DOKUZ EYLÜL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İLAÇ KEŞFİ VE GELİŞTİRMEDE YAPAY ZEKA KULLANIMI

Emre ALTAY

Şubat, 2026
İZMİR

İLAÇ KEŞFİ VE GELİŞTİRMEDE YAPAY ZEKA KULLANIMI

Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü
Yüksek Lisans Tezi
İstatistik Anabilim Dalı, Veri Bilimi

Emre ALTAY

Şubat, 2026
İZMİR

YÜKSEK LİSANS TEZİ SINAV SONUÇ FORMU

EMRE ALTAY, tarafından **DOÇ. DR. İDIL YAVUZ** yönetiminde hazırlanan "**İLAÇ KEŞFİ VE GELİŞTİRMEDE YAPAY ZEKA KULLANIMI**" başlıklı tez tarafımızdan okunmuş, kapsamı ve niteliği açısından Yüksek Lisans tezi olarak kabul edilmiştir.

.....

Doç. Dr. İdil YAVUZ

Danışman

.....	.....
Doç. Dr. Özgül Vupa ÇİLENGİROĞLU	Dr. Öğr. Üyesi Merve SAYLAM
Jüri Üyesi	Jüri Üyesi

Prof.Dr. Abdullah SEÇGİN
Müdür
Fen Bilimleri Enstitüsü

TEŞEKKÜR

Bu çalışmadaki değerli katkıları için Doç. Dr. İdil Yavuz'a teşekkürlerimi sunuyorum.

Emre ALTAY

İLAÇ KEŞFİ VE GELİŞTİRMEDE YAPAY ZEKA KULLANIMI

ÖZ

Bu tezde, ilaç moleküllerinin hedef proteinlere bağlanmasını, daha kısa sürede ve daha düşük maliyetle tahmin etmeye yönelik bir yaklaşım ele alınmaktadır. İlaç geliştirme süreçlerinde aday moleküllerin hedef proteinlerle kurduğu bağlanma afinitesini öngörmek, maliyet ve hız açısından önemlidir. Bu çalışmanın odağında, sadece SMILES ve FASTA dizilerini girdi kabul eden Transformer tabanlı bir mimari yer almaktadır. Kurgulanan yapıda protein ve ligand dizileri, öncelikle birbirinden bağımsız kodlayıcı bloklarda işlenir. Takip eden aşamada ise iki yönlü çapraz dikkat katmanları devreye girer. Temsiller burada karşılıklı hizalanır ve etkileşim sinyali belirginleşir. Doğrusal olmayan ilişkilerin esnek bir biçimde öğrenilmesi esastır. Bu esnekliği sağlamak adına, modelin ileri beslemeli ağında standart yapılar yerine GeGLU tabanlı kapılı mekanizmalar kullanılmıştır.

Bahsedilen yapılandırma sonucunda ortaya çıkan TransformGeGLU modeli, sadece SMILES dizileri ve FASTA sekansları ile eğitilmiş olmasına karşın, DAVIS veri setinde MSE açısından 0.2359, Uyum İndeksi (Concordance Index, CI) açısından 0.8786 ve Kesinlik–Geri Çağırım Eğrisi Altındaki Alan (AUPR) içinse 0.7243 seviyesine ulaşmıştır; KIBA veri kümesinde ise sırasıyla MSE=0.1774, CI=0.8782 ve AUPR=0.7966 değerlerini yakalamıştır. Çok-modlu yaklaşımların sıklıkla sunduğu ilave iki ve üç boyutlu yapısal bilginin avantajına karşın, sınırlı ve minimal girdiyle dahi afinite tahmini açısından rekabet gücü yüksek performans yakalanabildiği bu çalışma kapsamında net biçimde ortaya konmuştur.

Anahtar Kelimeler: İlaç–hedef afinitesi (DTA) tahmini, Transformatör mimarisi, Derin öğrenme, İlaç keşfi, Çapraz dikkat

İLAÇ KEŞFİ VE GELİŞTİRMEDE YAPAY ZEKA KULLANIMI

ABSTRACT

This thesis presents an approach for predicting the binding of drug molecules to target proteins in a shorter time and at a lower cost. Predicting the binding affinity of candidate molecules to target proteins is important in drug development processes in terms of cost and speed. This study focuses on a Transformer-based architecture that accepts only SMILES and FASTA sequences as input. In the designed structure, protein and ligand sequences are first processed in separate encoder blocks. In the following stage, bidirectional cross-attention layers are activated. Here, the representations are aligned with each other, and the interaction signal becomes apparent. Flexible learning of nonlinear relationships is essential. To provide this flexibility, GeGLU-based gated mechanisms are used in the feedforward network of the model instead of standard structures.

The TransformGeGLU model resulting from the aforementioned configuration, although trained only on SMILES and FASTA sequences, achieved an MSE of 0.2359, a Concordance Index (CI) of 0.8786, and an area under the precision–recall curve (AUPR) of 0.7243 on the DAVIS dataset. On the KIBA dataset, it achieved an MSE of 0.1774, a CI of 0.8782, and an AUPR of 0.7966. Despite the advantage of the additional two- and three-dimensional structural information often offered by multimodal approaches, this study clearly demonstrates that competitive performance in terms of affinity estimation can be achieved even with limited and minimal input.

Keywords: Drug–target affinity (DTA) prediction, Transformer architecture, Deep learning, Drug discovery, Cross-attention

İÇİNDEKİLER

Sayfa

YÜKSEK LİSANS TEZİ SINAV SONUÇ FORMU ii
TEŞEKKÜR iii
ÖZ iv
ABSTRACT v
TABLOLAR LİSTESİ ix

BÖLÜM BİR – GİRİŞ VE MOTİVASYON 1

1.1 İlaç Keşfi ve Hedef Protein Etkileşimlerinin Önemi 1
1.2 Geleneksel İlaç Geliştirme Yaklaşımları 2
1.2.1 Hesaplamalı Kimya Yaklaşımları 3
1.2.2 Geleneksel Makine Öğrenmesi Yaklaşımları 4
1.3 İlaç Geliştirmede Derin Öğrenme ve Yapay Zeka Tabanlı Yaklaşımlar 4

BÖLÜM İKİ – MATERYAL VE YÖNTEM 8

2.1 Veri Yapısı ve Temsili 8
2.1.1 SMILES Notasyonu ile Kimyasal Yapı Temsili 8
2.1.2 Protein Amino Asit Dizisi 8
2.1.3 Bağlanma Afinitesinin pKd ile Ölçümü 8
2.2 Tokenizasyon ve Tokenizasyon Yöntemleri 10
2.3 Dikkat Mekanizmaları ve Model Mimarisi 12
2.3.1 Self-Attention (Öz-Dikkat) Mekanizması 12
2.3.2 Cross-Attention (Çapraz-Dikkat) Mekanizması 16
2.3.3 Genel Model Akışı 18
2.4 Yapay Sinir Ağları 19
2.4.1 Aktivasyon Fonksiyonları 20
2.4.1.1 ReLU (Rectified Linear Unit) 20
2.4.1.2 GELU (Gaussian Error Linear Unit) 20

2.4.1.3 GeGLU (Gated GELU) 21
2.4.2 Gömme Katmanları (Embedding Layers) 21
2.5 Hiperparametre Optimizasyonu 22
2.5.1 Optuna Kütüphanesi 22
2.5.2 Optuna Kütüphanesinin Kullanımı 23
2.5.3 En İyi Konfigürasyon ile Yeniden Eğitim ve Test 26
2.6 Verisetinin Ayrılması 27

BÖLÜM ÜÇ – UYGULAMA 30

3.1 Verisetlerinin Açıklanması 30
3.1.1 DAVIS Veriseti 30
3.1.2 KIBA Veriseti 31
3.2 Metriklerin Açıklanması 33
3.2.1 Ortalama Kare Hata (MSE) 34
3.2.2 Kök Ortalama Kare Hata (RMSE) 34
3.2.3 Uyum İndeksi (Concordance Index, CI) 35
3.2.4 Ortalama Kesinlik–Geri Çağırım Eğrisi Altındaki Alan (AUPR) 35
3.3 Ablasyon Çalışmaları 37
3.3.1 Deney Tasarımı 37
3.3.2 Veri Kümesi ve Eğitim Kurulumu 39
3.3.3 Aşama 1: 10 Epoch Geniş Tarama Sonuçları 40
3.3.3.1 CI'ya Göre En İyi Mimari Konfigürasyonlar (10 Epoch) 40
3.3.3.2 MSE'ye Göre En İyi Mimari Konfigürasyonlar (10 Epoch) 41
3.3.4 Aşama 2: 40 Epoch İnce Eleme ve Nihai Mimari Seçimi 42
3.3.4.1 40 Epoch Sonrası CI Karşılaştırması 43
3.3.4.2 40 Epoch Sonrası MSE Karşılaştırması 43
3.3.4.3 Nihai Mimari 44
3.3.5 Genel Değerlendirme 45
3.4 Sonuçların Literatürle Karşılaştırılması 46
3.4.1 Yalnız SMILES ve FASTA Kullanan Sekans-Temelli Modellerle Karşılaştırma 47

3.4.2 Ek Yapısal / Çok-Modalli Bilgi Kullanan Modellerle Göreli Konum .. 49
3.5 Sonuçların Değerlendirilmesi ve Son Modele Karar Verilmesi 51
3.5.1 Ablasyon Çalışmalarının Özet Değerlendirmesi 51
3.5.2 40 Epoch Karşılaştırmalı Sonuçlar ve Model Seçimi 53
3.5.3 Nihai Modelin DAVIS ve KIBA Üzerindeki Performansı 54

BÖLÜM DÖRT – TARTIŞMA VE SONUÇ 56

4.1 Literatürle Karşılaştırma ve Çalışmanın Katkıları 57
4.1.1 Girdi Türüne Göre Model Sınıfları 57
4.1.2 Sekans-temelli ve Çok-Modalli Yöntemlerle Karşılaştırma 58
4.2 Sınırlılıklar ve Gelecek Çalışmalar 60
4.3 Genel Sonuç 61

KAYNAKLAR 62

EKLER 73

Ek 1 Model Kodu (TransformGeGLU) 73

TABLOLAR LİSTESİ

Sayfa

Tablo 3.1 DAVIS veri setindeki bağlanma afinitesi (pKd) değerlerinin istatistiksel özeti 31
Tablo 3.2 KIBA verisetindeki biyoaktivite skorlarının istatistiksel özeti 33
Tablo 3.3 DAVIS test kümesi üzerinde CI'ya göre en iyi beş mimari (10 epoch). 41
Tablo 3.4 DAVIS test kümesi üzerinde MSE'ye göre en iyi beş mimari (10 epoch). 41
Tablo 3.5 40 epoch eğitim sonrası CI bakımından öne çıkan mimariler. 43
Tablo 3.6 40 epoch eğitim sonrası MSE bakımından öne çıkan (Aşama 1'den seçilmiş) mimariler. 44
Tablo 3.7 Yalnız SMILES ve FASTA kullanan sekans-temelli derin öğrenme modellerinin DAVIS ve KIBA test kümeleri üzerindeki performanslarının önerilen TransformGeGLU modeli ile karşılaştırılması. 48
Tablo 3.8 Ek yapısal/çok-modalli bilgi kullanan seçili DTA modelleri ile önerilen TransformGeGLU mimarisinin DAVIS ve KIBA test kümeleri üzerindeki karşılaştırması. GraphDTA, GDilatedDTA, AttentionMGT-DTA ve MEGDTA, ilaç ve/veya protein için ek graf/3B bilgi kullanmaktadır. 50
Tablo 3.9 Aşama 2'de 40 epoch yeniden eğitim sonrası DAVIS test kümesi performansları. 53
Tablo 3.10 Nihai TransformGeGLU mimarisinin DAVIS ve KIBA test kümelerindeki performansı. 55

BÖLÜM BİR
GİRİŞ VE MOTİVASYON

1.1 İlaç Keşfi ve Hedef Protein Etkileşimlerinin Önemi

Modern tıbbın karşı karşıya kaldığı başlıca güçlüklerden biri, hastalıkları etkili biçimde tedavi edebilecek, güvenilir ve sağlam profil sunan ilaç adaylarının tasarlanmasıdır. İlacın terapötik başarısı, çoğunlukla hedeflenen proteinle arasındaki etkileşimin şiddetine (afiniteye) dayanır (Overington ve diğer., 2006, Santos ve diğer., 2017). Bu etkileşimler ise hücresel mekanizmanın kontrolünde temel düzenleyici olarak görev üstlenir. Dolayısıyla hastalıkları iyileştirme çabalarının merkezinde, söz konusu protein-ligand etkileşimlerinin doğasını çözümleme ihtiyacı yer alır. Bu kavrayış tedavinin temelini oluşturur.

DiMasi ve çalışma arkadaşlarının maliyet üzerine gerçekleştirdikleri incelemeler, tek bir ilaç adayının pazara sunulabilmesi için yaklaşık 10 ila 15 yıl süren, mali boyutu ise 2.6 milyar dolara varan uzun ve karmaşık bir süreç gerektiğini ortaya koymaktadır (DiMasi ve diğer., 2016). Bu süreç içerisinde özellikle moleküllerin hedef proteinlere bağlanma kuvvetinin ölçüldüğü aşama, belirgin bir darboğaz oluşturur. Deneylerin sayısı dramatik biçimde artarken kaynak kullanımı ise kritik sınırlara dayanır. Mevcut deneysel yöntemlerle milyonlarca molekül içerisinden sınırlı sayıda aday belirlemek pratik ve sürdürülebilir değildir. Araştırmacılar, söz konusu darboğazı hafifletebilmek adına hesaplamalı yöntemlere yönelmeyi tercih ederler. Böylece, molekül–protein afinitesinin daha geniş molekül uzaylarında ve erken aşamalarda tahmin edilmesi mümkün olabilir. Geliştirilen hesaplamalı modellerin sanal tarama süreçlerine dahil edilmesinin ilaç keşfini hızlandırabileceği öne sürülmektedir (Chen ve diğer., 2020).

İlaçların proteinlerle etkileşimi, tek bir etki türüne indirgenemeyecek kadar karmaşık bir doğaya sahiptir. Hidrojen bağlarıyla hidrofobik etkileşimlerin yanı sıra, van der Waals kuvvetleri, elektrostatik etkenler ve hatta π–π istiflenmeleri dahi birlikte işleyerek bağlanma afinitesini biçimlendirir (Bissantz ve diğer., 2010).

Molekülün üç boyutlu yapısında hafifçe gerçekleşen bir burulma bile serbest enerji profilini büyük oranda değiştirebilir; aynı durum, yan zincirlerin polaritesindeki ufak farklılıklar ve moleküler yüzeydeki yük dağılımındaki küçük oynamalar için de geçerlidir. Bu nedenle tahmin modelleri, karmaşık ve girintili bir enerji yüzeyinde yol bulmaya çalışır. Görünüşte parametreleri kolayca belirlenebilir modeller dahi bu girift enerji alanında genelleme yapma konusunda zorlanır ve tutarlı sonuçlar üretmekte güçlük çeker. Temsil kalitesi, böyle bir durumda azalır.

1.2 Geleneksel İlaç Geliştirme Yaklaşımları

İlaçların proteinlerle olan etkileşimlerini doğrudan ölçebilmek adına araştırmacılar, laboratuvarlarda farklı biyokimyasal ve biyofiziksel yöntemler kullanır. Bu yöntemler arasında yüzey plazmon rezonansı (SPR), izotermal titrasyon kalorimetrisi (ITC) ya da floresan polarizasyon gibi teknikler sıkça tercih edilir (Macarron ve diğer., 2011). Uygun koşullar altında söz konusu yöntemlerle bağlanma sabitleri oldukça hassas biçimde ölçülebilir. Ancak pratikte bahsedilen tekniklerin her biri, pahalı ekipmanlar, sürekli tüketilen sarf malzemeleri ve her molekül için ayrı ayrı düzenlenen deney düzenekleri gerektirir. Özellikle yüksek verimli tarama (HTS) senaryolarında yüz binleri bulan molekül setlerini analiz etmek gerektiğinde, bu süreçler laboratuvar açısından ciddi anlamda külfet oluşturur ve bu tarama işlemleri aylar sürer. Dahası bütçe yönetiminde de ayrıntılı planlama şarttır. Böyle büyük ölçekli deneylerin lojistik karmaşıklığı ve mali yükü, bilimsel çalışmanın sınırlarını zorlayacak noktaya ulaşabilir.

Protein-ligand yapılarını görmek için X-ışını kristalografisi ve kriyojenik elektron mikroskobu (cryo-EM) gibi yöntemler sıkça kullanılmaktadır (Renaud ve diğer., 2018). Bu yöntemlerle proteinlerin yapısı neredeyse atom seviyesinde görüntülenebilir. Bağlanma bölgesinin şeklini ortaya çıkarır ve hidrojen bağlarının nasıl oluştuğu gözlenebilir. Araştırmacılar bu görüntülerden yola çıkarak bağlanma süreci hakkında tahminler üretmekte ve yeni ilaç adayları bu bilgilerden faydalanılarak geliştirilmektedir. Fakat her protein bu tekniklere uygun değildir. Bazı

proteinleri kristalleştirmek zordur. Esnek proteinler ya da kararsız kompleksler kristal oluşturmamaktadır. Membran proteinleri ise daha zorlu olmaktadır. Cryo-EM numunelerini hazırlamak ise ayrı uzmanlık gerektirmekte bu nedenle, yapısal bilgiler faydalı olsa da ilaç keşfinde kısıtlı bir kullanıma sahip olmaktadır.

1.2.1 Hesaplamalı Kimya Yaklaşımları

Deneysel yöntemler genelde çok pahalı ve zaman alıcıdır. Bu nedenle kimya ve biyofizik alanındaki araştırmacılar, hesaplamalı yöntemlere yönelmiştir (Kitchen ve diğer., 2004, Karplus ve McCammon, 2002). Moleküler yerleştirme ve moleküler dinamik simülasyonları bunlardan birkaçıdır. Moleküler yerleştirme yöntemlerinde ligandın hedef proteine nasıl bağlanacağı bilgisayarla test edilir. Ligandın farklı konumları denenir ve bağlanma enerjileri hesaplanır. En iyi bağlanan moleküller sıralanır. AutoDock, GOLD ve Glide gibi yazılımlar bu görevleri üstlenir. Her yazılımın kendi hesaplama yöntemi vardır. Bu araçlar sayesinde çok sayıda molekül kısa sürede taranabilir ve bu yöntemler ilaç keşfinde sıkça kullanılırlar (Trott ve Olson, 2010, Jones ve diğer., 1997, Friesner ve diğer., 2004).

Docking yöntemleri hızlı sonuç verse de bazı şeyleri göz ardı eder. Örneğin, proteinin yapısını sabit kabul eder ve su ortamını gerçeğe göre daha basit düşünür (Huang ve Zou, 2010). Bu varsayımlar çoğu zaman işe yarasa da, proteinin şekil değiştirdiği durumlarda bizi yanıltabilir. Daha gerçekçi bir analiz için moleküler dinamik simülasyonu kullanmak gerekir. Bu yöntem, bağlanma sürecini tüm hareketliliğiyle gösterir (Durrant ve McCammon, 2011). Ancak FEP ve TI gibi hassas hesaplama yöntemler oldukça yavaştırlar. Tek bir ligand için bile günlerce beklemek gerekebilir (Wang ve diğer., 2015). Bu yüzden çok sayıda bileşiği hızlıca taramak için uygun değillerdir.

1.2.2 Geleneksel Makine Öğrenmesi Yaklaşımları

Makine öğrenmesi tabanlı yaklaşımlar, bilhassa Nicel Yapı–Aktivite İlişkisi (QSAR) geleneğinden türetilen yöntemler, ilaçların hedef proteinlerle nasıl etkileşime gireceğini tahmin etmede giderek daha sık kullanılmaktadır (Chen ve diğer., 2018). Bu bağlamda araştırmacılar, molekülleri ve sıklıkla hedef proteinleri sayısal tanımlayıcılar kullanarak betimler. Tahmin aşamasında ise Rastgele Orman, Destek Vektör Makineleri (SVM) veya Gradient Boosting gibi denetimli algoritmalar tercih edilir (Cao ve diğer., 2014, He ve diğer., 2017). ECFP parmak izi ve MACCS anahtarları gibi moleküler betimleyicilerle amino asit kompozisyonundan türetilen vektörler, literatürde sıkça başvurulan gösterimler arasındadır (Rogers ve Hahn, 2010). Elle seçilen bu tip betimleyiciler, moleküllerin üç boyutlu hareketliliğini ve bağlanma ortamına göre şekillenen etkileşim motiflerini ancak sınırlı düzeyde kapsar. Ayrıca, protein yüzeyinin karmaşık yapısını tam anlamıyla yakalayamaz. Eğitim veri kümesinde bulunmayan kimyasal alanlara veya alışılmadık protein sınıflarına gelindiğinde ise, bu modellerin tahmin gücü tutarsızlaşabilir. Araştırma topluluğunu daha kapsamlı gösterimlere ve derin öğrenme yöntemlerine yönelten temel sebep, tam da bu performans değişkenliğidir. Derin öğrenme algoritmaları, ham verilerden doğrudan özellik çıkarma yetisiyle söz konusu tahmin problemlerine özgün çözümler sunmaktadır.

