi yöntemlerine önem kazandırmıştır [20]. Hastalığın teşhisi sırasında kapsamlı bir dizi klinik test ve rapor sonuçlarının analiz edilmesi gerekmektedir. Bu süreç oldukça zordur ve çoğunlukla geleneksel klinik yöntemlere dayalıdır.

Sağlık sektöründe, geleneksel klinik değerlendirmelerin yanı sıra makine öğrenimi yöntemlerinin giderek daha fazla uygulanmasıyla, Parkinson hastalığının teşhisinde de bu makine öğrenimi tekniklerine ilişkin uygulamalar artmıştır. Bu yöntemlerden biri de kuantum makine öğrenme metodlarıdır.

Klasik sınıflandırma yöntemlerinden biri olan DVM, kuantum çekirdek fonksiyonu ile daha avantajlı hale gelir. Kuantum çekirdek tabanlı makine öğrenmesi (QKML) yöntemi, Hilbert uzayında yapılan hesaplamalar ile klasik çekirdek yöntemlerine oldukça benzerdir [4]. Bu benzer durumdan yararlanarak, doğrusal olarak ayrılabilen veri kümeleri üzerinde sınıflandırma yapılabilir.

Veri kümesinden belli öznelikler seçilerek kuantum çekirdek tabanlı DVM modeli eğitilir. Bu modelde ilk olarak, eğitim verileri kuantum duruma kodlanır, kodlanan kuantum durumlarının üzerinden çekirdeğin gözlemlenebilir ölçümleri hesaplanır. Daha sonra kuantum devrenin ölçümleri klasik bilgisayarlarda bit olarak işlenir ve tüm eğitim veri kümesi için kuantum çekirdek hesaplanır. Bu kuantum çekirdek bir gram matristir. Bu gram matris kullanılarak sınıflandırılma yapılır.

3.1.1 Veri Kümesi

Tezde oluşturulan kuantum çekirdek ve klasik çekirdek tabanlı sınıflandırma modellerinde, Max A. Little vd. [21] hazırlamış olduğu Parkinson Hastalığı Veri

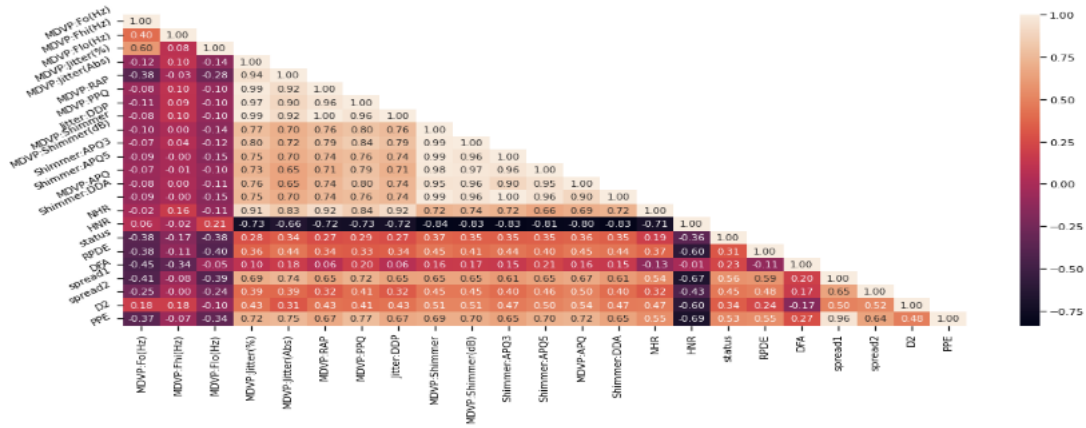
Kümesi kullanılmıştır. Bu veri kümesi, 23'ü Parkinson hastalık teşhisi konulmuş olan, 31 kadın ve erkek denekten alınan 195 sesli harf fonasyon kayıtlarından oluşmaktadır. Deneklerin yaşları 46 ila 85 arasındadır (ortalama 65.8, standart sapma 9.8) ve teşhisin konulmasından itibaren geçen süre ise 0 ila 28 yıldır. Her bir denekten, 1 ila 36 saniye arasında değişen, ortalama altı fonasyon kaydedilmiştir.