1.3 İlaç Geliştirmede Derin Öğrenme ve Yapay Zeka Tabanlı Yaklaşımlar

Son yıllarda derin öğrenme birçok alanda önemli gelişmeler sağlamıştır (LeCun ve diğer., 2015). Özellikle bilgisayarlı görü, doğal dil işleme ve biyoenformatikte bu etkiler açıkça görülmektedir. Bu alanlarda başarı elde edilmesinin sebebi, derin öğrenmenin verilerden kendi başına anlamlı bilgiler çıkarabilmesidir. İlaç keşfinde de eski yöntemlerin eksik kaldığı yerleri derin öğrenme doldurabilmektedir (Vamathevan ve diğer., 2019, Chen ve diğer., 2018). CNN yöntemleri, molekül yapılarını grafik ya da 3 boyutlu biçimlerde inceleyip önemli bilgileri bulabilir (Wallach ve diğer., 2015, Jiménez ve diğer., 2018) ve graf sinir ağları (GNN) ise molekülün atomlarını ve

bağlarını daha doğal ve kolay anlaşılır şekilde gösterir (Gilmer ve diğer., 2017, Yang ve diğer., 2019). SMILES gibi molekül dizileri ya da protein dizilerini analiz etmek için de RNN ve LSTM modelleri tercih edilmektedir (Gómez-Bombarelli ve diğer., 2018, Öztürk ve diğer., 2018).

2017 yılında Transformer isimli yeni bir model geliştirilmiştir (Vaswani ve diğer., 2017). Bu model, doğal dil işlemede pek çok şeyi değiştirmiştir ve self-attention denen yöntem sayesinde, kelimeler arasındaki uzak bağlantıları daha kolay yakalayabilmektedir. BERT, GPT ve T5 gibi modeller bu yöntemi kullanarak birçok farklı konuda başarılı olmuşlardır (Brown ve diğer., 2020). Biyoenformatik alanında da ProtBERT, ESM ve AlphaFold2 gibi modeller geliştirilmiştir. Bu modeller proteinlerin dizilerini ve yapılarını anlamakta oldukça başarılıdırlar (Jumper ve diğer., 2021). Sekans temelli derin öğrenme modelleri de ilaç ve protein arasındaki ilişkiyi anlamak için tercih edilmektedir (Öztürk ve diğer., 2018).

Eğitilmiş modeller, saniyeler içinde çok sayıda molekülün bağlanma gücünü tahmin edebilir. Bu hız, milyonlarca bileşikten oluşan kimyasal arşivlerin taranmasını kolaylaştırmaktadır (Stokes ve diğer., 2020). Modeller ayrıca moleküller için gerekli olan özellikleri kendi kendine öğrenebilirler. Bu durum, daha önce görülmemiş yeni moleküller üzerinde bile iyi sonuçlar elde etmeyi kolaylaştırır (Gómez-Bombarelli ve diğer., 2018). Farklı tipteki verileri, örneğin dizileri, molekül yapılarını ve biyolojik omik verileri tek model içinde birleştirebilir. Böylece, sorunlara daha bütüncül çözümler üretilebilir (Zitnik ve diğer., 2019). Başka büyük veri kümelerinde eğitilen modeller, küçük veri kümeleri için de ayarlanıp kullanılabilir. Bu özellikle nadir görülen hastalıklar ve yeni hedef proteinler için avantajlıdır (Hu ve diğer., 2020). Modellerin içindeki dikkat mekanizmaları ve aktivasyon haritaları sayesinde hangi moleküler bölgelere odaklandığı anlaşılabilir. Bu, bağlanma süreçlerinin daha net görülmesini sağlar (Vig, 2019). Son olarak, modellerin kullanımı ile yapılması gereken deneylerin sayısı azalır. Böylece, Ar-Ge çalışmalarının maliyeti düşer ve kaynaklar daha verimli kullanılır. Başarısız olma ihtimali yüksek moleküller erkenden elenerek, boşa harcanan zaman ve para azalır.

Derin öğrenme yöntemlerinin ilaç keşfindeki vaatleri sıkça gündeme gelse de, uygulamada bazı temel güçlüklerle karşı karşıya kalınır. Bunlardan biri, kaliteli ve doğruluğu deneylerle kanıtlanmış afinite verilerine ulaşmanın zorluğudur. Farklı deney ortamlarından elde edilen verilerin birbiriyle uyumsuzluğu ise model eğitimini ciddi anlamda karmaşık hale getirir (Tang ve diğer., 2014).

Bir başka belirsizlik ise Transformer modellerinin hangi varyasyonlarının en iyi sonucu vereceğidir. Çok sayıda farklı dikkat mekanizması, aktivasyon fonksiyonu ve normalizasyon tekniği vardır ve ilaç–hedef etkileşimi tahmininde hangisinin daha başarılı olacağı açık değildir. Bu konuda kapsamlı ve net sonuçlar henüz ortaya konulmamıştır.

Genelleme açısından bakıldığında modellerin, eğitim verisi dışında kalan proteinler veya yeni kimyasal bileşikler karşısında beklenen başarımı gösteremedikleri bilinir. Bu durum uygulamada karşımıza çıkan önemli bir durumdur ve çalışmaların gerçek hayattaki kullanılabilirliğini azaltır (Chen ve diğer., 2020).

Ayrıca, modellerin iç çalışma prensiplerinin net biçimde görülememesi araştırmacılar açısından güven sorununa yol açar. Kullanılan algoritmaların anlaşılabilir olması, ilaç geliştirme süreçlerinde çalışan uzmanların modellerin tahminlerine inanması için vazgeçilmezdir. Bu nedenle modellerin mekanistik açıdan daha açık ve anlaşılır olması gerekir.

Bu tezde, daha önceki çalışmalarda yeterince incelenmemiş konulara odaklanarak Transformer mimarisi içindeki çapraz dikkat mekanizmasıyla ilaçların hedef proteinlere bağlanma gücü tahmin edilmeye çalışılmıştır. Çalışmanın asıl fark yaratan kısmı ise detaylı ablasyon deneyleridir. Bu deneyler sayesinde Transformer içindeki farklı parçaların tahmin performansına ne derece etki ettiği açık biçimde ortaya konmuştur.

Spesifik olarak, bu tez aşağıdaki araştırma sorularını yanıtlamayı hedeflemektedir:

1. Konumsal kodlama: abs, RoPE, ALiBi gibi farklı pozisyonel encoding

yöntemlerinin afinite tahmini doğruluğuna etkisi.

2. Normalizasyon: LayerNorm vs. RMSNorm ve Pre-Norm vs. Post-Norm konfigürasyonlarının performansa etkisi.
3. Aktivasyon fonksiyonları: ReLU, GELU, SwiGLU, GeGLU gibi modern aktivasyon fonksiyonlarının regresyon görevindeki etkinliği.
4. Dikkat mekanizmaları: Çapraz dikkat, kapılı dikkat, gruplu sorgu dikkat (GQA) gibi farklı attention varyantlarının protein–ligand etkileşimlerini modellemedeki başarısı.

Bu tez kapsamında gerçekleştirilen sistematik inceleme ile hem teorik anlayışı derinleştirmek hem de pratik uygulayıcılar için mimari tasarım rehberi sunmak hedeflenmiştir. Toplam 256 model varyantının kapsamlı değerlendirmesi, gelecek çalışmalar için sağlam bir temel oluşturmaktadır.

BÖLÜM İKİ
MATERYAL VE YÖNTEM

2.1 Veri Yapısı ve Temsili

2.1.1 SMILES Notasyonu ile Kimyasal Yapı Temsili

Simplified Molecular Input Line Entry System ya da kısa adıyla SMILES, kimyasal bileşiklerin yapılarını tek boyutlu ASCII karakter dizileriyle gösteren özel bir notasyon sistemidir (Weininger, 1988, Öztürk ve diğer., 2018). Bu çalışmada kullanılan ilaç benzeri küçük moleküllerin (ligandların) yapısal özellikleri, SMILES formatı aracılığıyla kodlanmıştır. Söz konusu format; atom türlerini (örneğin karbon için `C`, azot için `N`, oksijen için `O` ya da klor için `Cl` ve brom için `Br`), kimyasal bağ türlerini (tek bağ `-`, çift bağ `=`, üçlü bağ ise `#` sembolüyle), halkasal yapıları (sayısal etiketler yardımıyla) ve stereokimyasal özellikleri (`@` ve `@@` işaretleri ile) açık ve yalın bir biçimde temsil etmektedir (Weininger, 1988, Öztürk ve diğer., 2018, Stokes ve diğer., 2020). Mesela etanol molekülü için kullanılan gösterim `CCO` iken, benzen halkası ise `c1ccccc1` şeklinde ifade bulur.

2.1.2 Protein Amino Asit Dizisi

Hedef proteinler, 20 standart amino asitten oluşan birincil yapı dizileri (primer sekans) ile temsil edilmiştir (Branden ve Tooze, 1999). Protein dizileri, her bir amino asidin tek harfli kodu (A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y) kullanılarak gösterilmiştir. Protein dizisi, üç boyutlu yapının belirlenmesinde ve ligand bağlanma bölgelerinin karakterizasyonunda kritik öneme sahiptir.

2.1.3 Bağlanma Afinitesinin pKd ile Ölçümü

İlaç-hedef protein etkileşimlerinin gücü, dissosiasyon sabiti (K_d) ve bunun negatif logaritmik formu olan pK_d değeri ile ölçülmüştür (Öztürk ve diğer., 2018).

Eşitlik (2.1)'de pK_d tanımı verilmiştir.

$$pK_d = -\log_{10}(K_d) \tag{2.1}$$

Burada K_d, molar birim cinsinden (M) protein–ligand kompleksinin denge durumundaki dissosiasyon sabitini ifade eder. Düşük K_d değeri güçlü bağlanmayı (yüksek afinite), yüksek K_d değeri ise zayıf bağlanmayı (düşük afinite) gösterir. pK_d transformasyonu, değerlerin daha anlamlı ve geniş bir ölçekte dağılmasını sağlar. Örneğin:

$K_d = 1$ nM (10^{-9} M) $\rightarrow$ p$K_d = 9.0$ (çok güçlü bağlanma)

$K_d = 1$ μM (10^{-6} M) $\rightarrow$ p$K_d = 6.0$ (orta derecede bağlanma)

$K_d = 1$ mM (10^{-3} M) $\rightarrow$ p$K_d = 3.0$ (zayıf bağlanma)

Bu çalışmada kullanılan Davis veri setinde, pK_d değerleri yaklaşık 5.0 ile 11.0 arasında değişmektedir ve biyolojik olarak anlamlı bir afinite spektrumunu kapsamaktadır (Davis ve diğer., 2011, Öztürk ve diğer., 2018, Deng ve diğer., 2022).

Protein–ligand bağlanma afinitesi, modelleme yaklaşımına bağlı olarak iki farklı şekilde ele alınabilir. Sürekli değişken yaklaşımı kullanan çalışmalarda pK_d değerleri sürekli (continuous) sayısal değişkenler olarak işlenmiştir. Model, regresyon problemi olarak tasarlanmış ve her bir protein–ligand çifti için gerçek pK_d değerini tahmin etmeyi amaçlamıştır. Bu yaklaşım, bağlanma afinitesinin nüanslarını korur ve farklı afinite düzeyleri arasındaki ince farkları modelleyebilir. Performans metrikleri olarak Ortalama Kare Hata (MSE), Kök Ortalama Kare Hata (RMSE) ve Konkordans İndeksi (CI) gibi regresyon metrikleri kullanılmıştır (Pahikkala ve diğer., 2015, He ve diğer., 2017, Öztürk ve diğer., 2018).

Kategorik değişken yaklaşımını benimseyen çalışmalar ise pK_d değerlerini belirli bir sınır üzerinden gruplamaktadır. Mesela p$K_d > 7.0$ değerinin üzerindeki moleküller güçlü bağlanan, altındakiler ise zayıf bağlanan olarak tanımlanabilir (Öztürk ve diğer., 2018, Deng ve diğer., 2022). Böylece problem sınıflama türüne

dönüşür. Bu yöntemde başarıyı değerlendirmek için AUPR, AUROC ya da F1 skoru gibi yöntemler tercih edilir (Yamanishi ve diğer., 2008, Öztürk ve diğer., 2018, Deng ve diğer., 2022). Bu sınıflama yaklaşımı, ilaç keşfinde moleküllerin aktif veya inaktif olup olmadığını belirlemek için oldukça kullanışlıdır. Ancak bağlanma gücünün hassas değerleri tam olarak korunamaz ve bazı bilgiler göz ardı edilir.

Bu tezde temel yaklaşım sürekli değişken regresyonu olmakla birlikte, modelin performansının kapsamlı değerlendirilmesi için, Davis veri seti üzerinde yaygın olarak kullanılan 7.0 pK_d eşik değeri esas alınarak AUPR metriği de ek olarak hesaplanmıştır (Öztürk ve diğer., 2018, Deng ve diğer., 2022).

2.2 Tokenizasyon ve Tokenizasyon Yöntemleri

Derin öğrenme modelleri, protein dizileri ve SMILES molekül yapılarını işleyebilmek için bu dizilerin sayısal hale getirilmesine ihtiyaç duyar (Öztürk ve diğer., 2018, Zhao ve diğer., 2022). Bu tezde proteinlerin ve moleküllerin farklı özellikleri düşünülerek iki ayrı tokenizasyon yöntemi kullanılmıştır.

Protein dizileri önce küçük parçalara bölünmüştür ve her amino asit için ayrı bir sembol belirlenmiştir. En sık görülen 20 amino aside ek olarak, az rastlanan veya değişikliğe uğramış amino asitler de sembollerle gösterilmiştir (Öztürk ve diğer., 2018, Zhao ve diğer., 2022). Dizilerde bulunan boşluklar çıkarılmış ve her dizi basit karakter listesine dönüştürülmüştür. Böylece proteinlerin doğal hali korunarak protein-afinite tahmininde kullanılan evrişim ve dikkat yöntemleriyle de iyi sonuçlar elde edilmiştir (Öztürk ve diğer., 2018, Zhao ve diğer., 2022).

SMILES (Simplified Molecular Input Line Entry System) dizilerinin parçalara ayrılması sürecinde düzenli ifadelerden yararlanılmıştır. Schwaller ve arkadaşlarının ortaya koyduğu kimyasal token esaslı ön-işleme stratejisi, burada da yol göstericidir (Schwaller ve diğer., 2019, Bjerrum ve diğer., 2021) Çok karakterli atom sembolleri ve köşeli parantezlerle çevrelenmiş atom ortamları, tek bir birim halinde ele alınmıştır. Hiyerarşik bir eşleştirme mantığı benimsenerek SMILES notasyonunun

taşıdığı kimyasal anlam bozulmamıştır. Atom merkezli bir çözünürlük hedeflenmiştir. Karakter düzeyinde tokenizasyon ise başka bir tablo çizerek bazı belirsizlikleri artırır ve kimyasal bağlam parçalanır (Li ve Fourches, 2021). Önerilen yaklaşım da bu sorunu bertaraf etmeye yönelik olarak tasarlanmıştır.

Tokenizasyon işlemi şu öncelik sırasına göre gerçekleştirilmiştir:

1. Köşeli parantez içi atomlar: [NH3+], [O-] gibi yüklü veya özel durumdaki atomlar
2. Halka kapatma sembolleri: %12, %13 gibi iki haneli halka referansları
3. İki harfli elementler: Br (brom), Cl (klor), Si (silisyum), Se (selenyum), vb.
4. Stereokimya gösterimleri: @ ve @@ sembolleri
5. Bağ tipleri: = (çift bağ), # (üçlü bağ)
6. Yapısal semboller: Parantezler (), dallanmalar
7. Rakamlar ve tek harfli elementler: 1–9 ve C, N, O gibi tek karakterli elementler
8. İşaretler: +, -, *, /, \ gibi diğer kimyasal semboller

Bu hiyerarşik eşleştirme yaklaşımı, literatürde karakter-düzeyindeki SMILES tokenizasyonuna yönelik eleştirilerde vurgulanan, “Cl” ve “Br” gibi çok karakterli atom sembollerinin tek bir kimyasal birim olarak yakalanmasını sağlar ve “C” ile “l” karakterlerinin ayrı tokenlar olarak ayrılmasını engeller (Li ve Fourches, 2021). Bu sayede “CCCCl” gibi bir SMILES dizisi [’C’,’C’,’C’,’Cl’] şeklinde doğru tokenize edilmekte, benzer şekilde “c1ccccc1Br” benzol halkası brom içeren yapı için [’c’,’1’,’c’,’c’,’c’,’c’,’c’,’1’,’Br’] tokenizasyonu elde edilmektedir.

`<pad>`: Farklı uzunluktaki sekansların eşit uzunluğa getirilmesi (padding)
`<unk>`: Kelime dağarcığında bulunmayan bilinmeyen tokenler
`<cls>`: Sınıflandırma token’ı (ileride kullanım için rezerve)
`<sep>`: Ayırıcı token (ileride kullanım için rezerve)

Bu özel tokenler, doğal dil işleme alanında Transformer tabanlı dil modellerinde kullanılan [CLS] ve [SEP] benzeri özel sembollerle aynı mantıkla tasarlanmıştır (Devlin ve diğer., 2019).

Kelime dağarcıkları, eğitim için kullanılan veri kümesindeki farklı ve tekrar etmeyen tüm tokenlerin toplanıp alfabetik olarak sıralanmasıyla oluşturulmuştur. Bununla beraber, kelime dağarcığının hemen başına, özel olarak tanımlanan tokenler sıfırdan başlayan bir indeksleme ile yerleştirilmiştir. Bu yöntem, SMILES tabanlı derin öğrenme modellerinde yaygın biçimde uygulanan sözlük odaklı kodlama ve verilerin tensöre dönüştürülmesi süreçleriyle de birebir örtüşmektedir (Bjerrum ve diğer., 2021). Protein ve SMILES dizileri için ayrı kelime dağarcıkları oluşturulmuş; her token, ilgili kelime dağarcığındaki indeks numarası ile sayısal forma dönüştürülmüştür.

Her bir token, kelime dağarcığındaki indeks numarası ile sayısal forma dönüştürülmüş, eğitim setinde bulunmayan tokenler <unk> token indeksi ile temsil edilmiştir. Sekanslar, aşama 1 için maksimum protein uzunluğu (1000) ve maksimum SMILES uzunluğu (100) parametreleri ile sınırlandırılmış, bu uzunluklardan kısa sekanslar <pad> token'ı (indeks: 0) ile doldurulmuştur. Aşama 2 için sekans uzunluğu her model için parametre optimizasyonu ile bulunmuştur.

2.3 Dikkat Mekanizmaları ve Model Mimarisi

2.3.1 Self-Attention (Öz-Dikkat) Mekanizması

Self-attention, bir dizideki her öğenin diğer öğelerle olan ilişkisini öğrenmesine olanak tanıyan bir yöntemdir. Geleneksel sinir ağları, dizileri işlerken sırayla gider ve belirli bir alana odaklanır. Self-attention ise tüm diziyi tek seferde görebilir. Bu sayede, dizinin başındaki bir öğe ile sonundaki öğe arasındaki ilişkileri bile kolayca yakalar (Vaswani ve diğer., 2017).

Protein dizileri açısından ele alındığında, bir amino asidin rolü, yalnızca kendisi ya

da yanındaki amino asitlerle değil, dizideki oldukça uzak konumlarda bulunan diğer amino asitlerle olan ilişkileriyle de belirlenebilir. Proteinlerin üç boyutlu yapıya katlanmasıyla birlikte, doğrusal dizilimde oldukça ayrı noktalarda yer alan amino asitler fiziksel anlamda yan yana gelip aktif bölgeleri meydana getirebilir. Self-attention mekanizması, bu uzak ilişkileri doğrudan yakalayarak modelleme yeteneğine sahiptir. Böylece büyük ölçekli protein dil modelleri, amino asit dizisinden hareketle, ikincil yapı oluşumu ve uzun mesafeli amino asit temasları gibi karmaşık biyolojik özellikleri başarılı biçimde tahmin edebilmiştir (Rao ve diğer., 2019, Elnaggar ve diğer., 2020, Rives ve diğer., 2021).

Self-attention mekanizması, üç temel bileşen üzerinden çalışır: Query (Sorgu), Key (Anahtar) ve Value (Değer). Transformer mimarisinde her token, bu üç rol için ayrı doğrusal projeksiyonlarla temsil edilir (Vaswani ve diğer., 2017).

Self-attention'ı bir kütüphanede bilgi aramaya benzetebiliriz. Query, "Hangi bilgiyi arıyorum?" sorusuna karşılık gelir ve her token, diğer tokenlardan ne tür bilgi almak istediğini belirten bir sorgu vektörü oluşturur. Key, "Ben ne tür bilgi sunuyorum?" sorusuna karşılık gelir ve her token, kendisinin ne tür bilgi içerdiğini tanımlayan bir anahtar vektörü oluşturur. Value ise "Gerçekte aktaracağım bilgi nedir?" sorusuna karşılık gelir ve her token, başkalarına iletilecek gerçek bilgiyi içeren bir değer vektörü oluşturur.

Her token için giriş temsili üç farklı öğrenilebilir ağırlık matrisi ile çarpılarak Query, Key ve Value vektörleri elde edilir. Protein dizisinde her amino asit, bu üç farklı rolle temsil edilir.

Benzerlik hesaplanması için ise bazı adımların takip edilmesi gerekir. Her Query vektörü, dizideki tüm Key vektörleri ile nokta çarpımı yapılır. Bu işlem, "sorgu sahibi token, hangi diğer tokenlara daha çok ilgi duymalı?" sorusunu yanıtlar. Yüksek nokta çarpımı, yüksek benzerlik ve dolayısıyla yüksek ilgiye işaret eder.