Veri kümesinin 195 sinyal kaydı için kullanılan öznitelik listesi Tablo 3.1'de verilmiştir:

Tablo 3.1 Parkinson Hastalığı Veri Kümesinin Öznitelikleri

No	Özellik	Açıklama
1	MDVP:Fo(Hz)	Ortalama temel ses frekansı
2	MDVP:Fhi(Hz)	Maksimum temel ses frekansı
3	MDVP:Flo(Hz)	Minimum temel ses frekansı
4	MDVP:Jitter(%)	Kay Pentax MDVP titreşimi
5	MDVP:Jitter(Abs)	Kay Pentax MDVP mikrosaniyede titreşimi
6	MDVP:RAP	Kay Pentax MDVP Bağıl Genlik Pertürbasyonu
7	MDVP:PPQ	Kay Pentax MDVP Beş Noktalı Pertürbasyon Katsayısı
8	Jitter:DDP	Döngüler arasındaki ortalama mutlak fark
9	MDVP:Shimmer	Kay Pentax MDVP yerel ışıltı
10	MDVP:Shimmer(dB)	Kay Pentax MDVP desibelde yerel ışıltı
11	Shimmer:APQ3	Üç Noktalı Genlik Pertürbasyon Katsayısı
12	Shimmer:APQ5	Beş Noktalı Genlik Pertürbasyon Katsayısı
13	MDVP:APQ	Kay Pentax MDVP 11 Noktalı Genlik Pertürbasyon Katsayısı
14	Shimmer:DDA	Ardışık genlikleri arasındaki ardışık ortalama mutlak farkı
15	NHR	Gürültü-Harmonik Oranı
16	HNR	Harmonik-Gürültü Oranı
17	RPDE	Tekrarlı Periyod Yoğunluk Entropisi
18	DFA	Eğilimsiz Dalgalanma Analizi
19	spread1	Ana frekans değişimi
20	spread2	Ana frekans değişimi
21	D2	Korelasyon Boyutu
22	PPE	Adım Periyodu Entropisi
23	status	Teşhis

Sekil 3.1'de tüm veri kümesinin korelasyon matrisi verilmiştir. Bu tez çalışmasında kullanılan RPDE, DFA, spread1, spread2, D2 ve PPE özniteliklerinin diğer özniteliklerle arasındaki korelasyonun düşük olduğu görülmektedir.



Şekil 3.1 Parkinson Veri Kümesi Korelasyon matrisi

3.1.2 Simülasyonlar ve Kuantum Bilgisayarlar

Günlük hayatta henüz kuantum bilgisayarlar kullanamamakta da kuantum hesaplama simülasyonları ile kuantum devreler kurulabilmektedir. Bu çalışmada Pennylane [18] ve Qiskit [22] simülasyon kütüphaneleri kullanılmıştır.

- **Pennylane:** Açık kaynaklı, kuantum türevlenebilir programlama kavramı üzerine kurulu, platformlar arası bilgi işleyebilen Python kütüphanesidir. Kuantum ve hibrit kuantum makine öğrenimi için hesaplama optimizasyonu ve farklı platform yöntemleri için birleşik bir mimari sunmaktadır [architec]. Pennylane'nin en önemli avantajı, varyasyonel kuantum devreleri ile gradyan hesaplamaya olanak sağlamasıdır. Pennylane kuantum hesaplama ortamı kuantum hesaplama simülasyonlarına ve bilgisayarlarına erişimine olanak sağlar. Klasik hesaplamada çok sık kullanılan TensorFlow, Pytorch, Keras gibi birçok Python makine öğrenmesi kütüphaneleri aracılığıyla arayüz oluşturur.
- **Qiskit:** IBM tarafından geliştirilen bir kuantum hesaplama simülasyonu olan Python kütüphanesidir. Kuantum fiziğinin sunmuş olduğu teoremler ve ansatzları dalga, foton formunda simüle eder. Bu doğrultuda atom altı parçacıkların kuantum durumları gözlemlenir; makine öğrenmesi, optimizasyon, biyoloji, optik ve kimya gibi birçok alanda uygulamalar geliştirmesine olanak sağlar.

3.1.3 Ön İşleme Aşaması

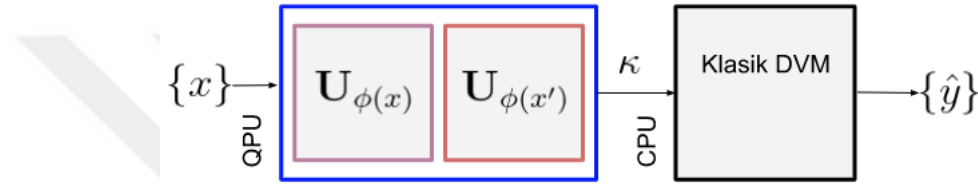
Veri kümesi StandardScaler metodu ile standardize edilir. Ortalama değeri 0, standart sapmanın ise 1 değerini aldığı, dağılımın normale yaklaştığı bir methodur. Veri kümesindeki ilgili sütun ortalaması çıkartılıp yine o sütun standart sapmasına

bölünerek bulunur. Böylelikle veri kümesindeki tüm girdiler -1 ile 1 arasında değer almış olur. Formülü: $z = \frac{(x-u)}{s}$

x girdinin değeri, u veri kümesindeki ilgili sütun ortalaması, s ise ilgili sütunun standart sapmasıdır.