Eşitlik (2.2)–(2.4)'de Transformer mimarisinde kullanılan *ölçeklendirilmiş nokta çarpımı dikkat* (scaled dot-product attention) mekanizmasının skor, dikkat ağırlığı ve

çıktı hesaplamaları verilmiştir (Vaswani ve diğer., 2017).

$$\text{score}_{ij} = \frac{Q_i \cdot K_j}{\sqrt{d_k}} \tag{2.2}$$

$$\alpha_{ij} = \frac{\exp(\text{score}_{ij})}{\sum_{k=1}^{n} \exp(\text{score}_{ik})} \tag{2.3}$$

$$\text{output}_i = \sum_{j=1}^{n} \alpha_{ij} \, V_j \tag{2.4}$$

Burada Q, K ve V sırasıyla sorgu (query), anahtar (key) ve değer (value) temsillerini ifade eder. Q_i, i'inci pozisyona ait sorgu vektörünü; K_j ve V_j ise j'inci pozisyona ait anahtar ve değer vektörlerini gösterir. $\cdot$ işleci vektörler arası iç çarpımı (nokta çarpımı) belirtir. d_k, sorgu/anahtar vektörlerinin boyutudur ve $\sqrt{d_k}$ terimi skorları ölçeklendirmek için kullanılır. n, dizideki (sequence) toplam eleman/token sayısını; score_{ij}, i'inci sorgunun j'inci anahtarla benzerlik skorunu; α_{ij}, i'inci pozisyonun j'inci pozisyona atadığı dikkat ağırlığını (her i için $\sum_{j=1}^{n} \alpha_{ij} = 1$); output_i ise değer vektörlerinin dikkat ağırlıklarıyla alınmış ağırlıklı toplamından oluşan çıktı vektörünü ifade eder.

Tek bir attention yerine, modelde sekiz paralel attention başlığı (head) kullanılır. Böylece farklı temsil alt-uzaylarında farklı ilişkiler öğrenilebilir; her bir başlık Query/Key/Value uzayına farklı bir doğrusal projeksiyon uygular (Vaswani ve diğer., 2017). Önerilen modelde $d_{\text{model}} = 256$ ve $h = 8$ olduğundan her başlık 32 boyutta çalışır; başlık çıktıları birleştirilip (concat) tekrar 256 boyuta projekte edilir.

Self-attention katmanından sonra, her pozisyon için bağımsız bir ileri beslemeli ağ (FFN) bulunur. Bu çalışmada standart ReLU yerine GeGLU kullanılmıştır (Shazeer, 2020):

$$\text{GeGLU}(x) = (xW_{\text{value}}) \odot \text{GELU}(xW_{\text{gate}})$$

GLU tabanlı aktivasyonların, Transformer'ın FFN (Feed-Forward Network) katmanlarında ReLU (Rectified Linear Unit) ve GeLU (Gaussian Error Linear Unit) gibi klasik aktivasyonlar yerine kullanıldığında model kalitesini artırabildiği gösterilmiştir. Bu kapsamda özellikle GeGLU (Gated GELU) gibi kapılamalı (gated) varyantlar, daha ifade gücü yüksek bir dönüşüm sağlayarak performans artışına katkıda bulunabilir.(Shazeer, 2020). Gating (Kapılama) nedeniyle parametre artışını dengelemek için ara boyut, standart FFN'in yaklaşık $\frac{2}{3}$'ü seçilir.

Bir encoder bloğu sırasıyla:

1. Multi-Head Self-Attention (girdideki her token'ın, dizideki diğer token'lara "dikkat" vererek bağlamlı temsil üretmesi; çoklu başlık ile farklı ilişki türleri paralel yakalanır)
2. Artık Bağlantı (Residual) + LayerNorm (artık bağlantı: alt-katmanın çıktısına orijinal girdiyi ekleyerek bilgi kaybını azaltır ve gradyan akışını kolaylaştırır; LayerNorm: her token'ın özelliklerini normalize ederek eğitimi daha kararlı hale getirir)
3. FFN (GeGLU) (token bazlı iki katmanlı ileri beslemeli ağ; GeGLU: GLU kapılama mantığını GELU aktivasyonu ile birleştirerek daha ifade gücü yüksek dönüşüm sağlar)
4. Artık Bağlantı + LayerNorm (artık bağlantı: FFN dönüşümünü girişle toplayıp temsilin korunmasına yardım eder; LayerNorm: ölçek/dağılımı dengeler, optimizasyonu stabil tutar)

Eşitlik (2.5)'de Transformer encoder bloğunun iki ardışık alt-katmandan (çok başlı öz-dikkat ve konum-bazlı ileri beslemeli ağ) oluşan güncelleme adımları verilmiştir.

$$\begin{aligned} X^{(1)} &= \mathrm{LayerNorm}\Big(X + \mathrm{MHA}(X, X, X)\Big), \\ X^{(2)} &= \mathrm{LayerNorm}\Big(X^{(1)} + \mathrm{FFN}_{\mathrm{GeGLU}}\big(X^{(1)}\big)\Big). \end{aligned} \tag{2.5}$$

Burada $X \in \mathbb{R}^{n \times d_{\text{model}}}$ encoder bloğuna giren gizil gösterimleri (sırasıyla dizi uzunluğu n ve gizil boyut d_{model}) ifade eder. $X^{(1)}$ çok başlı öz-dikkat (self-attention) alt-katmanı sonrası elde edilen ara temsili, $X^{(2)}$ ise ileri beslemeli ağ (FFN) alt-katmanı sonrası encoder bloğu çıktısını gösterir. $\text{MHA}(Q, K, V)$ çok başlı dikkat (Multi-Head Attention) işlemini belirtir; öz-dikkatte sorgu, anahtar ve değer matrisleri aynı girdiden türetilir ve bu nedenle $(Q, K, V) = (X, X, X)$ alınır. $\text{LayerNorm}(\cdot)$ katman normalizasyonunu, toplama işlemi $(+)$ ise artık bağlantıyı (residual connection) ifade eder. $\text{FFN}_{\text{GeGLU}}(\cdot)$ her bir konumda (token'da) bağımsız uygulanan konum-bazlı ileri beslemeli ağı ve bu ağda kullanılan GeGLU (Gated GELU) aktivasyonunu temsil eder.

Bu yapı, Transformer encoder bloğunun klasik tasarımını takip etmekte (Vaswani ve diğer., 2017), ancak FFN bileşeninde GeGLU aktivasyonu tercih edilmektedir (Shazeer, 2020).

Değişken uzunluktaki dizilerin aynı uzunlukta işlenebilmesi için, gerçek girdinin bulunmadığı konumlar özel bir *padding* (dolgu) tokenı ile tamamlanır. Dikkat (attention) hesaplamasında bu yapay tokenların modele katkı vermesini engellemek amacıyla bir maskeleme uygulanır ve skor matrisinde padding'e karşılık gelen konumlara $-\infty$ (uygulamada çok büyük negatif bir sabit) atanır. Böylece softmax sonrasında bu konumların dikkat ağırlıkları pratikte sıfıra yaklaşır ve model yalnızca geçerli tokenlara odaklanır; bu yaklaşım Transformer tabanlı modellerde yaygın olarak kullanılmaktadır (Vaswani ve diğer., 2017).

2.3.2 Cross-Attention (Çapraz-Dikkat) Mekanizması

Self-attention her dizinin kendi iç ilişkilerini modeller; protein–molekül arası etkileşimi yakalamak için ise cross-attention kullanılır. İlaç-hedef etkileşimi literatüründe, hem ilaç hem de protein dizilerini Transformer tabanlı kodlayıcılarla temsil edip bu temsillerden etkileşim tahmini yapan modellerin başarılı sonuçlar verdiği gösterilmiştir (Chen ve diğer., 2020, Huang ve diğer., 2021). Önerilen

mimaride bu fikir bir adım öteye taşınarak, protein ve SMILES temsilleri cross-attention katmanları ile doğrudan birbirine bağlanmaktadır.

Self-attention için Attn(X, X, X) tarzı bir gösterim uygunken Cross-attention için Attn$(X_{\text{prot}}, X_{\text{smi}}, X_{\text{smi}})$ (ve tersi) kullanılabilir. Her iki durumda da ölçeklendirilmiş nokta çarpımı dikkat formu kullanılır (Vaswani ve diğer., 2017).

Hem Protein→SMILES hem de SMILES→Protein yönlerinde uygulama yapıldığında skorlar aşağıdaki gibi hesaplanır:

$$\text{score}_{ij}^{P\to S} = \frac{Q_i^{\text{prot}} \cdot K_j^{\text{smi}}}{\sqrt{d_k}}, \qquad \text{score}_{ij}^{S\to P} = \frac{Q_i^{\text{smi}} \cdot K_j^{\text{prot}}}{\sqrt{d_k}} \tag{2.6}$$

Bu tür çift yönlü cross-attention mekanizmalarının, ilaç ve protein temsillerini karşılıklı olarak hizalayarak etkileşim özelliklerini zenginleştirdiği, farklı DTI/CPI modellerinde de gösterilmiştir (Nguyen ve diğer., 2023, Zeng ve diğer., 2024).

Eşitlik (2.7) ve Eşitlik (2.8)'de iki dizilim/kip (ör. protein ve molekül) temsilinin çift yönlü çapraz dikkat (cross-attention) ile etkileştirilmesi ve artık bağlantı (residual) ile *LayerNorm* sonrası güncellenmiş temsillerin elde edilmesi gösterilmektedir.

$$\begin{aligned} P_{\text{cross}} &= \text{MHA}\big(Q{=}P,\ K{=}S,\ V{=}S,\ \text{mask} = M_S\big), \\ P' &= \text{LayerNorm}(P + \text{Dropout}(P_{\text{cross}})) \end{aligned} \tag{2.7}$$

$$\begin{aligned} S_{\text{cross}} &= \text{MHA}\big(Q{=}S,\ K{=}P,\ V{=}P,\ \text{mask} = M_P\big), \\ S' &= \text{LayerNorm}(S + \text{Dropout}(S_{\text{cross}})) \end{aligned} \tag{2.8}$$

Burada P ve S, sırasıyla iki girdiye ait (uzunluk $\times d_{\text{model}}$ boyutlu) gizil gösterimleri; MHA$(\cdot)$ çok başlı dikkat (Multi-Head Attention) operatörünü ifade eder. P_{cross}, P'nin sorgu (Q) olarak kullanılıp S üzerinden anahtar/değer (K, V) okunmasıyla elde edilen çapraz-dikkat çıktısıdır; benzer biçimde S_{cross} de S'nin P üzerinden okuduğu çıktıdır. M_S ve M_P, ilgili dizilerdeki doldurma (*padding*)

konumlarının dikkate alınmasını engelleyen dikkat maskeleridir. Dropout$(\cdot)$ düzenlileştirme amacıyla rastgele birim düşürmeyi, LayerNorm$(\cdot)$ ise artık bağlantı ile toplanan aktivasyonların katman normalizasyonunu gösterir; P' ve S' güncellenmiş temsillerdir.

Protein dizisinden SMILES dizisine geçişte SMILES dizisinin padding bölgeleri maskelenir, yani hesaba katılmaz. Aynı şekilde SMILES dizisinden proteine geçiş sırasında da proteinin padding bölgeleri göz ardı edilir. Bu işlem sayesinde sadece anlam taşıyan gerçek diziler birbiriyle ilişkilendirilir. Cross-attention kullanan DTI modellerinde benzer maskeleme yöntemleri sıkça kullanılır (Nguyen ve diğer., 2023, Zeng ve diğer., 2024).

2.3.3 Genel Model Akışı

Her bir token için hesaplanan gömme vektörü ile pozisyonel gömme değerleri birleştirilir ve böylece her token özelinde d_{model} uzunluğunda bir başlangıç temsiline ulaşılır. Transformer mimarilerinde sıkça tercih edilen içerik tabanlı ve konum odaklı gömme stratejisinin temel prensibi burada uygulanmaktadır (Vaswani ve diğer., 2017).

Protein ve SMILES dizileri, ayrı ayrı tasarlanmış ve iki bloktan oluşan encoder gruplarından geçirilir. Bu sayede diziler, Transformer temelli ilaç-hedef etkileşim modellerinde kullanılan saf self-attention katmanlarına benzer şekilde, kendi içlerinde yer alan bağlamsal ilişkileri öğrenebilir (Chen ve diğer., 2020, Huang ve diğer., 2021).

Protein ve SMILES dizilerinin birbirleriyle olan etkileşimini ortaya koymak adına, iki yönlü cross-attention mekanizması devreye alınır. Böylelikle, elde edilen yeni gösterimler olan P' ve S', bir taraftaki dizinin diğer taraftaki bağlamı net şekilde görmesini sağlar. Yani protein dizisindeki her bir token, karşıdaki SMILES dizisine dair bilgilere erişebilirken, SMILES dizisindeki tokenlar da proteinlerin sunduğu bağlamı görebilir duruma gelir (Nguyen ve diğer., 2023, Zeng ve diğer., 2024).

Sonraki adımda ise, yalnızca gerçek ve anlam taşıyan tokenlar değerlendirilir; padding amaçlı tokenlar hesaba katılmaz. Bu gerçek tokenların ortalaması alınarak, protein ve SMILES dizileri için sabit uzunlukta iki ayrı vektör oluşturulur. Transformer türü modeller genellikle sabit uzunluklu temsiller üretmek amacıyla ya [CLS] benzeri özel tokenlar (*[CLS]: dizinin başına eklenen ve tüm diziyi özetleyen sınıflandırma tokeni*) ya da ortalama alma yöntemlerini kullanırlar (Devlin ve diğer., 2019, Reimers ve Gurevych, 2019).Bu tezde de benzer bir yaklaşım olarak *masked mean pooling* (padding konumları maske ile dışlanarak yalnızca geçerli token temsillerinin ortalamasının alınması) uygulanmıştır. Böylece dizilerin genel bağlamını doğru şekilde yansıtan sabit boyutlu temsiller elde edilmiştir.

Vektörler önce birleştirilir. Ardından bu birleşmiş vektörler, çok katmanlı bir yapıya gönderilir. Bu yapı sırasıyla 512, ardından 256 ve en sonunda 1 birime düşecek şekilde tasarlanmıştır. Katmanlar arasında ReLU aktivasyonu kullanılır. Ayrıca aşırı öğrenmeyi engellemek için *dropout* (%10 olasılıkla ara katman aktivasyonlarının rastgele sıfırlanması yoluyla modelin belirli nöronlara aşırı bağımlılığının azaltılması) uygulanır. Protein ve molekül temsillerini içeren iki farklı sembolik vektör bu adımda birleşmiş olur. Bu birleşmiş vektörler, tam bağlantılı katmanlarda işlenerek bağlanma gücünü ya da etkileşim skorunu hesaplar. Transformer tabanlı ilaç-hedef etkileşim modellerinde araştırmacılar genelde bu yöntemi kullanır (Chen ve diğer., 2020, Huang ve diğer., 2021).

2.4 Yapay Sinir Ağları

Yapay sinir ağları (YSA), çok sayıda basit hesaplama biriminin (nöron) katmanlar hâlinde birleştirilmesiyle tanımlanan, uçtan uca öğrenilebilir fonksiyonlardır (Haykin, 2009, Goodfellow ve diğer., 2016). Tek bir nöron, giriş vektörünü ağırlıklandırır, özetler ve doğrusal olmayan bir aktivasyon fonksiyonundan geçirir:

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right). \tag{2.9}$$

Burada x_i girişler, w_i öğrenilen ağırlıklar, b ise bias terimidir. Nöronların ardışık katmanlar hâlinde bağlanmasıyla çok katmanlı sinir ağları (MLP) elde edilir (Goodfellow ve diğer., 2016).

L katmanlı klasik bir ileri beslemeli ağda gizli temsiller,

$$\mathbf{h}^{(l)} = f\left(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\right), \quad \mathbf{h}^{(0)} = \mathbf{x} \tag{2.10}$$

şeklinde hesaplanır. Bu çalışmada hem son tahmin başlığı (MLP) hem de Transformer içindeki beslemeli alt ağlar, bu genel formun özel birer örneği olarak tasarlanmıştır (Goodfellow ve diğer., 2016).

2.4.1 Aktivasyon Fonksiyonları

2.4.1.1 ReLU (Rectified Linear Unit)

ReLU en yaygın kullanılan aktivasyonlardan biridir ve negatif girdileri sıfırlar:

$$\mathrm{ReLU}(x) = \max(0, x). \tag{2.11}$$

Doğrusal olmama durumu kazandırırken, pozitif bölgede gradyanı sabit tuttuğu için gradyan kaybolması problemini de azaltır (Nair ve Hinton, 2010). Önerilen modelde MLP tabanlı tahmin başlığında ReLU aktivasyonu kullanılmaktadır.

2.4.1.2 GELU (Gaussian Error Linear Unit)

Eşitlik (2.12)'de Gaussian Error Linear Unit (GELU) aktivasyon fonksiyonunun tanımı verilmiştir (Hendrycks ve Gimpel, 2016).

$$\mathrm{GELU}(x) = x\,\Phi(x) = x \cdot \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{x}{\sqrt{2}}\right)\right] \tag{2.12}$$

Burada x aktivasyon fonksiyonuna giren skaler girdiyi, $\Phi(x)$ standart normal dağılımın kümülatif dağılım fonksiyonunu (CDF) ve erf$(\cdot)$ hata fonksiyonunu ifade eder. GELU, girdiyi sert bir eşikleme ile kesmek yerine $\Phi(x)$ üzerinden büyüklüğe bağlı olarak ağırlıklandırdığından, özellikle Transformer benzeri derin mimarilerde yumuşak bir aktivasyon davranışı sağlar (Hendrycks ve Gimpel, 2016).

2.4.1.3 GeGLU (Gated GELU)

Bu çalışmada Transformer enkoderinin beslemeli alt ağında (FFN), standart iki katmanlı MLP yerine GeGLU kullanılmıştır (Shazeer, 2020). Eşitlik (2.13)'de GeGLU dönüşümü tanımlanmaktadır.

$$\mathrm{GeGLU}(\mathbf{x}) = \big(\mathbf{x}\mathbf{W}_v + \mathbf{b}_v\big) \odot \mathrm{GELU}\big(\mathbf{x}\mathbf{W}_g + \mathbf{b}_g\big) \tag{2.13}$$

Burada $\mathbf{x}$ girdi vektörünü, $\mathbf{W}_v, \mathbf{b}_v$ değer (value) dönüşümüne ait ağırlık ve bias terimlerini, $\mathbf{W}_g, \mathbf{b}_g$ ise kapı (gate) dönüşümüne ait ağırlık ve bias terimlerini ifade eder. GELU$(\cdot)$ Gaussian Error Linear Unit aktivasyon fonksiyonudur ve $\odot$ operatörü bileşen bazında (Hadamard) çarpımı göstermektedir. Parametre sayısını standart FFN ile karşılaştırılabilir düzeyde tutmak amacıyla ara boyut genellikle $d_{\text{glu}} \approx \left\lfloor \frac{2}{3} d_{\text{ff}} \right\rfloor$ seçilerek ölçeklendirilir (Shazeer, 2020).

2.4.2 Gömme Katmanları (Embedding Layers)

Girdi olarak hem protein aminoasit dizileri hem de ligand SMILES dizileri kullanılmaktadır. Her bir sembol, öğrenilebilir bir gömme (embedding) vektörüne eşlenir. Bir diziyi $\mathbf{s} = (s_1, \ldots, s_L)$ ile gösterirsek, token gömme ve konumsal gömme

toplamı ile ilk gizli temsil elde edilir:

$$\mathbf{e}_i = \mathbf{E}[s_i], \qquad \mathbf{h}_i^{(0)} = \mathbf{e}_i + \mathbf{p}_i,$$

burada $\mathbf{E}$ token gömme matrisi, $\mathbf{p}_i$ ise her pozisyona karşılık gelen konumsal gömme vektörüdür (Goodfellow ve diğer., 2016). Önerilen modelde protein ve SMILES için ayrı gömme matrisleri ve ayrı konumsal gömme katmanları bulunmaktadır.

2.5 Hiperparametre Optimizasyonu

Derin öğrenme modelleri iyi sonuç vermek için bazı ayarlara ihtiyaç duyar. Bu ayarlara hiperparametre denir (Goodfellow ve diğer., 2016, Bergstra ve Bengio, 2012, Bischl ve diğer., 2023). Bunlar katman sayısı, gömme boyutu, dropout oranı, öğrenme oranı ve batch boyutu gibi ayarlardır. Hiperparametreler, modelin doğru öğrenmesini doğrudan etkiler. Bu tezde hiperparametrelerin optimize edilmesi için Optuna isimli bir kütüphane kullanılmıştır (Akiba ve diğer., 2019).

2.5.1 Optuna Kütüphanesi

Optuna, hiperparametre optimizasyonu yapmak için tasarlanmış modern bir araçtır (Akiba ve diğer., 2019). Kara-kutu (*black-box*) tipindeki hedef fonksiyonları optimize etmek için kullanılır. Kullanıcı öncelikle bir amaç fonksiyonu (*objective function*) belirler. Bu fonksiyon, verilen hiperparametrelerle modeli eğitir. Daha sonra doğrulama setinde modelin başarısını ölçer ve sonuç olarak bir performans değeri verir. Optuna sürekli olarak farklı hiperparametreler dener ve en iyi sonucu veren hiperparametre setini seçmeye çalışır. Böylece en başarılı modeli bulur.

Kütüphane temel olarak üç ayrı parçadan oluşur. Bunlardan ilki olan Study, optimizasyon çalışmalarının ana çerçevesini belirler. Bu parça, çalışmanın hedefini (minimize veya maximize etmek gibi), hangi örnekleyicinin seçildiğini ve budama işleminin nasıl yapılacağını düzenler. Diğer bir parça ise Sampler adı verilen

örnekleyicidir. Bu tezde Tree-structured Parzen Estimator (TPE) isimli Bayes temelli bir yöntem kullanılmıştır (Bergstra ve diğer., 2011). TPE yöntemi, önceki sonuçlara bakarak sonraki denemeleri başarılı sonuç verebilecek alanlara yönlendirir. Bu da deneme-yanılma yönteminden daha verimli sonuçlar sağlar. Son parça Pruner olarak bilinir ve amacı hesaplama yükünü azaltmaktır. Pruner, başarısız olacağı anlaşılan denemeleri erkenden iptal ederek kaynakları korur. Bu çalışmada MedianPruner yöntemi kullanılmıştır. MedianPruner, mevcut denemenin sonuçlarını önceki denemelerin ortanca değeriyle kıyaslayarak düşük performans gösteren denemeler erkenden sonlandırılmış, böylece zaman ve işlem gücü tasarrufu sağlanmıştır.