3.1.4 Kuantum Çekirdek Temelli Destek Vektör Makinesi Modeli

Kuantum çekirdek temelli destek vektör makinaları modeli temelde iki bölümden oluşur (Şekil 3.2). QPU kısmında kuantum devre oluşturulur ve hesaplamalar yapılır, buradan elde edilen çekirdek sonuçları ile klasik makinede(CPU) tahminler(\hat{y}) elde edilir.



Şekil 3.2 Kuantum devre ve klasik DVM

- (i) Eğitim ve test kuantum çekirdek matrisleri oluşturulur. Eğitim ve test veri kümelerinin her ikili girdisi için denklem (3.1) kullanılarak çekirdek fonksiyonları oluşturulur.

$$\kappa(x, x') = |\langle \phi(x') | \phi(x) \rangle|^2 \quad (3.1)$$

- (ii) Klasik destek vektör makinesi sınıflandırılmasında eğitim ve test veri kümelerinin kuantum çekirdek matrisleri kullanılır.

Kuantum devresi(Şekil 3.2'deki QPU kısmı), kuantum durumların Hilbert uzayında kodlandığı kuantum çekirdeklerdir. Bu devre beş adımda oluşturulur.

1. Adım: Veri kümesinden bazı öznelikler seçilir ve öznelik sayısı kadar kubit kullanılır. Bu kubitlere veriler bağlanır.
2. Adım: Veri bağlanmış kubitlere, U üniter işlemci uygulanarak veriye bağlı kubitlerin kuantum durumları kodlanır.
3. Adım: Satır bazlı girdinin kuantum durum vektörü ile diğer satır bazlı girdilerinin kuantum durum vektörleri ile işleme girerek kuantum çekirdek κ elde edilir.

4. Adım: Gözlemlenebilir sonuçlar için \mathcal{M} Hermisyen matrisi uygulanarak ölçüm yapılır.
5. Adım: Ölçüm sonuçları klasik makine bitine çevrilir.



Kuantum sistemler üzerinde çalışırken kullanılan kubit sayısı veri kümesinin öznitelik sayısı kadar olmalıdır. Günümüz kuantum teknolojisinde, kuantum sistemlerin maliyetli ve hassas yapıda olmasından dolayı belli miktarda kubit kullanılabilir [17]. İki öznitelik arasındaki korelasyon katsayısının +1'e yakın değer olması öznitelikler arasında doğrusal bir ilişki olduğunu göstermektedir. Yani özniteliklerden birinin ölçüm değerleri diğer öznitelik ölçüm değerleri kullanılarak hesaplanabilmektedir. Birbirleriyle yüksek korelasyona sahip özniteliklerden birinin veri kümesi içerisinde silinmesi makina öğrenmesinde kullanılan en temel öznitelik indirgeme yöntemlerinden biridir. Şekil 3.1'deki korelasyon matrisi incelendiğinde öznitelikler arasındaki korelasyon katsayılarının yüksek olduğu görülmektedir. Yüksek korelasyona sahip öznitelikler hariç diğer öznitelikler kullanılarak farklı kombinasyonlarda 6 öznitelikten oluşan veri alt kümeleri oluşturulmuş ve 6 kubitlik kuantum devreler kullanılarak sonuçlar elde edilmiştir. En iyi başarı oranı RPDE, DFA, spread1, spread2, D2, PPE özniteliklerinden oluşan veri alt kümesinden elde edilmiştir. Klasik çekirdek modeller ile farklı kuantum simülatörlerde hesaplanan Kuantum Destek Vektör Makinesi(KDVM) modellerinin başarı oranları Tablo 4.1'de verilmiştir. En yüksek başarı oranı olan %93,87 Pennylane simülatörlerinde çalıştırılan KDVM modelinde elde edilmiştir. Sonuçlar, Parkinson veri kümesinin %75'i eğitim kümesi %25'i test kümesi olarak rastgele örneklemeyle ayrılıp, bu işlem 10 kez tekrarlanarak başarı oranlarının ortalaması alınarak hesaplanmıştır.