Optuna'nın klasik yöntemler olan grid arama ya da rastgele aramaya göre bazı net avantajları vardır (Bergstra ve Bengio, 2012, Bischl ve diğer., 2023). Örneğin sürekli tipte hiperparametrelerle, ayrık tiptekileri aynı yapı içinde rahatça bir araya getirebilir. Bütçe ya da zaman gibi hesaplama kaynakları kısıtlı olduğunda budama tekniği devreye girer ve gereksiz işlemlerden kaçınılır. Bu da daha hızlı ve ekonomik sonuçlar demektir. Hiperparametrelerin kendi içlerinde karmaşık bağlantıları vardır, Optuna bu bağlantıları kolaylıkla çözümler ve modeller. Bu tezde tek hedefli optimizasyon (doğrulama RMSE minimizasyonu) kullanıldı.

2.5.2 Optuna Kütüphanesinin Kullanımı

Bu tezde tasarlanan çapraz-dikkat tabanlı DTA modeli için hiperparametre optimizasyonu, veri kümesinden bağımsız olacak şekilde aynı yöntemle yürütülmüş; ancak her veri kümesi (örneğin Davis ve KIBA (Davis ve diğer., 2011, Tang ve diğer., 2014, Öztürk ve diğer., 2018)) için ayrı bir *study* oluşturularak optimizasyonlar bağımsız olarak gerçekleştirilmiştir. Böylece her veri kümesi için, etiket dağılımı ve dizi uzunlukları gibi özelliklere uyumlu, kendi en iyi hiperparametre kombinasyonu elde edilmiştir. Burada sunulan kod parçası, Davis veri kümesi üzerinde çalışan model için kullanılan Optuna kurulumuna karşılık gelmektedir.

Optimizasyon sürecinin odağında, Optuna tarafından tekrar tekrar çağrılan bir

amaç fonksiyonu bulunmaktadır. Bu fonksiyon, her bir deneme (trial) için şu adımları uygular:

1. Deneme kapsamında Optuna tarafından önerilen bir hiperparametre konfigürasyonu θ alınır.
2. Bu konfigürasyona sahip model oluşturulur ve sabit eğitim/ doğrulama bölünmesi üzerinde belirli sayıda epoch boyunca eğitilir.
3. Her epoch sonunda doğrulama kaybı ve doğrulama kök ortalama kare hatası (RMSE) hesaplanır.
4. En iyi doğrulama kaybı ve RMSE değeri tutulur; eğitimin erken durdurma kriteri sağlanırsa (belirli sayıda epoch boyunca iyileşme olmazsa) eğitim sonlandırılır.
5. Denemenin nihai skoru olarak doğrulama RMSE değeri $f(\theta) = \mathrm{RMSE}_{\mathrm{val}}(\theta)$ Optuna'ya geri döndürülür.

Hiperparametre optimizasyonu, doğrulama kümesinde hesaplanan kök ortalama karesel hatayı (RMSE) en küçükleyen hiperparametrelerin seçilmesi olarak ele alınabilir. Bu amaçla, doğrulama RMSE'sini minimize eden optimizasyon problemi Eşitlik (2.14)'de verilmiştir.

$$\theta^* = \arg\min_{\theta \in \Theta} \mathrm{RMSE}_{\mathrm{val}}(\theta) \tag{2.14}$$

Burada Θ, önceden tanımlanan hiperparametre arama uzayını; θ, bu uzaydan seçilen belirli bir hiperparametre kombinasyonunu; θ^* ise doğrulama kümesindeki hata ölçütü $\mathrm{RMSE}_{\mathrm{val}}(\theta)$ değerini en aza indiren optimal hiperparametreleri ifade eder.

Her deneme için eğitim, ortalama kare hata (MSE) kaybı ile yapılmakta, Optuna için geriye döndürülen amaç değeri ise bu kayıttan türetilen doğrulama RMSE'si olmaktadır. Ayrıca her epoch sonunda Optuna'nın `trial.report` fonksiyonu ile ara

skorlar raporlanmakta ve `trial.should_prune` çağrısı ile MedianPruner, düşük performanslı denemeleri erken budayabilmektedir.

Optuna üzerinde tanımlanan hiperparametre arama uzayı, hem mimari hem de eğitimle ilgili çeşitli büyüklükleri içermektedir. Bu çalışmada kullanılan başlıca hiperparametreler ve aralıkları aşağıda özetlenmiştir:

Gömme boyutu (d_{model}): {192, 256, 320, 384}. Bu parametre, protein ve SMILES dizileri için kullanılan gömme (*embedding*) ve Transformer katmanlarının gizil boyutunu belirler.

Çoklu başlık sayısı (n_{head}): {4, 8, 16} kümesinden seçilir, ancak Optuna deneme sırasında d_{model} ile uyumlu (tam bölen) başlık sayıları ile arama yapılır. Böylece çoklu-başlıklı dikkat mekanizmasının boyut uyumsuzluğu engellenir.

Encoder katman sayısı (`num_encoder_layers`): Protein ve SMILES için kullanılan Transformer encoder bloklarının sayısı 1 ile 3 arasında örneklenir.

Çapraz-dikkat katmanı sayısı (`num_cross_layers`): Protein ve ligand gösterimleri arasında karşılıklı dikkat uygulayan katman sayısı 1 ile 2 arasında seçilir.

FFN genişliği (`dim_feedforward`): GeGLU temelli beslemeli ağın ara boyutu, {768, 1024, 1536, 2048} kümesinden seçilir.

Dropout oranı (`dropout`): Hem encoder hem de yoğun katmanlarda kullanılan dropout oranı, $[0.0, 0.30]$ aralığından sürekli bir değer olarak örneklenir.

Maksimum dizi uzunlukları (`prot_max_len`, `smi_max_len`): Protein dizileri için maksimum uzunluk {800, 1000, 1200}, SMILES dizileri için maksimum uzunluk {80, 100, 120, 150} değerleri arasından seçilir. Bu parametreler, padding ve gömme boyutları üzerinden hem hesaplama maliyetini hem de modelin bağlam penceresini etkiler.

Batch büyüklüğü (`batch_size`): {16, 32, 64} değerleri arasından seçilir; eğitim

stabilitesi ile bellek kullanımı arasında bir denge gözetilir.

Öğrenme oranı (`lr`): $[10^{-5}, 5 \times 10^{-4}]$ aralığından log-ölçekli olarak örneklenir. Böylece hem çok küçük hem de nispeten büyük öğrenme oranları verimli bir şekilde taranabilir.

Model eğitilirken farklı değerler denenir ve bu değerlere göre en iyi sonuç taranır. Katman sayısı veya katmanların genişliği değiştirilebilir. Dropout oranı ayarlanır, batch büyüklüğü ve öğrenme oranı gibi parametrelerle oynamalar yapılır (Chen ve diğer., 2020, Huang ve diğer., 2021).

Her yeni denemede doğrulama hatası takip edilir. Eğer doğrulama hatası belli bir süre azalmazsa eğitim erkenden durdurulur (*early stopping*). Bu durumda, o ana kadar elde edilen en düşük doğrulama RMSE değeri Optuna sistemine aktarılır.

Optuna içerisindeki `MedianPruner` bileşeni, farklı denemeleri karşılaştırarak zayıf sonuç verenleri erkenden devre dışı bırakır. Özellikle epoch sayılarının arttığı senaryolarda hesaplama yükünü azaltır bu yöntem. Böylece, daha yüksek potansiyel barındıran hiperparametre setleri için kaynakları etkin biçimde yönlendirmek mümkün olur ve kombinasyonlarının denenmesine imkan tanır.

2.5.3 En İyi Konfigürasyon ile Yeniden Eğitim ve Test

Optuna optimizasyonu tamamlandığında, her veri kümesi için en düşük doğrulama RMSE değerini sağlayan deneme t^* ve buna karşılık gelen hiperparametre konfigürasyonu θ^* elde edilir. Bu konfigürasyon, izleyen adımlarda şu şekilde kullanılmaktadır:

1. En iyi denemeye ait hiperparametreler `best_trial.params` üzerinden alınarak `best_cfg` adlı yapılandırma oluşturulur.
2. Model, bu `best_cfg` kullanılarak eğitim ve doğrulama kümeleri üzerinde yeniden eğitilir. Bu aşamada epoch sayısı ve *patience* değeri, daha istikrarlı bir

yakınsama elde edebilmek için HPO sırasında kullanılan ayarlardan daha uzun seçilebilir.

3. Eğitim süreci sonunda, eğitilmiş model, HPO sürecine hiç dahil edilmemiş olan bağımsız test kümesi üzerinde değerlendirilir. Test kümesi için ortalama kare hata (MSE), kök ortalama kare hata (RMSE), uyum indeksi (CI) ve belirli bir eşik değerine göre hesaplanan Kesinlik–Geri Çağırım Eğrisi Altındaki Alan (AUPR) gibi farklı ölçütler raporlanır (Pahikkala ve diğer., 2015, Öztürk ve diğer., 2018).

Bu çalışmada Optuna ile hiperparametre optimizasyonu 2. aşama için uygulanmış olup 1. aşamada sabit parametreler ile farklı mimariler denenmiştir.

Optuna ile hiperparametreler belirlenirken modelin karmaşıklığına ve aşırı öğrenmeye dikkat edilir. Bu yöntemde amaç, doğrulama kümesindeki RMSE değerini en aza indirmektir. Elde edilen parametreler bağımsız test verisine uygulanır. Böylece modelin farklı verilerdeki başarısı anlaşılır. Davis ve KIBA gibi farklı veri setlerinde optimizasyon işlemi ayrı ayrı yapılmıştır. Her veri seti için farklı ve en uygun hiperparametreler seçilmiştir (Chen ve diğer., 2020, Huang ve diğer., 2021).

2.6 Verisetinin Ayrılması

Modelin eğitimi, doğrulama ve test süreçlerinde kullanılacak veriler üç farklı alt kümeye ayrılmıştır. Bu işlem iki adımda tamamlanmış ve Python ortamında yaygın biçimde kullanılan scikit-learn kütüphanesindeki `train_test_split` fonksiyonundan yararlanılmıştır (Pedregosa ve diğer., 2011).

Öncelikle, tüm veri kümesi rastgele olarak %80'i eğitim setine ayrılacak biçimde iki parçaya bölünmüştür. Geriye kalan %20'lik kısım ise geçici bir küme olarak saklanmıştır:

$$\text{train_df},\ \text{temp_df} = \texttt{train_test_split}\Big(\text{df},\ \texttt{test_size} = 0.2,\ \texttt{random_state} = 42\Big). \tag{2.15}$$

Sonrasında, elde edilen geçici küme tekrar iki eşit parçaya ayrılarak doğrulama ve test kümeleri oluşturulmuştur:

$$\text{val_df},\ \text{test_df} = \texttt{train_test_split}\Big(\text{temp_df},\ \texttt{test_size} = 0.5,\ \texttt{random_state} = 42\Big). \tag{2.16}$$

Tüm bu adımların ardından veri setinin son hali şu şekildedir:

Eğitim seti, tüm verinin %80'ini kapsar durumdadır. Doğrulama ve test setleri ise verilerin kalan %20'lik kısmını eşit olarak paylaşarak, ayrı ayrı %10'luk dilimleri oluşturmaktadır.

Her iki bölümleme adımında da `random_state=42` parametresi kullanılarak sonuçların tekrarlanabilirliği sağlanmıştır. Böylece aynı veri seti üzerinde farklı zamanlarda yapılan deneylerde aynı train/validation/test ayrımının elde edilmesi garanti altına alınmış ve tüm mimarilerin adil bir şekilde karşılaştırılması mümkün olmuştur. Eğitim, validasyon ve test alt kümelerinin bu şekilde ayrıştırılması, makine öğrenimi literatüründe önerilen deneysel kurguya uygundur (Goodfellow ve diğer., 2016, Hastie ve diğer., 2009).

Eğitim seti, modelin ağırlıklarını güncellemek ve öğrenme sürecini gerçekleştirmek için kullanılmıştır. Validasyon seti, eğitim sırasında her epoch sonunda modelin performansını izlemek, en iyi epoch'u seçmek ve hiperparametre aramalarında değerlendirme metriği olarak kullanılmak üzere ayrılmıştır. Test seti ise yalnızca eğitim tamamlandıktan ve en iyi model seçildikten sonra, modelin genelleme yeteneğini değerlendirmek amacıyla kullanılmıştır. Bu sayede test seti eğitim

sürecinden tamamen bağımsız kalmış ve modelin gerçek performansı tarafsız biçimde ölçülmüştür (Goodfellow ve diğer., 2016).

BÖLÜM ÜÇ
UYGULAMA

3.1 Verisetlerinin Açıklanması

Modelin başarısının ölçülmesi amacıyla, literatürde sık kullanılan DAVIS ve KIBA isimli iki veri seti seçilmiştir (Davis ve diğer., 2011, Tang ve diğer., 2014). Bu veri setlerinin, kinaz inhibitörlerinin proteinlerle nasıl bağlandığını gösteren verileri içerdiği belirtilmiştir. İlaç keşfi alanında çalışan birçok araştırmacı tarafından değerlendirme süreçlerinde bu veri setlerinin kullanıldığı ifade edilmiştir. Bu nedenle, modelin performansının anlaşılması için söz konusu veri setlerinin uygun olduğu değerlendirilmiştir.

3.1.1 DAVIS Veriseti

DAVIS veriseti (Davis ve diğer., 2011), kinaz inhibitörlerinin seçiciliğini ölçmek amacıyla oluşturulmuş kapsamlı bir biyokimyasal veri koleksiyonudur. Orijinal çalışmada 72 kinaz inhibitörü, 442 kinaz hedefi üzerinde test edilmiştir. DTA literatüründe yaygın olarak kullanılan işlenmiş DAVIS sürümü ise 68 ilaç, 442 protein ve 30 056 ilaç-hedef etkileşiminden oluşmaktadır (Öztürk ve diğer., 2018, Nguyen ve diğer., 2021). Bu tez kapsamında da aynı benchmark sürüm kullanılmıştır.

Orijinal çalışmada nanomolar (nM) biriminde raporlanan dissosiasyon sabiti değerlerini pK_d ölçeğine dönüştürmek için kullanılan dönüşüm Eşitlik (3.1)'de verilmiştir.

$$pK_d = -\log_{10}\left(\frac{K_d}{10^9}\right) \tag{3.1}$$

Burada K_d, nanomolar (nM) cinsinden protein–ligand kompleksinin denge durumundaki dissosiasyon sabitini ifade eder. $\log_{10}(\cdot)$ taban 10 logaritmayı göstermektedir ve 10^9 çarpanı nM birimindeki K_d değerinin molar (M) birime

dönüştürülmesi için kullanılır.

Bu çalışmada DAVIS adlı veri seti kullanılmış ve söz konusu veri setinde toplam 30,056 ilaç–protein çifti yer almaktadır. Veri setindeki proteinler 442 farklı kinazdan oluşmakta olup, bu proteinlerle etkileşen 68 kinaz inhibitörü bulunmaktadır. İlaç–protein çiftlerinin biyolojik aktiviteleri, sürekli bir ölçüt olan pK_d skorları ile ifade edilmiştir. Ölçülen pK_d değerlerinin 5.0 ile 10.8 aralığında değiştiği görülmekte ve afinite değerlerinin ortalaması 5.45 ve standart sapması 0.89 olarak hesaplanmıştır.

DAVIS verisetinde etiketlerin istatistiksel dağılımı Tablo 3.1'te verilmiştir. Verisetinin yaklaşık %75'inde bağlanma afinitesi 5.52 pKd değerinin altındadır; bu da çoğu ilaç-protein çiftinin orta-zayıf afiniteye sahip olduğunu göstermektedir.

Tablo 3.1 DAVIS veri setindeki bağlanma afinitesi (pKd) değerlerinin istatistiksel özeti

İstatistik	**Değer**
Örnek Sayısı	30 056
Ortalama	5.45
Standart Sapma	0.89
Minimum	5.00
1. Çeyrek (%25)	5.00
Medyan (%50)	5.00
3. Çeyrek (%75)	5.52
Maksimum	10.80

3.1.2 KIBA Veriseti

KIBA (Kinase Inhibitor BioActivity) veriseti (Tang ve diğer., 2014), birden fazla kaynaktan (ChEMBL (Gaulton ve diğer., 2012), BindingDB Liu ve diğer. (2007), PubChem Bolton ve diğer. (2008) ve STITCH Szklarczyk ve diğer. (2016)) derlenmiş kapsamlı bir kinaz inhibitör biyoaktivite veritabanıdır. DAVIS verisetinden farklı olarak, KIBA veriseti Ki, Kd ve IC_{50} gibi birden fazla biyoaktivite ölçümünü, Tang ve arkadaşlarının önerdiği model tabanlı bir entegrasyon yöntemi ile tek bir KIBA skoruna dönüştürmektedir. KIBA skoru, bu farklı biyoaktivite tipleri arasındaki istatistiksel tutarlılığı eniyileyecek şekilde tanımlanmış bir bileşik biyoaktivite ölçüsüdür; ayrıntılı formülasyon orijinal çalışmada sunulmaktadır (Tang ve diğer.,

2014).

Bu tezde, ilaç-hedef etkileşimlerinin incelendiği çalışmalarda sıkça başvurulan KIBA veri seti tercih edilmiştir. Bu veri seti daha önce SimBoost (He ve diğer., 2017), DeepDTA (Öztürk ve diğer., 2018) ve GraphDTA (Nguyen ve diğer., 2021) gibi bilinen çalışmalarda da yaygın olarak kullanılmıştır. Veri setinde 2.111 farklı ilaç ve bunlarla etkileşime giren 229 kinaz hedef proteini bulunmaktadır. Toplam etkileşim sayısı ise 118,254 olarak belirlenmiştir (Öztürk ve diğer., 2018, Nguyen ve diğer., 2021).

Orijinal KIBA skorlarında küçük değerler daha yüksek afiniteleri temsil ettiğinden, bu tezde literatürle uyumlu olacak şekilde skorların işareti ters çevrilmiş ve tüm değerlere sabit bir ofset eklenmiştir. Böylece etiketler 0.0–17.2 aralığına yeniden ölçeklendirilmiş ve *daha büyük KIBA skorları daha güçlü bağlanma afinitesini* temsil eder hâle getirilmiştir.

Bu çalışmada kullanılan KIBA veri setinde toplam 118,254 adet ilaç–protein çifti bulunduğu belirtilmiştir. Veri setinde 229 farklı kinaz proteini yer almaktadır. Bu proteinlerle etkileşime giren ilaç sayısı ise 2,111 dir. Kullanılan etiketlerin sürekli değerlerden oluştuğu ve yeniden ölçeklendirilmiş KIBA skorları ile gösterilmiş ve söz konusu skorların 0.0 ile 17.2 aralığında değiştiği ifade edilmiştir. Veri setindeki ortalama aktivite değerinin 11.72 ve standart sapmanın 0.84 olarak hesaplandığı bildirilmiştir.

KIBA verisetindeki etiket dağılımı Tablo 3.2'te sunulmuştur. Etiketlerin büyük kısmı 11.20–11.92 aralığında yoğunlaşmakta, 15 ve üzeri skorlar ise daha az sayıda fakat yüksek afiniteye sahip etkileşimleri temsil etmektedir.

Verisetleri, eğitim (%80), doğrulama (%10) ve test (%10) setlerine ayrılmıştır. Bölünme işlemi, DeepDTA ve GraphDTA çalışmalarında olduğu gibi, ilaç-hedef etkileşim çiftleri bazında rastgele yapılmıştır (Öztürk ve diğer., 2018, Nguyen ve diğer., 2021). Dolayısıyla aynı ilaç veya protein hem eğitim hem de test kümelerinde bulunabilmekte; ancak her bir ilaç-hedef çifti yalnızca tek bir alt kümede

Tablo 3.2 KIBA verisetindeki biyoaktivite skorlarının istatistiksel özeti

İstatistik	**Değer**
Örnek Sayısı	118 254
Ortalama	11.72
Standart Sapma	0.84
Minimum	0.00
1. Çeyrek (%25)	11.20
Medyan (%50)	11.50
3. Çeyrek (%75)	11.92
Maksimum	17.20

yer almaktadır. Bu ayar, literatürde yaygın olarak kullanılan *random split* benchmark senaryosuyla birebir uyumludur.

3.2 Metriklerin Açıklanması

Bu çalışmada geliştirilen ilaç–hedef etkileşim (DTA) modellerinin ne kadar iyi sonuç verdiği, farklı boyutlardan bakan çeşitli değerlendirme yöntemleriyle incelenmiştir. Tahminlerin doğruluğu tek başına yeterli bir kriter olarak görülmemiş; bileşiklerin kendi aralarındaki sıralama uyumunun da dikkate alınması gerektiği vurgulanmıştır. Bu sebeple, DAVIS ve KIBA veri kümeleriyle yapılan analizlerde, literatürde standart hâline gelmiş dört farklı değerlendirme ölçütünden faydalanılmıştır (Öztürk ve diğer., 2018, Lin ve diğer., 2020, D'Souza ve diğer., 2023). Bu ölçütlerden biri, tahminlerdeki sapmayı gösteren ortalama kare hata (Mean Squared Error, MSE) olup, onun daha duyarlı hâli olan kök ortalama kare hata (Root Mean Squared Error, RMSE) ile birlikte kullanılmıştır. Uyum indeksi (Concordance Index, CI) ise modellerin gerçek verilerle ne kadar tutarlı olduğunun değerlendirilmesinde kullanılmıştır. Ayrıca, modellerin özellikle sınıflandırma tahminlerindeki performansını değerlendiren kesinlik–geri çağırım eğrisi altındaki alan (Kesinlik–Geri Çağırım Eğrisi Altındaki Alan, AUPR) da değerlendirmeye alınmıştır. İzleyen kısımlarda bu ölçütlerin her biri kısaca tanıtılarak, ilaç–hedef etkileşimi bağlamında tam olarak neyi ifade ettikleri açıklanacaktır.

3.2.1 Ortalama Kare Hata (MSE)

Ortalama kare hata (Mean Squared Error, MSE) tanımı Eşitlik (3.2)'de verilmiştir.

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2 \tag{3.2}$$

Burada N, değerlendirmeye dahil edilen toplam örnek sayısını; y_i, i'inci örneğe ait gerçek (deneysel) değeri; $\hat{y}_i$ ise aynı örnek için model tarafından üretilen tahmini değeri ifade eder. Hata terimlerinin karesinin alınması, büyük sapmaların metriğe daha fazla katkı yapmasına neden olur. DTA çalışmalarında daha düşük MSE, tahmin edilen afinite değerlerinin deneysel ölçümlere daha yakın olduğunu gösterir (Öztürk ve diğer., 2018, D'Souza ve diğer., 2023).

3.2.2 Kök Ortalama Kare Hata (RMSE)

Eşitlik (3.3)'de kök ortalama kare hata (Root Mean Squared Error, RMSE) tanımı verilmiştir.

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2} \tag{3.3}$$

Burada N, değerlendirmeye alınan örnek sayısını; y_i, i'inci örneğin gerçek hedef değerini; $\hat{y}_i$, modelin i'inci örnek için ürettiği tahmini; MSE ise ortalama kare hatayı ifade eder. RMSE, hata büyüklüğünü hedef değişkenin birimi cinsinden verdiği için, tipik tahmin sapmasının düzeyini doğrudan yorumlamaya imkân tanır. Örneğin hedef değişken log-dönüştürülmüş bağlanma afinitesi (örneğin pK_d) olarak tanımlanmışsa, RMSE değeri modelin bu ölçekte yaptığı ortalama sapmayı gösterir.

3.2.3 Uyum İndeksi (Concordance Index, CI)

Uyum İndeksi, modelin örnekler arasındaki *göreli sıralamayı* ne kadar doğru yakaladığını ölçen bir metriktir. Başlangıçta sağkalım analizinde risk skoru üreten modellerin ayırt ediciliğini değerlendirmek için tanımlanan konkordans olasılığına dayanır (Gönen ve Heller, 2005). DTA problemlerinde çoğu zaman esas amaç, bileşiklerin tam afinite değerinden ziyade güçlü ve zayıf bağlanan adayların doğru sıralanması olduğundan, CI özellikle kritik bir rol oynar; DeepDTA ve onu izleyen pek çok DTA çalışmasında temel sıralama metriği olarak kullanılmıştır (Öztürk ve diğer., 2018, Lin ve diğer., 2020, D'Souza ve diğer., 2023).

Uyum indeksi (Concordance Index, CI), model tahminlerinin gerçek değerlerin sıralamasıyla ne ölçüde tutarlı olduğunu ölçmek için kullanılır. Eşitlik (3.4)'de CI ölçütünün tanımı verilmiştir.

$$\mathrm{CI} \;=\; \frac{1}{N_{\text{pairs}}} \sum_{i<j} \mathbb{I}[(y_i - y_j)(\hat{y}_i - \hat{y}_j) > 0] \tag{3.4}$$

Burada y_i ve y_j sırasıyla i ve j örneklerine ait gerçek değerleri, $\hat{y}_i$ ve $\hat{y}_j$ aynı örnekler için model tahminlerini ifade eder. Toplam N_{pairs}, $y_i \neq y_j$ koşulunu sağlayan ve bu nedenle karşılaştırılabilir kabul edilen tüm (i, j) çiftlerinin sayısıdır. $\mathbb{I}[\cdot]$ gösterge fonksiyonu olup, parantez içindeki koşul sağlandığında 1, aksi durumda 0 değerini alır. CI değeri 0 ile 1 arasında değişir; 0.5 rastgele sıralamaya, 1.0 ise mükemmel sıralamaya karşılık gelir.

3.2.4 Ortalama Kesinlik–Geri Çağırım Eğrisi Altındaki Alan (AUPR)

DTA, temel olarak bir regresyon problemi olmasına rağmen, pratikte araştırmacılar çoğu zaman "güçlü bağlayıcı" bileşikleri seçmeye odaklanmaktadır. Bu motivasyonla, çalışmada ayrıca modelin yüksek afiniteye sahip bileşikleri ne kadar iyi ayırt edebildiğini ölçmek için sınıflandırma temelli bir metrik olarak AUPR

kullanılmıştır. AUPR, özellikle dengesiz veri kümelerinde ROC eğrisi altındaki alana (AUC) göre daha duyarlı bir ölçü olarak önerilmektedir (Saito ve Rehmsmeier, 2015) ve son yıllarda DTA çalışmalarında da güçlü bağlayıcıların geri çağrımını değerlendirmek için yaygın biçimde kullanılmaktadır (Lin ve diğer., 2020, D'Souza ve diğer., 2023).

Öncelikle, belirli bir eşik değeri t bu çalışmada DAVIS veri seti için literatürde yaygın kullanılan değer izlenerek $t = 7.0$ alınmıştır. KIBA veri setinde etiketler 0.0–17.2 aralığına yeniden ölçeklendirildiğinden, AUPR hesabında 'güçlü bağlayıcı' sınıfı için literatürde yaygın kullanılan eşik izlenerek $t = 12.1$ seçilmiştir. (Öztürk ve diğer., 2018, He ve diğer., 2017) (D'Souza ve diğer., 2023)). Bu değerlerin üzerinde bağlanma afinitesine sahip örnekler *pozitif* (güçlü bağlayıcı), diğerleri ise *negatif* sınıf olarak etiketlenir:

$$y_i^{(\text{bin})} = \begin{cases} 1, & y_i > t \\ 0, & y_i \leq t. \end{cases} \tag{3.5}$$

AUPR değeri, özellikle pozitif sınıfın (güçlü bağlayıcıların) az sayıda olduğu dengesiz veri kümelerinde, ROC eğrisi altındaki alana göre daha duyarlı bir ölçüdür (Saito ve Rehmsmeier, 2015). Bu nedenle yüksek AUPR, modelin güçlü bağlanma gösteren bileşikleri geri çağırma ve yanlış pozitifleri azaltma konusunda başarılı olduğuna işaret eder.

Sonuç olarak, bu dört metrik birlikte ele alındığında, geliştirilen modelin hem bağlanma afinitesini sayısal olarak ne kadar doğru tahmin ettiği (MSE, RMSE), hem örnekleri ne kadar doğru sıraladığı (CI), hem de istatistiksel olarak öngörücü ve güçlü bağlayıcıları ne derece başarılı şekilde ayıklayabildiği (AUPR) bütüncül olarak değerlendirilebilmektedir. Bu metrikler, çalışmada DAVIS veri seti için ana model (TransformGeGLU) de dâhil olmak üzere tüm deneylerde tutarlı biçimde kullanılmıştır.

3.3 Ablasyon Çalışmaları

Bu bölümde, önerilen çapraz-dikkat tabanlı DTA mimarisinin hangi bileşen kombinasyonları ile en iyi performansı verdiğini incelemek amacıyla gerçekleştirilen kapsamlı ablasyon çalışmaları sunulmaktadır. Çalışma iki aşamalı olarak tasarlanmıştır:

Aşama 1: 4 farklı konumsal kodlama, 4 farklı normalizasyon şeması, 4 farklı FFN tipi ve 4 farklı çapraz-dikkat tipi olmak üzere toplam $4 \times 4 \times 4 \times 4 = 256$ mimari konfigürasyon, her biri 10 epoch eğitilerek taranmıştır.

Aşama 2: Aşama 1 sonucunda seçilen en iyi dört aday model için model-bazlı küçük ölçekli bir hiperparametre araması yapılarak her bir modelin kendi en iyi ayarı bulunmuş ve nihai karşılaştırma bu ayarlarla raporlanmıştır.

Buradaki amacımız, modelin kapasitesini hiç değiştirmeden, konumsal kodlama türleri, normalizasyon yöntemleri, FFN alt ağı ve çapraz dikkat bloklarının farklı kombinasyonlarının performansa nasıl yansıdığını tek tek incelemektir. Değerlendirmede hata oranları MSE ile ölçülmüştür. Modellerin tahminleri doğru sıralayıp sıralamadığını anlamak içinse CI değerine bakılmıştır. CI, bağlanma afinitesi çalışmalarında literatürde sıklıkla kullanılan bir değerlendirme yöntemidir (Öztürk ve diğer., 2018).

3.3.1 Deney Tasarımı

Ablasyon çalışması dört temel eksende şekillendirilmiştir. Bu eksenlerden ilki, dizilerin konumsal bilgisini modellemek amacıyla kullanılan kodlama türünü ifade eden `pos_type` parametresi olarak tanımlanmıştır. Protein ve SMILES dizileri için öğrenilebilir mutlak konum tablolarından yararlanıldığı durumda yöntem `abs` etiketiyle adlandırılmıştır. Yalnızca dikkat mekanizması içerisinde uygulanan Rotary Positional Embedding (RoPE) yaklaşımı `rope` adı altında tercih edilmiştir (Su ve diğer., 2021). Dikkat skorlarına doğrusal bir yanlılık (bias) ekleyerek göreli konum

bilgisini yansıtan ALiBi tekniği ise `alibi` olarak kullanılmıştır (Press ve diğer., 2021). Herhangi bir konumsal kodlama bileşeninin dahil edilmediği ve yalnızca token temsilleriyle işlem yapıldığı durumda ise `none` ifadesi kullanılmıştır.

Normalizasyon şemasına (`norm`) ilişkin denemeler, encoder ve çapraz-dikkat blokları için dört farklı seçenek üzerinden yürütülmüştür. Bu kapsamda RMSNorm, geleneksel LayerNorm yöntemine kıyasla alternatif bir normalizasyon stratejisi olarak ele alınmış ve kök-ortalama-kare prensibine dayandırılmıştır (Zhang ve Sennrich, 2019). Artık bağlantıların ardından LayerNorm uygulanması, klasik Transformer mimarilerinde yaygın biçimde kullanılan bir yaklaşım olarak değerlendirilmiş ve `post_ln` etiketiyle belirtilmiştir. Artık bağlantı sonrası RMSNorm kullanımı ise `post_rms` ismiyle ifade edilmiştir. Bununla beraber, artık bağlantıların hemen öncesinde LayerNorm kullanıldığı durumda (pre-norm yaklaşımı) ortaya çıkan strateji `pre_ln` olarak adlandırılmıştır. Aynı noktada RMSNorm tercih edilmesi durumunda da yöntem `pre_rms` şeklinde ifade edilmiştir.

FFN tipi (`ffn_type`) için dört farklı yapı denenmiştir. Bunlardan ilkinde iki katmanlı MLP'nin aktivasyon fonksiyonu olarak klasik ReLU (Rectified Linear Unit) fonksiyonu kullanılmıştır; bu yaklaşım `relu` etiketiyle ifade edilmiştir. Alternatif olarak, ReLU yerine GELU (Gaussian Error Linear Unit) aktivasyonu tercih edildiğinde yöntem `gelu` olarak anılmıştır. Bunlara kıyasla daha karmaşık bir yapıya sahip olan SwiGLU (Swish-Gated Linear Unit) metodu ise SiLU (Sigmoid Linear Unit) aktivasyonu ile birlikte kullanılarak `swiglu` ismiyle belirtilmiştir. Öte yandan GeGLU (GELU-Gated Linear Unit) yaklaşımı da GELU aktivasyonu ile bütünleştiğinde, `geglu` şeklinde adlandırılmıştır (Shazeer, 2020).

Çapraz-dikkat yapısı (`attn_type`) dört farklı yaklaşım üzerinden ele alınmıştır. Bu yaklaşımlardan ilki, protein ve ligand dizileri arasında uygulanan standart çift yönlü çok kafalı çapraz-dikkat mekanizması olup `standard` etiketiyle ifade edilmiştir. Alternatif olarak, artık bağlantıların üzerine eklenen skaler bir kapı (*gate*) mekanizmasıyla zenginleştirilmiş kapılı çapraz-dikkat yöntemi tercih edilmiş ve `gated` şeklinde tanımlanmıştır. Daha spesifik bir uygulama olan Grouped-Query

Attention (GQA) yaklaşımında ise sorgu kafalarının sayısı artırılırken anahtar ve değer kafaları azaltılarak kullanılmış; bu yöntem literatürde `gqa` olarak belirtilmiştir (Ainslie ve diğer., 2023). Bunların aksine, herhangi bir çapraz-dikkat modülünün dahil edilmediği; protein ve ligand dizilerinin yalnızca bağımsız encoder bloklarından geçirilip son aşamada basitçe birleştirildiği yaklaşım `none` adıyla ifade edilmiştir.

Aşama 1 kapsamında toplamda 256 farklı mimari varyasyon test edilirken, bu süreçte birtakım hiperparametreler sabit biçimde korunmuştur. Gömme boyutu ve genel model boyutu için $d_{\text{model}} = 256$ değeri kullanılmıştır. Dikkat mekanizması sekiz kafadan oluşacak şekilde $n_{\text{head}} = 8$ olarak belirlenmiştir. Protein ve SMILES dizileri için encoder blokları, her biri iki katmanlı olacak biçimde tasarlanmıştır. Çapraz-dikkat katman sayısı tek bir katman ile sınırlandırılmıştır. FFN iç boyutu $d_{\text{ff}} = 1024$ olarak ayarlanmıştır. Dropout oranı ise 0.1 düzeyinde sabit tutulmuştur.

3.3.2 Veri Kümesi ve Eğitim Kurulumu

Ablasyon analizlerinde, DAVIS veri seti temel olarak kullanılmıştır (Davis ve diğer., 2011). Veri dosyasında yer alan sütunlar, çalışmanın gereksinimlerine uygun olacak şekilde `ID`, `Target`, `SMILES`, `Sequence` ve `Label` biçiminde adlandırılmıştır. Hedef değişken olarak `Label` sütunu belirlenmiştir. Veriler, rastgele ve etkileşim bazlı bölme yaklaşımıyla eğitim, doğrulama ve test alt kümelerine ayrılmış; veri setinin %80'i eğitim için tahsis edilirken, kalan bölüm %10 doğrulama ve %10 test olacak şekilde eşit oranlarda dağıtılmıştır.

Her iki deney aşamasında da birtakım ortak parametreler belirlenmiştir. Her bir mini-batch, 32 örnek içerecek şekilde ayarlanmıştır. Optimizasyon yöntemi olarak Adam tercih edilmiştir (Kingma ve Ba, 2014) ve öğrenme oranı $\eta = 10^{-4}$ düzeyine ayarlanmıştır. Kayıp fonksiyonu olarak ortalama kare hata (MSE) seçilmiştir. Ayrıca, sonuçların tutarlı biçimde yeniden elde edilebilmesi için tüm deneylerde aynı seed (42) kullanılmıştır; böylelikle başlangıç ağırlıklarının ve veri bölünmelerinin

değişmemesi sağlanmıştır.

Aşama 1 sürecinde her yapı 10 epoch boyunca eğitilmiş, doğrulama kaybının en düşük olduğu anda oluşan ağırlıklar ayrıca kaydedilmiştir. Değerlendirme adımı da doğrudan bu kayıttan yürütülmüştür. Bu geniş tarama, toplam 256 seçeneğin yer aldığı mimari uzamda hangi bileşenlerin daha umut vadettiğini anlamak için kullanılmıştır. Bazen konumsal kodlama tercihleri, bazen de FFN'in yapısal özellikleri belirgin biçimde öne çıkmış; kimi denemeler ise yalnızca belirli örüntülerde sınırlı bir avantaj göstermiştir. Böylece hangi bölgelerin daha yakından incelenmesi gerektiğine dair daha somut bir çerçeve oluşmuştur.

Aşama 2'ye geçildiğinde, önceki taramada CI ya da MSE bakımından öne çıkan 4 aday yapı yeniden ele alınmış ve bu kez 40 epoch süren daha uzun bir eğitim döngüsü uygulanmıştır. Bu yaklaşım, kısa eğitimlerde birbirine çok benzeyen eğilimler gösteren modeller arasındaki ince farkları daha görünür kılmayı amaçlamıştır. Eğitim süresi uzadıkça bu farklar belirginleşmiş, kimi mimariler daha istikrarlı bir çizgi izlerken bazıları erken dönemdeki avantajını sürdürememiştir. Bu ikinci aşama, nihai seçimin çok daha sağlıklı yapılmasını sağlayan bir filtre işlevi görmüştür.

3.3.3 Aşama 1: 10 Epoch Geniş Tarama Sonuçları

Bu bölümde, 256 konfigürasyonun 10 epoch'luk eğitim sonrası test kümesindeki performansından elde edilen en iyi 5 mimari, CI ve MSE metriklerine göre özetlenmektedir.[1]

3.3.3.1 CI'ya Göre En İyi Mimari Konfigürasyonlar (10 Epoch)

Aşama 1 sonunda konkordans indeksine (CI) göre en yüksek skora sahip ilk beş mimari Tablo 3.3'de verilmiştir.

[1] 10 epoch sonuç tablosu burada sadece ablasyon eğilimlerini göstermek amacıyla özetlenmiştir; nihai mimari seçimi Aşama 2'deki 40 epoch sonuçlarına dayanmaktadır.

Tablo 3.3 DAVIS test kümesi üzerinde CI'ya göre en iyi beş mimari (10 epoch).

Sıra	Config No	Pos	Norm	FFN	Cross-Attn	CI ↑ / MSE ↓
1	29	abs	post_rms	geglu	standard	0.8544 / 0.3787
2	13	abs	post_ln	geglu	standard	0.8480 / 0.3744
3	51	abs	pre_rms	relu	gqa	0.8457 / 0.3477
4	25	abs	post_rms	swiglu	standard	0.8448 / 0.3947
5	55	abs	pre_rms	gelu	gqa	0.8443 / 0.3374

Görülen tablo, CI açısından öne geçen tüm mimarilerin ortak bir çizgide buluştuğunu gösterir; mutlak konumsal kodlamayı tercih eden, standart ya da GQA yapısına yaslanan bir çapraz dikkat düzeniyle çalışan ve FFN aşamasında GeGLU, SwiGLU ya da GELU benzeri daha yumuşak geçişli aktivasyonları benimseyen modellerdir bunlar. Config 29 ile Config 13 bu bileşenleri aynı hatta toplar; mutlak konumsal embedding yaklaşımı, GeGLU yapısının akışkan geçişi ve klasik çapraz dikkat düzeni bir araya gelince, her iki model de CI değerlerinde belirgin bir üstünlük sergiler. Bu nedenle Aşama 2'de uygulanacak 40 epoch'luk yeniden eğitim sürecine, adı geçen iki yapı temel aday olarak alınmıştır.

3.3.3.2 MSE'ye Göre En İyi Mimari Konfigürasyonlar (10 Epoch)

Benzer şekilde, 10 epoch sonunda MSE'ye göre en iyi beş konfigürasyon Tablo 3.4'de özetlenmiştir.

Tablo 3.4 DAVIS test kümesi üzerinde MSE'ye göre en iyi beş mimari (10 epoch).

Sıra	Config No	Pos	Norm	FFN	Cross-Attn	MSE ↓ / CI ↑
1	53	abs	pre_rms	gelu	standard	0.3353 / 0.8410
2	37	abs	pre_ln	gelu	standard	0.3355 / 0.8379
3	55	abs	pre_rms	gelu	gqa	0.3374 / 0.8443
4	50	abs	pre_rms	relu	gated	0.3402 / 0.8366
5	39	abs	pre_ln	gelu	gqa	0.3403 / 0.8386

Tablodaki verilerin analizi sonucunda, üst sıralarda konumlanan modellerin tamamına yakınında konumsal embedding yöntemi olarak abs kullanıldığı gözlenmektedir. FFN katmanlarında ise çoğunlukla GELU aktivasyon fonksiyonu öne çıkmakta ve alternatif yöntemlere üstünlük sağlamaktadır. Normalizasyon

yöntemleri açısından bakıldığında da, pre rms ve pre ln gibi pre-norm tekniklerinin ağırlık kazandığı dikkat çekmektedir.

MSE performansı yönünden öne çıkan iki mimari, Config 53 ve Config 37, Aşama 2 kapsamında 40 epoch'luk daha kapsamlı bir eğitim sürecine tabi tutulmak üzere belirlenmiştir.

İlk aşamadaki geniş çaplı değerlendirme göstermiştir ki, mutlak konumsal embedding (`abs`) kullanan, bununla beraber GELU ve GeGLU gibi kapılı veya yumuşak aktivasyon fonksiyonlarına sahip FFN katmanları içeren ve cross-attention mekanizmasını barındıran mimariler, CI ve MSE metriklerinde belirgin biçimde avantaj sağlamaktadır. İkinci aşama olarak adlandırılan 40 epochluk hassas eleme sürecinde Config 29, Config 13 ile birlikte daha önce bahsedilen Config 53 ve Config 37 merkezde yer almıştır.

3.3.4 Aşama 2: 40 Epoch İnce Eleme ve Nihai Mimari Seçimi

İkinci aşamada, önceki aşamada belirlenmiş olan aday mimariler, aynı veri kümesi bölünmesi korunarak bu kez 40 epoch boyunca yeniden eğitime tabi tutulmuştur. Her bir modelin eğitim sürecinde, doğrulama kaybının en düşük olduğu epoch temel alınarak ilgili ağırlıklar kaydedilmiş; test değerlendirmeleri de yalnızca bu kaydedilmiş modeller üzerinden gerçekleştirilmiştir. Bu sayede, başlangıçta yalnızca 10 epoch ile birbirine oldukça yakın performans sergilemiş mimariler, daha uzun eğitim sürecinde ortaya çıkan ayrışmalar doğrultusunda istikrar açısından daha belirgin biçimde karşılaştırılabilir hâle getirilmiştir.