Tablo 4.1 Model Performans Sonuçları

Simülatör	Çekirdek Tipi	Başarı Oranı(%)	Doğru Pozitif (%)	Doğru Negatif (%)
Pennylane	Rotasyon	93,87	100	72,72
Qiskit	ZZFeatureMap	77,55	100	0
Klasik DVM	RBF	89	100	63,63
Klasik DVM	POLY	89	100	63,63

Klasik DVM için scikit-learn [23] kütüphanesinin SVM modeli RBF ve POLY

çekirdekleri varsayılan değerleri ile modeller oluşturulmuştur.

Pennylane ve Qiskit simülatörleri klasik sistemlerde üretilen verinin kuantum duruma getirilmesi için farklı kodlama yöntemleri kullanmaktadır. Pennylane simülatöründe oluşturulan modelde rotasyon kodlaması, Qiskit simülatörü kullanılarak oluşturulan modelde ise genlik temelli ZZFeatureMap kodlaması kullanılmıştır.

Pennylane ve Qiskit simülatörlerinde çalıştırılan KDVM ve klasik DVM modellerinin karışıklık matrisi Şekil 4.1’de verilmiştir. Karışık matrisleri incelendiğinde Qiskit simülatörlerinde çalıştırılan KDVM modelinin sağlıklı(normal) insan verilerini başarılı tahminleyemediği görülmektedir. Bu durum verinin kuantum durumuna dönüştürülmesi sırasında kullanılan veri kodlama çeşidinin modelin performansını nasıl etkilediğini göstermektedir.

		Gerçek Değerler	
		Parkinson	Normal
Tahmini Değerler	Parkinson	TP 38	FP 3
	Normal	FN 0	TN 8

Pennylane KDVM

		Gerçek Değerler	
		Parkinson	Normal
Tahmini Değerler	Parkinson	TP 38	FP 11
	Normal	FN 0	TN 0

Qiskit KDVM

		Gerçek Değerler	
		Parkinson	Normal
Tahmini Değerler	Parkinson	TP 38	FP 4
	Normal	FN 0	TN 7

DVM(RBF)

		Gerçek Değerler	
		Parkinson	Normal
Tahmini Değerler	Parkinson	TP 38	FP 4
	Normal	FN 0	TN 7

DVM(Poly)

Şekil 4.1 Karışıklık Matrisleri

Karşılaştırma yapmak amacıyla literatürde Parkinson veri kümesiyle klasik DVM algoritması kullanılarak yapılan [21] çalışmasındaki en yüksek performansı veren öznelikler ile Pennylane simülatöründe KDVM modeli oluşturulup, model başarı oranları Tablo 4.2’de verilmiştir. [21]’in sonuçlarıyla karşılaştırıldığında HNR, RPDE, DFA, PPE özneliklerini kullanan KDVM modeli daha düşük başarı oranı gösterirken RPDE, DFA, PPE öznelikleri için KDVM modeli klasik DVM’ye göre daha yüksek başarı oranı göstermiştir. En yüksek başarı oranı Tablo 4.1’de görüldüğü gibi 93,87 ile RPDE, DFA, spread1, spread2, D2, PPE özneliklerini kullanan KDVM modelinden elde edilmiştir.

Tablo 4.2 Klasik DVM ile KDVM Sonularının Karşılařtırılması

Öznitelikler	Klasik DVM [21]'in Başarı Oranı(%)	KDVM Başarı Oranı(%)
RPDE, DFA, PPE	89,5	91,83
HNR, RPDE, DFA, PPE	91,4	89,79



5 SONUÇ VE ÖNERİLER

Kuantum bilgisayarlarda, klasik bilgisayarlara göre çok büyük hesaplamaları ve büyük veri analizi gerektiren işlemleri çok kısa sürede yapılabildiği bilinmektedir. Pennylane'nin kuantum simülatörü klasik bilgisayarda çalıştırılmış olsa da kuantum modellerin tahminlemede başarılı olabileceğini göstermektedir. Qiskit ile olan çalışmada, klasik verinin kuantum duruma getirilmesi için kullanılan kodlamanın modelin eğitiminde önemli bir role sahip olduğu görülmektedir. Bu tez çalışmasında kuantum çekirdeklerin en az klasik çekirdekler kadar başarılı olabildiği görülmektedir. Gelecekte kuantum simülatörlerin daha da gelişmesi ile daha hızlı ve başarılı sonuçlar elde edebilmek umut vericidir.