Gerçekleştirilen eğitimler 40 epoch boyunca devam etmiş ve nihayetinde dört farklı model umut vadeden performanslar ortaya koymuştur. Config 29 adıyla tanımlanan ilk modelde konumsal gömme yöntemi olarak `abs` benimsenmiş, normalizasyon tekniği ise `post_rms` biçiminde uygulanmıştır. Standart dikkat mekanizması eşliğinde, FFN katmanları `geglu` aktivasyon fonksiyonuyla yapılandırılmıştır. İkinci sırada gelen ve Config 13 şeklinde isimlendirilen model de

yine konumsal gömmeyi `abs` ile gerçekleştirmiş, fakat normalizasyon tercihini bu kez `post_ln` yönünde kullanmıştır. FFN katmanında da aynı şekilde `geglu` yöntemine yer verilmiş olup, dikkat mekanizması standart formuyla işlem görmüştür. Üçüncü model olan Config 53, `abs` türünde konumsal gömme ile yapılandırılırken, normalizasyon tekniğini farklı bir perspektiften ele alarak `pre_rms` yöntemine odaklanmıştır.

Bu yapılandırmada FFN katmanı `gelu` fonksiyonuyla oluşturulmuş ve standart dikkat mekanizması korunmuştur. Son olarak Config 37 adını taşıyan modelde ise konumsal gömme yine `abs` yöntemiyle gerçekleştirilmiş, normalizasyon tercihi ise `pre_ln` biçiminde yapılmıştır. FFN katmanları yine `gelu` aktivasyonu ile şekillenmiş ve standart dikkat yapısı aynen sürdürülmüştür. Modellerin tamamına ilişkin performans değerleri, hazırlanan tablolar üzerinden okuyucuya aktarılmıştır.

3.3.4.1 40 Epoch Sonrası CI Karşılaştırması

Tablo 3.5 40 epoch eğitim sonrası CI bakımından öne çıkan mimariler.

Sıra	**Config No**	**Pos**	**Norm**	**FFN**	**Cross-Attn**	**CI ↑ / MSE ↓**
1	29	abs	post_rms	geglu	standard	0.8609 / 0.2570
2*	13	abs	post_ln	geglu	standard	**0.8734** / **0.2455**

Tablo 3.5'de görüldüğü üzere, 10 epoch'luk geniş taramada CI açısından ikinci sırada yer alan Config 13, 40 epoch eğitim sonrasında hem CI (0.8734) hem de MSE (0.2455) bakımından Config 29'u geride bırakmış ve en iyi genel performansı sergilemiştir. Tabloda yıldız ($*$) ile işaretlenen Config 13, bu nedenle *nihai mimari* olarak seçilmiştir.

3.3.4.2 40 Epoch Sonrası MSE Karşılaştırması

Aşama 1'de MSE bakımından en iyi iki mimari olarak öne çıkan Config 53 ve Config 37'nin 40 epoch sonrası sonuçları Tablo 3.6'de özetlenmiştir:

Bu iki mimari, 40 epoch sonrasında da MSE ve CI açısından güçlü performans

Tablo 3.6 40 epoch eğitim sonrası MSE bakımından öne çıkan (Aşama 1'den seçilmiş) mimariler.

Sıra	Config No	Pos	Norm	FFN	Cross-Attn	CI ↑ / MSE ↓
1	53	abs	pre_rms	gelu	standard	0.8540 / 0.2874
2	37	abs	pre_ln	gelu	standard	0.8693 / 0.2716

sergilemeye devam etmiştir. Ancak her iki durumda da MSE değerleri (0.2874 ve 0.2716) ve CI skorları, Config 13'ün elde ettiği 0.2455 MSE ve 0.8734 CI değerlerinin gerisinde kalmıştır. Dolayısıyla, hem sıralama duyarlılığı (CI) hem de tahmin hatası (MSE) birlikte değerlendirildiğinde, Config 13 açık biçimde en iyi *genel* mimari olarak öne çıkmıştır.

3.3.4.3 Nihai Mimari

Ablasyon çalışmasının iki aşamalı değerlendirilmesi sonunda, tezde temel alınacak mimari olarak *Config 13* tercih edilmiştir. Bu yapı, protein ve SMILES dizilerinin her birine öğrenilebilir mutlak konumsal kodlama (`pos_type = abs`) uygulanarak başlar. Encoder ile cross-attention bloklarında ise, düzenleyici bir yapı sunan post-norm LayerNorm (`norm = post_ln`) kullanılır. Feed-forward bölümünde, Shazeer ve çalışma arkadaşlarının 2020'de tanımladığı GeGLU tabanlı kapılı ağ (`ffn_type = geglu`) yer alır. Bunun yanında, protein ile ligand dizilerinin birbirine bağlanmasını sağlayan standart çok başlı çapraz-dikkat mekanizması (`attn_type = standard`) mimarinin merkezindeki etkileşimi kurar.

Bu mimari, yaklaşık kırk epoch süren eğitimden sonra DAVIS test kümesinde CI $= 0.8734$ ve MSE $= 0.2455$ değerlerine ulaşmıştır. İlk aşamadaki geniş tarama sürecinde öne çıkan eğilimler, bu sonuçlarla daha açık biçimde doğrulanmıştır. Mutlak konumsal kodlamanın sağladığı düzenli temsil, GeGLU yapısının esnek geçişleri ve çapraz-dikkatin iki diziyi uyumlu biçimde bir araya getirişi, uzun süreli eğitim koşullarında diğer aday yapıların önüne geçmiş ve tezde esas alınacak modeli belirgin biçimde ortaya çıkarmıştır.

3.3.5 Genel Değerlendirme

İki farklı aşamada gerçekleştirilen 4×4×4×4 boyutlu kapsamlı taramanın neticesinde oluşan tablo irdelenmiştir. Başlangıçta daha geniş aralıklı denemeler yapılmış ve ardından uzun soluklu eğitimler gerçekleştirilmiştir. Bu süreçte mutlak konum bilgisini öğrenen modellerin, diğer konum kodlama yöntemlerine göre daha kararlı bir seyir izlediği tespit edilmiştir. RoPE ve ALiBi yöntemlerinin ise ancak belirli koşullarda avantaj sağladığı gözlemlenmiştir; buna karşın dizilerde bulunan yapısal örüntüleri daha istikrarlı biçimde kavrama açısından öğrenilebilir mutlak konum tablolarının üstünlük gösterdiği belirlenmiştir. Eğitim süresi uzadıkça iki yaklaşım arasındaki performans farkının giderek daha açık biçimde belirginleştiği sonucuna varılmıştır.

Normalizasyon yöntemleri gözden geçirildiğinde, başlangıçta RMSNorm kullanan modellerin genelde daha üst sıralarda yer aldığı ortaya çıkmıştır. Bu durumun, kısa eğitimlerde yöntemin modele belli ölçüde denge sağlamasından kaynaklandığı düşünülmüştür. Ancak eğitim süresi uzadıkça tablonun farklılaştığı görülmüştür. Post-norm LayerNorm ve GeGLU birlikte kullanıldığında performansın daha iyi sonuçlar verdiği gözlemlenmiştir. Eğitim süresinin artmasıyla birlikte normalizasyonun etkisinin sabit kalmadığı, aksine modelin daha farklı bir dengeye doğru kaydığı tespit edilmiştir. Böylelikle, son durumda ortaya çıkan model yapısının ilk baştakinden oldukça farklı olduğu belirlenmiştir.

FFN katmanlarında kapılı mekanizmaların yer aldığı modeller ile doğrusal olmayan ilişkilerin daha iyi bir biçimde yakalandığı gözlemlenmiştir. GeGLU ve buna benzer yapıların kullanımı, hata değerlerinde belirgin iyileşmeler sağlarken, modellerin sıralamadaki konumunu da yükseltmiştir. Nihai mimaride GeGLU tercihinin öne çıkması, kullanılan yöntemin problemi kavrayışına iyi uyum gösterdiğini ortaya koymuştur. Klasik ReLU'nun daha katı ve esnek olmayan yapısının ise modeller arasında geriye düşmesine sebep olduğu belirlenmiştir.

Protein ve ligand dizilerinin nasıl bağlandığının daha açık biçimde

gözlemlenebilmesi amacıyla çapraz-dikkat yaklaşımı kullanılmıştır. Elde edilen sonuçların beklenenden daha net olduğu görülmüştür. Gated veya GQA tabanlı yöntemler belirli koşullarda iyi performans sergilemiş olsa da, çalışma boyunca en dengeli sonuçların `standard` çapraz-dikkat (cross-attention) ile elde edildiği tespit edilmiştir.

Elde edilen tüm bu sonuçlar değerlendirildiğinde, ilaç-hedef etkileşimi görevine en uygun düşen mimari netleşmiştir. Bu mimarinin, öğrenilebilir mutlak konum bilgilerini temel aldığı ve post-norm LayerNorm ile kararlılığını sağlamlaştıran bir yapı sergilediği görülmüştür. FFN katmanlarında GeGLU kullanımının ise modelin daha esnek bir dönüşüm gücü elde etmesini sağladığı belirlenmiştir. Dikkat mekanizmasının tüm yapının merkezinde yer aldığı ve standart iki yönlü çapraz dikkat ağının etkileşimleri net biçimde kurduğu gözlemlenmiştir. Tüm deneyler bu çerçevede şekillendirilmiştir. Tezin ilerleyen analizleri de belirlenen bu model üzerinde yürütülmüştür.

3.4 Sonuçların Literatürle Karşılaştırılması

Bu bölümde önerilen TransformGeGLU mimarisinin performansı, çapraz-dikkat düzeneklerine dayalı bir yapı üzerinden DAVIS ve KIBA verisetlerinde yer alan diğer DTA modelleriyle karşılaştırılmaktadır. Amaç, değerlendirmeyi olabildiğince dengeli kılmaktır. Bu nedenle molekül grafı, proteinlerin üç boyutlu yapıları ya da docking skorları gibi ek bilgiler modele özellikle dahil edilmemiştir. Model yalnızca ilaçlara ait SMILES dizilerini ve proteinlere ait FASTA dizilerini kullanır. İlk aşamada, aynı tür veriyi işleyen sekans-temelli modellerle kıyaslama yapılmış ve böylelikle sonuçların daha tutarlı bir çerçevede yorumlanması sağlanmıştır. Sonrasında, daha zengin biyofiziksel ve yapısal bilgi kullanan çok-modlu yöntemler de değerlendirmeye eklenmiş fakat bu yöntemlerin veri çeşitliliği açısından belirgin bir avantaja sahip olduğu ayrıca vurgulanmıştır. Nihai TransformGeGLU test çıktıları ise Tablo 3.10 içinde ayrıntılı biçimde sunulmaktadır.

3.4.1 Yalnız SMILES ve FASTA Kullanan Sekans-Temelli Modellerle Karşılaştırma

Sekans temelli ilaç-hedef etkileşimleri konusunda, molekül ve protein bilgilerini yalnızca diziler üzerinden yorumlayan pek çok farklı model bulunur. Bu modellerden bazıları, evrişimli ağlardan yararlanan DeepDTA (Öztürk ve diğer., 2018). Tekrarlı ve evrişimli ağları birlikte kullanan DeepCDA (Abbasi ve diğer., 2020) ve yarı denetimli üretici ağlarla tasarlanan GANsDTA (Zhao ve diğer., 2020) şeklinde karşımıza çıkar. Transformer mimarisini benimseyen MT-DTI (Shin ve diğer., 2019), dikkat mekanizmasını merkeze alan AttentionDTA (Zhao ve diğer., 2023) ve çok ölçekli dikkat yapısını kullanan MSSAN (Zeng ve diğer., 2022) modelleri de bu alanda sıkça atıf alan çalışmalardandır. Daha yakın dönemde ortaya çıkan InceptionDTA (Seq) (Kalemati ve diğer., 2025) modeli ise farklı ölçeklerdeki Inception bloklarını kullanarak kendisini önceki yöntemlerden ayırır. Bu yaklaşımların hepsi, tezde geliştirilen TransformGeGLU mimarisi gibi sadece basit sekans verilerini temel alır. Yani protein ve ligandın yapısal verileri ya da docking sonuçları gibi ek özelliklere ihtiyaç duymazlar.

Tablo 3.7, DAVIS ve KIBA veri kümelerindeki sekans odaklı yöntemlerin, literatürde belirtilen MSE ve CI performans ölçütleri ile TransformGeGLU modelinin elde ettiği değerleri bir araya getirmektedir. DeepDTA, MT-DTI, GANsDTA ve MSSAN sonuçları Zeng ve meslektaşlarının çalışmasındaki karşılaştırma sundan (Zeng ve diğer., 2022); DeepCDA, AttentionDTA ve InceptionDTA(Seq) performansları ise Hou ve ekibinin çalışmasındaki karşılaştırma sundan derlenmiştir (Hou ve diğer., 2025). Çalışmaların tümü, DAVIS ve KIBA veri kümelerinin standart test ayrımlarını temel aldığından, karşılaştırma açısından yöntemlerin makul ölçüde bir tutarlılık gösterdiği kabul edilmiştir.

Tablo 3.7 incelendiğinde, DAVIS veriseti için önerilen TransformGeGLU mimarisinin MSE bakımından klasik DeepDTA ve GANsDTA'dan daha düşük hata ürettiği (0.261 ve 0.276'ya karşı 0.2359) ve DeepCDA ile MT-DTI'ye oldukça yakın bir seviyede konumlandığı görülmektedir. AttentionDTA, MSSAN ve InceptionDTA(Seq) gibi daha yeni sekans-temelli modeller, özellikle DAVIS üzerinde

Tablo 3.7 Yalnız SMILES ve FASTA kullanan sekans-temelli derin öğrenme modellerinin DAVIS ve KIBA test kümeleri üzerindeki performanslarının önerilen TransformGeGLU modeli ile karşılaştırılması.

Yöntem	**DAVIS MSE ↓**	**DAVIS CI ↑**	**KIBA MSE ↓**	**KIBA CI ↑**
DeepDTA (Öztürk ve diğer., 2018)	0.261	0.878	0.194	0.863
DeepCDA (Abbasi ve diğer., 2020)	0.248	0.891	0.176	0.889
GANsDTA (Zhao ve diğer., 2020)	0.276	0.881	0.224	0.866
MT-DTI (Shin ve diğer., 2019)	0.245	0.887	0.152	0.882
AttentionDTA (Zhao ve diğer., 2023)	0.216	0.893	0.214	0.852
MSSAN (multi-scaled SAN) (Zeng ve diğer., 2022)	0.233	0.890	0.155	0.890
InceptionDTA (Seq) (Kalemati ve diğer., 2025)	0.242	0.897	0.166	0.890
TransformGeGLU (önerilen)	**0.2359**	**0.8786**	**0.1774**	**0.8782**

MSE ve CI bakımından hâlâ en iyi sonuçları veren yöntemlerdir; TransformGeGLU bu modellerin bir miktar gerisinde kalsa da yalnızca ham dizilim girdisiyle rekabetçi bir hataya ulaşmaktadır. CI açısından bakıldığında, TransformGeGLU'nun 0.8786 değeri, DeepDTA ile neredeyse aynı düzeydedir ve sekans-temelli modeller arasında ortalarda konumlanmaktadır; en yüksek CI değerleri AttentionDTA, MSSAN ve InceptionDTA(Seq) civarında yoğunlaşmaktadır.

KIBA veri kümesi ile yürütülen karşılaştırmada, TransformGeGLU yönteminin, daha klasik bir yaklaşımı temsil eden DeepDTA'ya oranla daha kararlı sonuçlar verdiği net olarak dikkat çekmektedir. Özellikle MSE değerinin 0.194'ten 0.1774'e düşüşü, CI değerinin ise 0.863'ten 0.8782'ye yükselişi bu durumu açıkça ortaya koyar niteliktedir.

Halbuki karşılaştırma alanı genişletildiğinde daha karmaşık bir hâl almaktadır. GANsDTA ya da AttentionDTA gibi modellerle yüz yüze geldiğinde TransformGeGLU çoğu zaman benzer bir hata aralığında seyretmekte, kimi koşullarda ise hatayı hafifçe aşağı çekebilmektedir. Fakat bu benzerlik, modern sekans modellerinin tümüyle geride bırakıldığı anlamına gelmez. MT-DTI, MSSAN ve InceptionDTA(Seq) gibi daha yoğun etkileşim temsili kullanan yöntemler, KIBA üzerinde hâlâ daha düşük MSE değerlerine (0.152–0.166 aralığı) ve CI açısından yaklaşık 0.89 düzeyine ulaşarak bir adım önde durmaktadır. Bu sonuçlar, söz konusu modellerin özellikle uzun sekans yapılarındaki ince ilişkileri daha ayrıntılı yakalayabildiğine işaret etmektedir. Kısa bir ifadeyle, TransformGeGLU bu grubun

hemen gerisinde kendine yer bulmaktadır.

Bununla beraber, yalnız SMILES ve FASTA temsilleriyle çalışan sekans odaklı yöntemler arasında TransformGeGLU'nun genel konumunu tanımlamak daha açıklayıcı olmaktadır. DeepDTA gibi erken dönem CNN tabanlı yaklaşımlara kıyasla hem hata hem de sıralı karşılaştırma ölçütlerinde dikkate değer bir iyileşme sunmaktadır. Yarı-denetimli bir yapı içeren GANsDTA ya da LSTM temelli DeepCDA ile yan yana getirildiğinde ise çoğunlukla aynı mertebede bir hata düzeyinde konumlanmaktadır. Buna karşın MSSAN, MT-DTI veya InceptionDTA(Seq) benzeri ileri seviye sekans modelleri hâlâ belirgin biçimde daha iyi sonuçlara ulaşmakta; TransformGeGLU ise bu çizginin biraz gerisinde, fakat onlara yakın sayılabilecek bir performans düzeyinde varlık göstermektedir.

Ek olarak, bu tezde sunulan AUPR değerleri (DAVIS için 0.7243, KIBA için 0.7966; bkz. Tablo 3.10), literatürde DeepDTA için belirtilen sonuçlarla kıyaslandığında (Ranjan ve diğer., 2024), sadece sekans bazlı modeller arasında sınıflandırma odaklı ölçütlerde de TransformGeGLU'nun oldukça güçlü olduğunu ortaya koymaktadır. Literatürdeki çalışmaların tamamı AUPR değerini raporlamadığı için, doğrudan sayısal kıyaslamalardan kaçınılmış, bunun yerine geliştirilen modelin aktif etkileşimleri önceliklendirme yeteneğinin mevcut literatürdeki modellerle benzer seviyede olduğu ifade edilmiştir.

3.4.2 Ek Yapısal / Çok-Modalli Bilgi Kullanan Modellerle Göreli Konum

Sekans bazlı modeller dışında, literatürde ilaçları ve proteinleri daha ayrıntılı betimleyebilen başka yaklaşımlar da ortaya atılmıştır. GraphDTA bunlardan biridir (Nguyen ve diğer., 2021). Bu yöntemde ilaçlar, graf sinir ağlarıyla iki boyutlu molekül grafı biçiminde incelenirken, protein dizileri CNN yardımıyla analiz edilir. GDilatedDTA modeli ise molekül grafında seyreltme işlemiyle farklı yapısal detayları yakalar ve proteinleri yine sekans tabanlı işler (Zhang ve diğer., 2024). Daha farklı bir yöntem olan AttentionMGT-DTA, AlphaFold'un sağladığı protein cep grafını ve ilaç

grafını ortak bir dikkat mekanizması ile bütünleştirir (Wu ve diğer., 2024). Yeni nesil modellerden MEGDTA daha karmaşık bir yaklaşım izleyerek Morgan parmak izi, molekül grafı, protein sekansı ve proteinlerin üç boyutlu kalıntı grafını tek çatı altında toplamayı amaçlar (Hou ve diğer., 2025). Bu sayede daha geniş bir perspektif sunar.

Bu yöntemler, TransformGeGLU'dan *daha fazla* bilgi (2B/3B yapı, parmak izi, cep grafı vb.) kullandıkları için doğrudan "aynı seviyede" bir kıyas yapmak adil değildir; ancak, yalnız SMILES + FASTA ile çalışan bir modelin bu çok-modalli yöntemlere göre yaklaşık olarak nerede konumlandığını görmek faydalıdır. Tablo 3.8'de, bu tür çok-modalli modellerin DAVIS ve KIBA üzerindeki performansları ile TransformGeGLU'nun sonuçları yan yana verilmiştir. GraphDTA, GDilatedDTA, AttentionMGT-DTA ve MEGDTA için MSE ve CI değerleri Hou vd. (Hou ve diğer., 2025)'nden alınmıştır.

Tablo 3.8 Ek yapısal/çok-modalli bilgi kullanan seçili DTA modelleri ile önerilen TransformGeGLU mimarisinin DAVIS ve KIBA test kümeleri üzerindeki karşılaştırması. GraphDTA, GDilatedDTA, AttentionMGT-DTA ve MEGDTA, ilaç ve/veya protein için ek graf/3B bilgi kullanmaktadır.

Yöntem	**DAVIS MSE ↓**	**DAVIS CI ↑**	**KIBA MSE ↓**	**KIBA CI ↑**
GraphDTA (Nguyen ve diğer., 2021)	0.229	0.893	0.139	0.891
GDilatedDTA (Zhang ve diğer., 2024)	0.237	0.885	0.156	0.876
AttentionMGT-DTA (Wu ve diğer., 2024)	0.193	0.891	0.140	0.893
MEGDTA (Hou ve diğer., 2025)	0.196	0.911	0.136	0.886
TransformGeGLU (önerilen)	**0.2359**	**0.8786**	**0.1774**	**0.8782**

Tablo 3.8'den görüldüğü üzere, graf ve 3B cep bilgisi gibi ek yapısal girdiler kullanan modeller, hem DAVIS hem de KIBA verisetlerinde genellikle TransformGeGLU'dan daha düşük MSE ve daha yüksek CI değerlerine ulaşmaktadır. Örneğin GraphDTA ve MEGDTA, KIBA üzerinde MSE'yi 0.136–0.139 bandına kadar düşürmekte; AttentionMGT-DTA ve MEGDTA ise DAVIS için CI'yi 0.89–0.91 seviyesine çıkarabilmektedir. Bu sonuçlar, ek yapısal bilginin modele kazandırdığı ilave öngörü gücünü açıkça yansıtmaktadır.

Buna rağmen, yalnızca ham SMILES ve FASTA dizilerini kullanarak analiz yapan TransformGeGLU modeli, DAVIS veriseti özelinde GraphDTA ve GDilatedDTA gibi yöntemlerin hata değerlerine yakın sonuçlar elde etmiştir (0.2359 karşısında 0.229–0.237 aralığı). KIBA veriseti için durum biraz daha farklı; DeepDTA veya

AttentionDTA gibi yalnız sekansa odaklı modellere kıyasla daha düşük hata seviyesine sahipken, graf ya da üç boyutlu bilgi kullanan ileri düzey yaklaşımların gerisinde kalmıştır. Dolayısıyla TransformGeGLU, veri girişini basitleştirerek, karmaşık temsillere dayalı çok modlu modellere göre hiç de küçümsenmeyecek bir performansa göstermektedir. Böyle bir sonuç, yöntemin giriş verisi açısından oldukça ekonomik ve etkili olduğunun bir göstergesidir.

3.5 Sonuçların Değerlendirilmesi ve Son Modele Karar Verilmesi

Bu bölümde, önceki kısımda sunulan $4 \times 4 \times 4 \times 4$ ablasyon çalışmasının özet sonuçları ve bu sonuçlar ışığında *son model* seçim süreci kısaca değerlendirilmektedir. Amaç, farklı mimari bileşen kombinasyonlarının performansını karşılaştırmalı olarak gösterip, nihai mimarinin nasıl belirlendiğini ve DAVIS ile KIBA verisetlerindeki nihai sonuçları raporlamaktır (Davis ve diğer., 2011, Tang ve diğer., 2014, Öztürk ve diğer., 2018).

3.5.1 Ablasyon Çalışmalarının Özet Değerlendirmesi

Ablasyon çalışmasının Aşama 1 bölümünde farklı mimari seçenekleri test edilmiştir. Bunun için konumsal kodlama (`pos_type`), normalizasyon yöntemi (`norm`), FFN tipi (`ffn_type`) ve çapraz-dikkat türü (`attn_type`) olarak dört tane farklı parametre belirlenmiştir. Her parametre için dört farklı seçenek seçilmiştir ve böylece toplamda $4 \times 4 \times 4 \times 4 = 256$ tane farklı mimari model elde edilmiştir. Bu modellerin hepsi, diğer tüm ayarları değiştirmeden, 10 epoch boyunca eğitilmiştir. Eğitimlerde kullandığımız verileri ve rastgele seçimleri her model için aynı tutulmuştur. Böylece mimarideki değişikliklerin sonuçlar üzerindeki etkisi daha net görülmüştür.

Konumsal kodlama açısından bakıldığında, mutlak konum embedding'i (`abs`) tercih eden modellerin, döner konumsal kodlama (RoPE) veya dikkat skorlarına doğrusal yanlılık getiren ALiBi gibi alternatiflere oranla CI ve MSE değerlerinde

daha tutarlı bir üstünlük sağladığı fark edilmiştir. RoPE ve ALiBi yöntemleri ise bazı durumlarda performanslarını sabit tutmakta zorlanmıştır. Bu çalışmadaki gözlemler, söz konusu yöntemlerin her zaman istikrarlı olmayabileceğini vurgular nitelikte olmuştur.

İleri-beslemeli ağların (FFN) yapılandırması üzerine yapılan değerlendirmelerde kapılı mekanizmalar içeren ve daha esnek davranan aktivasyon fonksiyonları, örneğin GeGLU, SwiGLU ya da GELU, standart ReLU aktivasyonunu geride bırakmıştır. Özellikle GeGLU ve GELU kullanan modeller daha iyi sonuçlar vermiştir. Literatürde genellikle tercih edilen klasik ReLU aktivasyonu burada belirgin biçimde geride kalmıştır. Önceki akademik yayınların çoğuyla da örtüşen bu sonuçlar, GLU tabanlı yapıların Transformer mimarilerinde performansı artırdığını bir kez daha teyit etmiştir.

Protein ve ligand dizileri arasındaki ilişkiyi net bir biçimde ortaya koyan çapraz-dikkat mekanizmaları (`standard`, `gated`, `gqa`), dikkat mekanizması kullanılmayan (`attn_type=none`) modellerin performansını geride bırakmıştır. Bu durumun temel sebebi çok başlıklı dikkat bloklarının protein–ligand bağlanma dinamiklerini doğrudan yakalayabilmesi olabilmektedir.

RMSNorm ve LayerNorm kullanılan konfigürasyonların genel olarak iyi sayılabilecek sonuçlar ürettiği görülmüştür; ancak asıl dikkat çekici bulguların daha özel varyasyonlarda ortaya çıktığı tespit edilmiştir. Özellikle ilaç–protein bağlanma tahmini görevinde `pre_rms`, `post_rms` ve `post_ln` gibi varyantların belirgin biçimde öne çıktığı gözlenmiştir. Daha önce gerçekleştirilen çalışmalarda da bu tür normalizasyon yaklaşımlarının derin modellerin kararlı biçimde eğitilmesine katkı sağladığı ifade edilmiştir. Elde edilen bulgular ile literatürde dile getirilen bu görüşlerin desteklendiği ortaya konmuştur.

Bu eğilimler temel alınarak, Aşama 2 için CI ve MSE bakımından en iyi sonuç veren

dört aday mimari seçilmiştir:

Config 29, Config 13, Config 53, Config 37.

Bu dört konfigürasyon, aynı veri ayrımı üzerinde bu kez 40 epoch boyunca yeniden eğitilmiştir. Her bir model için küçük bir hiperparametre araması yapılarak (örneğin öğrenme oranı, dropout oranı vb.) kendi en iyi hiperparametre seti belirlenmiş, ardından 40 epoch'luk eğitim sonucu en iyi doğrulama kaybını veren epoch'taki ağırlıklar test kümesi üzerinde değerlendirilmiştir.

3.5.2 40 Epoch Karşılaştırmalı Sonuçlar ve Model Seçimi

Aşama 2'de yeniden eğitilen dört aday mimarinin DAVIS test kümesi üzerindeki performansı Tablo 3.9'de özetlenmektedir.

Tablo 3.9 Aşama 2'de 40 epoch yeniden eğitim sonrası DAVIS test kümesi performansları.

Config No	Pos	Norm	FFN	Cross-Attn	CI ↑	MSE ↓
29	abs	post_rms	geglu	standard	0.8609	0.2570
13	abs	post_ln	geglu	standard	**0.8734**	**0.2455**
53	abs	pre_rms	gelu	standard	0.8540	0.2874
37	abs	pre_ln	gelu	standard	0.8693	0.2716

Tablo 3.9 incelendiğinde:

İncelenen tüm konfigürasyonlarda ortak seçim olarak `pos_type=abs` kullanılmıştır, cross-attention bloğunda ise standart yapı tercih edilmiştir ki bu da zaten ilk aşamada görülen eğilimlerle birebir uyuşmaktadır. CI değerinde (0.8734) en yüksek noktaya ulaşan Config 13, aynı anda MSE açısından da (0.2455) en düşük seviyeyi yakalamıştır. Bu durum iki metriği bir arada dengelemesiyle diğer seçeneklere üstünlük sağladığını göstermektedir. FFN yapıları karşılaştırıldığındaysa, GeGLU kullanan modeller (özellikle Config 29 ve Config 13), GELU bazlı olan Config 53 ve Config 37'ye CI değerlerinde net biçimde üstünlük sağlamıştır.

Bu tezde, Config 13 mimarisi TransformGeGLU olarak adlandırılmış ve tüm ileri

deneyler bu yapı temel alınarak gerçekleştirilmiştir.

3.5.3 Nihai Modelin DAVIS ve KIBA Üzerindeki Performansı

Mimari seçimler denemeler sonrasında belirlenmesiyle, TransformGeGLU modeli bu defa DAVIS ve KIBA verileri üstünde daha uzun süreli eğitimlere tabi tutulmuştur. Eğitim sonuçlarını ölçmek için literatürde yaygın kullanılan ortalama karesel hata (MSE), bunun kök değeri olan RMSE ve sıralama duyarlılığını ölçmekte etkin olan Uyum İndeksi (Concordance Index, CI) dikkate alınmıştır. Kesinlik–Geri Çağırım Eğrisi Altındaki Alan (AUPR) değeri de ayrıca hesaplanmıştır (Öztürk ve diğer., 2018, Shah ve diğer., 2025). DAVIS için eğitim uzun tutulmuş, tam 500 epoch sürmüştür bu süreç. KIBA içinse süre kısaltılarak 120 epoch tercih edilmiştir. En düşük doğrulama kaybına ulaşılan epoch belirlendikten sonra, bu aşamadaki ağırlıklar test setinde uygulamalı olarak sınanmıştır.

DAVIS verisetinde TransformGeGLU modelinin 500 epoch süren eğitim sonrası ulaştığı test sonuçları aşağıdaki gibidir: Test MSE değeri 0.2359, RMSE ise 0.4858 olarak hesaplanmıştır. Modelin sıralama performansını gösteren uygunluk indeksi (CI), 0.8786 seviyesine ulaşmıştır. Ayrıca doğruluk–geri çağırım eğrisinin altındaki alan (AUPR) ise 0.7243 değerine ulaşmıştır.

KIBA veri kümesi özelinde mimarinin uyarlanmış şekli olan TransformGeGLU_KIBA varyasyonu, 120 epoch süren eğitim sonrası test kümesinde ölçülen değerler bakımından şu sonuçları ortaya koydu: test MSE 0.1774, RMSE 0.4213, CI 0.8782 ve AUPR 0.7966 olarak gerçekleşmiştir.

Her iki veri kümesinde ulaşılan son metrikler, tercih edilen mimarinin hata temelli ölçütler (MSE ve RMSE) ile sıralama odaklı kriterler (CI ve AUPR) açısından güçlü olduğu kadar tutarlı bir performansa sahip olduğuna işaret etmektedir. Bu karşılaştırmalı sonuçlar Tablo 3.10 kapsamında ayrıntılı biçimde sunulmuştur.

İki aşamadan oluşan ablasyon sürecinde belirginleşen TransformGeGLU mimarisi,

Tablo 3.10 Nihai TransformGeGLU mimarisinin DAVIS ve KIBA test kümelerindeki performansı.

Veriseti	**MSE ↓**	**RMSE ↓**	**CI ↑**	**AUPR ↑**
DAVIS (500 epoch)	0.2359	0.4858	0.8786	0.7243
KIBA (120 epoch)	0.1774	0.4213	0.8782	0.7966

DAVIS ve KIBA veri kümelerinde sergilediği rekabetçi metrikler sayesinde bu tezde önerilen sekans-temelli (SMILES + FASTA) çift-girdili çapraz-dikkat temelli DTA yaklaşımının nihai modeli olarak kabul edilmiştir.

BÖLÜM DÖRT
TARTIŞMA VE SONUÇ

Bu tez kapsamında, küçük molekül ilaç adaylarının SMILES notasyonlarıyla, ilgili hedef proteinlerin FASTA formatındaki dizilerinin dışında herhangi bir ek yapısal girdi kullanmaksızın, ilaç–hedef afinite tahmini problemi, çapraz dikkat mekanizmasına dayalı bir Transformer modeli kullanılarak incelenmiştir. Mevcut literatür derinlemesine irdelendiğinde fark edilmektedir ki, derin öğrenme yöntemlerini kullanan DTA çalışmalarının sayısı son dönemlerde dikkate değer biçimde artmış olsa da, iki temel araştırma boşluğu hâlen açıklığa kavuşturulmamıştır. Bunlardan ilki, Transformer temelli dizisel modeller içerisinde bulunan konum kodlama stratejileri, normalizasyon türleri, FFN yapılarının farklı varyantları ve çapraz dikkat mimarilerinin, model performansına olan katkılarının tutarlı ve kapsamlı bir şekilde değerlendirilmemiş olmasıdır. İkinci açık sorun ise, sadece dizisel veri üzerinden tahmin yapan yöntemlerle dizisel girdilerin yanında üç boyutlu yapısal ya da multimodal ek bilgilerden yararlanan modeller arasındaki karşılaştırmaların, kullanılan veri miktarına göre eşit şartlarda gerçekleştirilmemesidir.

Elde edilen bulgulara göre, bu tezde önerilen TransformGeGLU mimarisi iki aşamalı detaylı bir ablasyon çalışmasıyla seçilmiştir. DAVIS ve KIBA veri kümeleri kullanılarak yapılan testlerde, bu mimari hem hata ölçümlerinde (MSE, RMSE), hem de sıralama yeteneği ölçümlerinde (CI, AUPR) yalnızca dizisel girdi kullanan diğer DTA modelleri ile rekabetçi sonuçlar vermiştir. Modelin giriş verileri oldukça basit tutulmasına rağmen, daha karmaşık üç boyutlu yapı veya çok-modlu verilerle çalışan modellere yakın performans elde edilmiştir. Bu durum, kullanılan veri miktarı düşünüldüğünde oldukça iyi bir verim sağlandığını göstermektedir. Tez kapsamında Transformer modelindeki farklı bileşenlerin tahmin sonuçlarına etkisi açıkça gösterilerek ileride grafik tabanlı veya üç boyutlu yapısal bilgilerin kolayca eklenebileceği hafif ve esnek bir temel model ortaya koyulmuştur.

4.1 Literatürle Karşılaştırma ve Çalışmanın Katkıları

4.1.1 Girdi Türüne Göre Model Sınıfları

Mevcut literatürde bulunan DTA yöntemleri, ele alınan girdi türlerine göre temelde üç ayrı gruba ayrılarak incelenmektedir. İlk kategoriyi oluşturan yöntemler, ilaçları yalnızca SMILES gösterimleri üzerinden, hedef proteinleri ise aminoasit dizileri aracılığıyla tanımlar. Bu kategoride DeepDTA, DeepCDA, GANsDTA, MT-DTI, AttentionDTA, MSSAN ve InceptionDTA(Seq) gibi yöntemler öne çıkmakta ve literatürde geniş kabul görmektedir (Öztürk ve diğer., 2018, Abbasi ve diğer., 2020, Zhao ve diğer., 2020, Shin ve diğer., 2019, Zhao ve diğer., 2023, Zeng ve diğer., 2022, Kalemati ve diğer., 2025).

Bu çalışma kapsamında geliştirilen TransformGeGLU modeli, doğrudan söz konusu ilk kategori içerisinde konumlanmaktadır. İkinci kategoriyi tanımlayan modeller ise sekans verilerinin yanı sıra bu sekanslardan elde edilen domain ve motif yapılarını da bağlamsal öznitelikler şeklinde kullanmaktadır ki WideDTA modeli, bu bağlamda literatürde önemli bir örnek teşkil eder (Öztürk ve diğer., 2019). Dış kaynaklı veritabanlarından sağlanan motif veya domain tabanlı sözlük bilgisi, bu yaklaşımların sekans verilerini anlamsal açıdan zenginleştirmesine olanak tanır ve daha kapsamlı özelliklerle besler.

GraphDTA, GDilatedDTA, AttentionMGT-DTA ve MEGDTA gibi üçüncü gruptaki yöntemler, çok-modlu bir yaklaşımla ilaç molekülünün grafiksel temsilini, proteinlerin üç boyutlu yapısal özelliklerini ve farklı türlerdeki parmak izi bilgilerini sentezler (Nguyen ve diğer., 2021, Zhang ve diğer., 2024, Wu ve diğer., 2024, Hou ve diğer., 2025). Sekans bazlı veriler ise bu modellerde sadece ham hâlde kalmaz; açık yapısal verilerle harmanlanarak temsil uzayının karmaşıklığı katman katman arttırılmıştır.

4.1.2 Sekans-temelli ve Çok-Modalli Yöntemlerle Karşılaştırma

Yalnız sekans girdisi kullanan DeepDTA, DeepCDA, GANsDTA, MT-DTI, AttentionDTA, MSSAN ve InceptionDTA(Seq) gibi yöntemlerle karşılaştırıldığında, TransformGeGLU'nun genel olarak bu grup içinde orta–üst seviyede konumlandığı söylenebilir. DAVIS veri kümesi üzerinde TransformGeGLU'nun MSE $= 0.2359$ ve CI $= 0.8786$ değerleri, özellikle ilk kuşak CNN veya RNN tabanlı yaklaşımlar olan DeepDTA ve GANsDTA'ya kıyasla daha düşük hata ve benzer/rekabetçi CI sağlamaktadır. Buna karşın, MSSAN, AttentionDTA ve InceptionDTA(Seq) gibi daha yeni sekans-temelli Transformer veya multi-scale attention mimarileri, özellikle DAVIS üzerinde hâlâ bir miktar daha düşük MSE ve daha yüksek CI değerlerine ulaşabilmektedir.

KIBA veri kümesindeki tablo görünümünde farklı bir tablo ortaya çıkmaktadır. TransformGeGLU_KIBA modeli, DeepDTA ve GANsDTA gibi ilk nesil yöntemlerle karşılaştırıldığında hatayı aşağı çekerken CI skorunu yükseltmektedir. Ancak MT-DTI, MSSAN ve InceptionDTA(Seq) gibi daha güncel modeller, özellikle MSE metriğinde daha öne çıkmış durumdadır. Buradan anlaşılabileceği gibi, TransformGeGLU performans anlamında sekans bazlı modeller arasında zirveye tam oturmasa da, basit ve kolay yorumlanabilir tasarım avantajıyla dikkat çekmektedir. Performans skalasında, güncel üst düzey modellerin biraz gerisinde kalırken, erken dönem yöntemlerle karşılaştırıldığında çok daha belirgin bir üstünlük sağlamaktadır.

Graf temsilleri, protein cep grafı veya üç boyutlu yapısal bilgi gibi ek modaliteleri kullanan GraphDTA, GDilatedDTA, AttentionMGT-DTA ve MEGDTA gibi çok-modalli yöntemler ile yapılan karşılaştırmalarda ise doğal olarak daha belirgin farklar ortaya çıkmaktadır. Bu modeller, DAVIS ve KIBA veri kümeleri üzerinde genellikle TransformGeGLU'dan daha düşük MSE ve daha yüksek CI değerleri elde etmektedir. Bunun temel nedeni, bu yöntemlerin daha zengin ve doğrudan biyofiziksel bilgiyi (2B/3B yapı, docking skorları, parmak izi temsilleri vb.) modele enjekte etmeleridir. Buna rağmen, yalnızca ham SMILES ve FASTA girdileriyle çalışan TransformGeGLU'nun bazı durumlarda GraphDTA veya GDilatedDTA'nın

DAVIS üzerindeki hata aralığına oldukça yakın sonuçlar verebilmesi, giriş uzayı açısından son derece ekonomik bir tasarımla verimli bir temsil gücüne ulaşılabildiğini göstermektedir.

Sonuç olarak, önerilen TransformGeGLU mimarisi, yalnız SMILES ve FASTA dizilerini girdi olarak alan DTA modelleri içerisinde, hata ve sıralama performansı açısından *güçlü ve dengeli* bir alternatif sunmaktadır. Çapraz-dikkat mekanizmasının ilaç ve protein dizileri arasındaki etkileşimleri doğrudan modellemesi ve GeGLU tabanlı beslemeli ağın doğrusal olmayan ilişkilere esnek uyum sağlayabilmesi sayesinde, ek yapısal bilgiye ihtiyaç duymadan literatürdeki sekans-temelli SOTA yöntemlere oldukça yakın sonuçlar elde edilmektedir. Bu da, ileride graf/3B yapı gibi ek modalliklerin TransformGeGLU mimarisi ile birleştirilmesi durumunda çok-modalli DTA tahmininde daha da güçlü modeller geliştirmek için sağlam bir temel oluşturduğunu göstermektedir.

Yukarıdaki değerlendirmeler ışığında, tez kapsamındaki başlıca katkılar aşağıda özetlenebilir:

Transformer tabanlı DTA modellerinde kullanılan konumsal kodlama, normalizasyon, FFN yapısı ve çapraz dikkat gibi temel parçalar genellikle sadece tahminlere dayanarak seçilmektedir. Çalışmamızda bu dört ana bileşenin DAVIS veri kümesi üzerindeki gerçek etkilerini ölçmek hedeflenmiştir. Bu amaçla $4 \times 4 \times 4 \times 4$ boyutlarında çok geniş bir parametre matrisi hazırlandı. İki aşamalı bir ablasyon analizi yürüterek her parça sistemli şekilde incelenmiştir.

TransformGeGLU olarak adlandırılan yeni bir mimari ise, öğrenilebilir mutlak konum embedding'leri (pos_type = abs), post-norm uygulamalı LayerNorm (norm = post_ln), GeGLU tabanlı feed-forward ağı (ffn_type = geglu) ve klasik çift yönlü, çok başlıklı çapraz dikkat (attn_type = standard) bileşenlerini bir araya getirir. DAVIS ve KIBA veri setlerinde yapılan deneyler, bu mimarinin hata metrikleri (MSE, RMSE) ve sıralama performansı ölçütleri (CI, AUPR) bakımından literatürdeki güncel yaklaşımlarla rekabet edebilecek düzeyde olduğunu ortaya koymuştur.