- [1] L. Gyongyosi, S. Imre, “A survey on quantum computing technology,” *Computer Science Review*, vol. 31, pp. 51–71, 2019.
- [2] Havlíček, Vojtěch and Córcoles, Antonio D and Temme, Kristan and Harrow, Aram W and Kandala, Abhinav and Chow, Jerry M and Gambetta, Jay M, “Supervised learning with quantum-enhanced feature spaces,” *Nature*, vol. 567, no. 7747, pp. 209–212, 2019.
- [3] M. Schuld, N. Killoran, “Quantum machine learning in feature hilbert spaces,” *Physical review letters*, vol. 122, no. 4, p. 040 504, 2019.
- [4] M. Schuld, “Supervised quantum machine learning models are kernel methods,” *arXiv preprint arXiv:2101.11020*, 2021.
- [5] S. Lloyd, M. Schuld, A. Ijaz, J. Izaac, N. Killoran, “Quantum embeddings for machine learning,” *arXiv preprint arXiv:2001.03622*, 2020.
- [6] R. Shaydulin, S. M. Wild, “Importance of kernel bandwidth in quantum machine learning,” *arXiv preprint arXiv:2111.05451*, 2021.
- [7] F. Phillipson, “Quantum machine learning: Benefits and practical examples.,” in *QANSWER*, 2020, pp. 51–56.
- [8] E. ŞUHUBİ, *Fonksiyonel Analiz*. ITU Vakfi Yayinlari, 2001.
- [9] B. Scholkopf, A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2018, p. 36.
- [10] S. Metlek, K. Kayaalp, “Makine öğrenmesinde, teoriden örnek matlab uygulamalarına kadar destek vektör makineleri,” *İksad Yayınevi*, 2020.
- [11] A. Çevik, *Kuantum Hesaplama Kuramı*. ODTÜ Geliştirme Vakfi Yayıncılık ve İletişim A. Ş., 2021.
- [12] T. G. Wong, *Introduction to Classical and Quantum Computing*. Rooted Grove, 2022.
- [13] M. Schuld, I. Sinayskiy, F. Petruccione, “An introduction to quantum machine learning,” *Contemporary Physics*, vol. 56, no. 2, pp. 172–185, 2015.
- [14] M. Wilson, R. Stromswold, F. Wudarski, S. Hadfield, N. M. Tubman, E. G. Rieffel, “Optimizing quantum heuristics with meta-learning,” *Quantum Machine Intelligence*, vol. 3, no. 1, pp. 1–14, 2021.
- [15] M. Schuld, F. Petruccione, *Supervised learning with quantum computers*. Springer, 2018, vol. 17.
- [16] E. Aïmeur, G. Brassard, S. Gambs, “Machine learning in a quantum world,” in *Conference of the Canadian Society for Computational Studies of Intelligence*, Springer, 2006, pp. 431–442.

- [17] A. Cho, “Ibm promises 1000-qubit quantum computer—a milestone—by 2023,” *Science*, vol. 15, 2020.
- [18] V. Bergholm *et al.*, “PennyLane: Automatic differentiation of hybrid quantum-classical computations,” *arXiv preprint arXiv:1811.04968*, 2018.
- [19] R. Chatterjee, T. Yu, “Generalized coherent states, reproducing kernels, and quantum support vector machines,” *arXiv preprint arXiv:1612.03713*, 2016.
- [20] E. R. Dorsey *et al.*, “Global, regional, and national burden of parkinson’s disease, 1990–2016: A systematic analysis for the global burden of disease study 2016,” *The Lancet Neurology*, vol. 17, no. 11, pp. 939–953, 2018.
- [21] M. Little, P. McSharry, E. Hunter, J. Spielman, L. Ramig, “Suitability of dysphonia measurements for telemonitoring of parkinson’s disease,” *Nature Precedings*, pp. 1–1, 2008.
- [22] M. S. ANIS *et al.*, *Qiskit: An open-source framework for quantum computing*, 2021. DOI: 10.5281/zenodo.2573505.
- [23] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

TEZDEN ÜRETİLMİŞ YAYINLAR

1. G. SABAH ve N. GÜLER BAYAZIT, "Diagnosis of the Parkinson Disease by Using SVM Based on KQML", ICOMAA 2022 5th International ONLINE Conference on Mathematical Advances and Applications May, 11-14, 2022 Yıldız Technical University Istanbul, Turkey