Bununla beraber TransformGeGLU, yalnızca SMILES ve FASTA dizilerini girdi olarak kullanan sekans temelli yaklaşımlarla karşı karşıya getirilmiş ve erken dönemin yaygın CNN ya da RNN tabanlı yöntemlerinden belirgin biçimde ayrışmıştır. Aradaki fark nettir. Güncel ve daha karmaşık mimarilerle yapılan karşılaştırmalarda ise hata ve sıralama ölçütleri bakımından benzer bir düzey korunmuş, buna karşın daha yalın bir yapı sunması ve karar süreçlerinin izlenebilir olması sayesinde TransformGeGLU alternatiflerine kıyasla ayrıcalıklı bir konum elde etmiştir.

4.2 Sınırlılıklar ve Gelecek Çalışmalar

Mevcut tezin veri giriş sınırları, bilinçli bir kararla yalnızca SMILES ve FASTA dizileri üzerinden kurgulanmıştır. Temel motivasyon, "minimum girdiyle sınırların ne kadar zorlanabileceği" sorunsalını irdelemektir. Moleküler graflar, protein ceplerine dair 3B koordinatlar yahut docking skorları gibi ilave biyofiziksel parametreler denklem dışı bırakılmıştır. Mimari sadelik ve karşılaştırılabilirlik adına atılan bu adım, avantajlı bulunmuştur. Ancak modelin ulaşabileceği performans tavanını da doğal olarak baskılamakta olduğu değerlendirilmiştir. İlerleyen süreçte, burada temelleri atılan TransformGeGLU iskeletinin yapısal modüllerle (2B/3B) zenginleştirilmesi mümkün görülmüştür. Ön-eğitimli protein dil modelleri veya moleküler graf gömmeleri sisteme entegre edilebilecektir. Bu tür takviyelerin, gerek hata metriklerinde gerekse sıralama hassasiyetinde belirgin bir iyileşmeyi beraberinde getireceği öngörülmüştür.

Gelecekteki çalışmalar için bir diğer önemli eksen de modelin açıklanabilirlik boyutudur. Çapraz-dikkat ağı, hangi amino asit–atom çiftlerinin tahmini afinitede daha etkili olduğuna dair doğrudan bir "ısı haritası" üretebilecek niteliktedir; ancak bu tez kapsamında bu başlık yalnızca genel hatlarıyla tartışılmış, detaylı bir açıklanabilirlik analizi yapılmamıştır. Dikkat ağırlıklarının, bilinen bağlanma cepleri, kritik kalıntılar ve yapı temelli farmakofor modelleri ile karşılaştırıldığı sistematik çalışmalar, modelin biyolojik olarak ne öğrendiğini daha iyi anlamaya katkı sağlayacaktır. Böyle bir analiz, gelecekte tasarlanacak açıklanabilir DTA veya

hedef–bileşik tasarım sistemleri için de önemli bir adım olabilir.

4.3 Genel Sonuç

Sistematik bir ablasyon kurgusuyla ele alınan bu çalışmada, çapraz-dikkat eksenli Transformer bileşenleri derinlemesine irdelenmiştir. Ortaya çıkan TransformGeGLU mimarisi; öğrenilebilir mutlak konum kodlaması, GeGLU tabanlı FFN ve standart çok başlı çapraz-dikkat modüllerini tek bir çatıda birleştirmektedir. Literatürü domine eden çok-modalli ve işlem yükü ağır karmaşık yapılarla kıyaslandığında bu model, sadeliğiyle öne çıkmaktadır. Modern sekans tabanlı yöntemlerin hemen ardında yer alsa da, yorumlanabilirlik avantajı ve yalınlık sunmaktadır.

Böylece hem mimari tasarım alanında "hangi Transformer bileşenleri bu görevde gerçekten fark oluşturuyor?" sorusuna ampirik bir yanıt verilmiş, hem de ileride geliştirilecek çok-modalli ve generatif DTA yaklaşımları için güçlü ve hafif bir temel model ortaya konmuştur. Tezin sonuçları, hesaplamalı ilaç keşfi sürecinde dizisel veriye dayalı yöntemlerin hâlâ önemli bir potansiyele sahip olduğunu ve doğru mimari seçimlerle oldukça verimli bir maliyet–performans dengesi kurulabileceğini göstermektedir.

KAYNAKLAR

Abbasi, K., Razzaghi, P., Poso, A., Amanlou, M., Ghasemi, J. B. ve Masoudi-Nejad, A. (2020). Deepcda: Deep cross-domain compound–protein affinity prediction through LSTM and convolutional neural networks. *Bioinformatics*, *36*(17), 4633–4642.

Ainslie, J., Lee-Thorp, J., de Jong, M., Zemlyanskiy, Y., Lebron, F. ve Sanghai, S. (2023). GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, (4895–4901)., Singapore. Association for Computational Linguistics.

Akiba, T., Sano, S., Yanase, T., Ohta, T. ve Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (2623–2631)., New York, NY, USA. ACM.

Bergstra, J. ve Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*, 281–305.

Bergstra, J. S., Bardenet, R., Bengio, Y. ve Kégl, B. (2011). Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*, volume 24, (2546–2554).

Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., Thomas, J., Ullmann, T., Becker, M., Boulesteix, A.-L., Deng, D. ve Lindauer, M. (2023). Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *13*(2), e1484.

Bissantz, C., Kuhn, B. ve Stahl, M. (2010). A medicinal chemist's guide to molecular interactions. *Journal of Medicinal Chemistry*, *53*(14), 5061–5084.

Bjerrum, E., Rastemo, T., Irwin, R., Kannas, C. ve Genheden, S. (2021). PySMILESUtils – enabling deep learning with the SMILES chemical language. *ChemRxiv*. Preprint.

Bolton, E. E., Wang, Y., Thiessen, P. A. ve Bryant, S. H. (2008). Pubchem: Integrated platform of small molecules and biological activities. *Annual Reports in Computational Chemistry*, *4*, 217–241.

Branden, C. I. ve Tooze, J. (1999). *Introduction to Protein Structure* (2 ed.). New York: Garland Science.

Brown, T. B., Mann, B., Ryder, N. ve diğer. (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33.

Cao, D.-S., Zhang, L.-X., Tan, G.-S., Xiang, Z., Zeng, W.-B., Xu, Q.-S. ve Chen, A. F. (2014). Computational prediction of drug–target interactions using chemical, biological, and network features. *Molecular Informatics*, *33*(10), 669–681.

Chen, H., Engkvist, O., Wang, Y., Olivecrona, M. ve Blaschke, T. (2018). The rise of deep learning in drug discovery. *Drug Discovery Today*, *23*(6), 1241–1250.

Chen, L., Tan, X., Wang, D., Zhong, F., Liu, X., Yang, T., Luo, X., Chen, K., Jiang, H. ve Zheng, M. (2020). Transformercpi: Improving compound–protein interaction prediction by sequence-based deep learning with self-attention mechanism and label reversal experiments. *Bioinformatics*, *36*(16), 4406–4414.

Davis, M. I., Hunt, J. P., Herrgard, S., Ciceri, P., Wodicka, L. M., Pallares, G., Hocker, M., Treiber, D. K. ve Zarrinkar, P. P. (2011). Comprehensive analysis of kinase inhibitor selectivity. *Nature Biotechnology*, *29*(11), 1046–1051.

Deng, L., Zeng, Y., Liu, H., Liu, Z. ve Liu, X. (2022). Deepmhadta: Prediction of drug–target binding affinity using multi-head self-attention and convolutional neural network. *Current Issues in Molecular Biology*, *44*(5), 2287–2299.

Devlin, J., Chang, M., Lee, K. ve Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (4171–4186). Association for Computational Linguistics.

DiMasi, J. A., Grabowski, H. G. ve Hansen, R. W. (2016). Innovation in the pharmaceutical industry: New estimates of r&d costs. *Journal of Health Economics*, *47*, 20–33.

D'Souza, S., Prema, K. V., Balaji, S. ve Shah, R. (2023). Deep learning-based modeling of drug–target interaction prediction incorporating binding site information of proteins. *Interdisciplinary Sciences: Computational Life Sciences*, *15*(2), 306–315.

Durrant, J. D. ve McCammon, J. A. (2011). Molecular dynamics simulations and drug discovery. *BMC Biology*, *9*, 71.

Elnaggar, A., Heinzinger, M., Dallago, C., Rihawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., Bhowmik, D. ve Rost, B. (2020). Prottrans: Towards cracking the language of life's code through self-supervised deep learning and high performance computing. *arXiv preprint arXiv:2007.06225*.

Friesner, R. A., Banks, J. L., Murphy, R. B., Halgren, T. A., Klicic, J. J., Mainz, D. T., Repasky, M. P., Knoll, E. H., Shelley, M., Perry, J. K., Shaw, D. E., Francis, P. ve Shenkin, P. S. (2004). Glide: A new approach for rapid, accurate docking and scoring. 1. method and assessment of docking accuracy. *Journal of Medicinal Chemistry*, *47*(7), 1739–1749.

Gaulton, A., Bellis, L. J., Bento, A. P., Chambers, J., Davies, M., Hersey, A., Light, Y., McGlinchey, S., Michalovich, D., Al-Lazikani, B. ve Overington, J. P. (2012). ChEMBL: A large-scale bioactivity database for drug discovery. *Nucleic Acids Research*, *40*(D1), D1100–D1107.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. ve Dahl, G. E. (2017). Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, (1263–1272).

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P. ve Aspuru-Guzik, A. (2018). Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, *4*(2), 268–276.

Gönen, M. ve Heller, G. (2005). Concordance probability and discriminatory power in proportional hazards regression. *Biometrika*, *92*(4), 965–970.

Goodfellow, I., Bengio, Y. ve Courville, A. (2016). *Deep Learning*. Cambridge, MA: MIT Press. http://www.deeplearningbook.org.

Hastie, T., Tibshirani, R. ve Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2 ed.). Springer Series in Statistics. New York, NY: Springer.

Haykin, S. (2009). *Neural Networks and Learning Machines* (3 ed.). Upper Saddle River, NJ: Pearson Education.

He, T., Heidemeyer, M., Ban, F., Cherkasov, A. ve Ester, M. (2017). Simboost: A read-across approach for predicting drug–target binding affinities using gradient boosting machines. *Journal of Cheminformatics*, *9*, 24.

Hendrycks, D. ve Gimpel, K. (2016). Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Hou, Z., Li, Y., Zhai, H., Luo, J., Ding, Y. ve Pan, Y. (2025). MEGDTA: Multi-modal drug–target affinity prediction based on protein three-dimensional structure and ensemble graph neural network. *BMC Genomics*, *26*(1), 738.

Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V. ve Leskovec, J. (2020). Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*.

Huang, K., Xiao, C., Glass, L. M. ve Sun, J. (2021). Moltrans: Molecular interaction transformer for drug–target interaction prediction. *Bioinformatics*, *37*(6), 830–836.

Huang, S.-Y. ve Zou, X. (2010). Advances and challenges in protein–ligand docking. *International Journal of Molecular Sciences*, *11*(8), 3016–3034.

Jiménez, J., Škalič, M., Martínez-Rosell, G. ve De Fabritiis, G. (2018). KDEEP: Protein–ligand absolute binding affinity prediction via 3d-convolutional neural networks. *Journal of Chemical Information and Modeling*, *58*(2), 287–296.

Jones, G., Willett, P., Glen, R. C., Leach, A. R. ve Taylor, R. (1997). Development and validation of a genetic algorithm for flexible docking. *Journal of Molecular Biology*, *267*(3), 727–748.

Jumper, J., Evans, R. ve diğer. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, *596*, 583–589.

Kalemati, M., Zamani Emani, M. ve Koohi, S. (2025). Inceptiondta: Predicting drug–target binding affinity with biological context features and inception networks. *Heliyon*, *11*(3), e42476.

Karplus, M. ve McCammon, J. A. (2002). Molecular dynamics simulations of biomolecules. *Nature Structural Biology*, *9*(9), 646–652.

Kingma, D. P. ve Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kitchen, D. B., Decornez, H., Furr, J. R. ve Bajorath, J. (2004). Docking and scoring in virtual screening for drug discovery: methods and applications. *Nature Reviews Drug Discovery*, *3*(11), 935–949.

LeCun, Y., Bengio, Y. ve Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Li, X. ve Fourches, D. (2021). SMILES pair encoding: A data-driven substructure tokenization algorithm for deep learning. *Journal of Chemical Information and Modeling*, *61*(4), 1560–1569.

Lin, X., Zhao, K., Xiao, T., Quan, Z., Wang, Z. ve Yu, P. S. (2020). Deepgs: Deep representation learning of graphs and sequences for drug–target binding affinity prediction. In *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI 2020)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, (1301–1308). IOS Press.

Liu, T., Lin, Y., Wen, X., Jorissen, R. N. ve Gilson, M. K. (2007). BindingDB: A web-accessible database of experimentally determined protein–ligand binding affinities. *Nucleic Acids Research*, *35*(suppl_1), D198–D201.

Macarron, R., Banks, M. N., Bojanic, D., Burns, D. J., Cirovic, D. A., Garyantes, T., Green, D. V. S., Hertzberg, R. P., Janzen, W. P., Paslay, J. W., Schopfer, U. ve Sittampalam, G. S. (2011). Impact of high-throughput screening in biomedical research. *Nature Reviews Drug Discovery*, *10*(3), 188–195.

Nair, V. ve Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, (807–814).

Nguyen, N.-Q., Jang, G., Kim, H. ve Kang, J. (2023). Perceiver CPI: A nested cross-attention network for compound–protein interaction prediction. *Bioinformatics*, *39*(1), btac731.

Nguyen, T., Le, H., Quinn, T. P., Nguyen, T., Le, T. D. ve Venkatesh, S. (2021). Graphdta: Predicting drug–target binding affinity with graph neural networks. *Bioinformatics*, *37*(8), 1140–1147.

Overington, J. P., Al-Lazikani, B. ve Hopkins, A. L. (2006). How many drug targets are there? *Nature Reviews Drug Discovery*, *5*(12), 993–996.

Öztürk, H., Ozkirimli, E. ve Özgür, A. (2019). Widedta: Prediction of drug–target binding affinity. *arXiv preprint*.

Pahikkala, T., Airola, A., Pietila, S., Shakyawar, S., Szwajda, A., Tang, J. ve Aittokallio, T. (2015). Toward more realistic drug–target interaction predictions. *Briefings in Bioinformatics*, *16*(2), 325–337.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. ve Duchesnay, É. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Press, O., Smith, N. A. ve Lewis, M. (2021). Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv*.

Ranjan, A., Bess, A., Alvin, C. ve Mukhopadhyay, S. (2024). MDF-DTA: A multi-dimensional fusion approach for drug–target binding affinity prediction. *Journal of Chemical Information and Modeling*, *64*(13), 4980–4990.

Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, X., Canny, J., Abbeel, P. ve Song, Y. S. (2019). Evaluating protein transfer learning with TAPE. In *Advances in Neural Information Processing Systems*, volume 32, (9689–9701).

Reimers, N. ve Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, (3982–3992).

Renaud, J.-P., Chari, A., Ciferri, C., Liu, W.-T., Rémigy, H.-W., Stark, H. ve Wiesmann, C. (2018). Cryo-em in drug discovery: achievements, limitations and prospects. *Nature Reviews Drug Discovery*, *17*(7), 471–492.

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J. ve Fergus, R. (2021). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences of the United States of America*, *118*(15), e2016239118.

Rogers, D. ve Hahn, M. (2010). Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, *50*(5), 742–754.

Saito, T. ve Rehmsmeier, M. (2015). The precision–recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, *10*(3), e0118432.

Santos, R., Ursu, O., Gaulton, A., Bento, A. P., Donadi, R. S., Bologa, C. G., Karlsson, A., Al-Lazikani, B., Hersey, A., Oprea, T. I. ve Overington, J. P. (2017). A comprehensive map of molecular drug targets. *Nature Reviews Drug Discovery*, *16*(1), 19–34.

Schwaller, P., Laino, T., Gaudin, T., Bolgar, P., Hunter, C. A., Bekas, C. ve Lee, A. A. (2019). Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction. *ACS Central Science*, *5*(9), 1572–1583.

Shah, P. M., Zhu, H., Lu, Z., Wang, K., Tang, J. ve Li, M. (2025). Deepdtagen: A multitask deep learning framework for drug–target affinity prediction and target-aware drugs generation. *Nature Communications*, *16*(1), 5021.

Shazeer, N. (2020). GLU variants improve transformer. *arXiv preprint*.

Shin, B., Park, S., Kang, K. ve Ho, J. C. (2019). Self-attention based molecule representation for predicting drug–target interaction. In *Proceedings of the 4th Machine Learning for Healthcare Conference*, volume 106 of *Proceedings of Machine Learning Research*, (230–248). PMLR.

Stokes, J. M., Yang, K., Swanson, K. ve diğer. (2020). A deep learning approach to antibiotic discovery. *Cell*, *180*(4), 688–702.e13.

Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B. ve Liu, Y. (2021). RoFormer: Enhanced transformer with rotary position embedding. *arXiv*.

Szklarczyk, D., Santos, A., von Mering, C., Jensen, L. J., Bork, P. ve Kuhn, M. (2016). STITCH 5: Augmenting protein–chemical interaction networks with tissue and affinity data. *Nucleic Acids Research*, *44*(D1), D380–D384.

Tang, J., Szwajda, A., Shakyawar, S., Xu, T., Hintsanen, P., Wennerberg, K. ve Aittokallio, T. (2014). Making sense of large-scale kinase inhibitor bioactivity data sets: A comparative and integrative analysis. *Journal of Chemical Information and Modeling*, *54*(3), 735–743.

Trott, O. ve Olson, A. J. (2010). Autodock vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of Computational Chemistry*, *31*(2), 455–461.

Vamathevan, J., Clark, D., Czodrowski, P., Dunham, I., Ferran, E., Lee, G., Li, B., Madabhushi, A., Shah, P., Spitzer, M. ve Zhao, S. (2019). Applications of machine

learning in drug discovery and development. *Nature Reviews Drug Discovery*, *18*(6), 463–477.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. ve Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, (5998–6008).

Vig, J. (2019). A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, (37–42)., Florence, Italy. Association for Computational Linguistics.

Wallach, I., Dzamba, M. ve Heifets, A. (2015). Atomnet: A deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *arXiv preprint*.

Wang, L., Wu, Y., Deng, Y., Kim, B., Pierce, L., Krilov, G., Lupyan, D., Robinson, S., Dahlgren, M. K., Greenwood, J., Romero, D. L., Masse, C., Knight, J. L., Steinbrecher, T., Beuming, T., Damm, W., Harder, E., Sherman, W., Brewer, M., Wester, R., Murcko, M., Frye, L., Farid, R., Lin, T., Mobley, D. L., Jorgensen, W. L., Berne, B. J., Friesner, R. A. ve Abel, R. (2015). Accurate and reliable prediction of relative ligand binding potency in prospective drug discovery by way of a modern free-energy calculation protocol and force field. *Journal of the American Chemical Society*, *137*(7), 2695–2703.

Weininger, D. (1988). SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, *28*(1), 31–36.

Wu, H., Liu, J., Jiang, T., Zou, Q., Qi, S., Cui, Z., Tiwari, P. ve Ding, Y. (2024). AttentionMGT-DTA: A multi-modal drug-target affinity prediction using graph transformer and attention mechanism. *Neural Networks*, *169*, 623–636.

Yamanishi, Y., Araki, M., Gutteridge, A., Honda, W. ve Kanehisa, M. (2008). Prediction of drug–target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics*, *24*(13), i232–i240.

Yang, K., Swanson, K., Jin, W., Coley, C., Eiden, P., Gao, H., Guzman-Perez, A., Hopper, T., Kelley, B., Mathea, M., Palmer, A., Settels, V., Jaakkola, T., Jensen, K. ve Barzilay, R. (2019). Analyzing learned molecular representations for property prediction. *Journal of Chemical Information and Modeling*, *59*(8), 3370–3388.

Zeng, X., Chen, W. ve Lei, B. (2024). CAT-DTI: Cross-attention and transformer network with domain adaptation for drug–target interaction prediction. *BMC Bioinformatics*, *25*, 141.

Zeng, Y., Chen, X., Peng, D., Zhang, L. ve Huang, H. (2022). Multi-scaled self-attention for drug–target interaction prediction based on multi-granularity representation. *BMC Bioinformatics*, *23*(1), 314.

Zhang, B. ve Sennrich, R. (2019). Root mean square layer normalization. *arXiv preprint*.

Zhang, L., Zeng, W., Chen, J., Chen, J. ve Li, K. (2024). Gdilateddta: Graph dilation convolution strategy for drug target binding affinity prediction. *Biomedical Signal Processing and Control*, *92*, 106110.

Zhao, L., Wang, J., Pang, L., Liu, Y. ve Zhang, J. (2020). GANsDTA: Predicting drug–target binding affinity using GANs. *Frontiers in Genetics*, *10*, 1243.

Zhao, Q., Duan, G., Yang, M., Cheng, Z., Li, Y. ve Wang, J. (2023). Attentiondta: Drug–target binding affinity prediction by sequence-based deep learning with attention mechanism. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *20*(2), 852–863.

Zhao, Q., Zhao, H., Zheng, K. ve Wang, J. (2022). Hyperattentiondti: Improving drug–protein interaction prediction by sequence-based deep learning with attention mechanism. *Bioinformatics*, *38*(3), 655–662.

Zitnik, M., Nguyen, F., Wang, B., Leskovec, J., Goldenberg, A. ve Hoffman, M. M. (2019). Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities. *Information Fusion*, *50*, 71–91.

Öztürk, H., Özgür, A. ve Ozkirimli, E. (2018). Deepdta: Deep drug–target binding affinity prediction. *Bioinformatics*, *34*(17), i821–i829.

EKLER

Ek 1 Model Kodu (TransformGeGLU)

```
##########################################
# I_GeGLU MODEL - FULL SCRIPT (KIBA, Only GeGLU)
# Not: Mimari, veri/loader, metrikler aynidir.
#      Sadece:
#        - Dataset: KIBA (/TransformDTA/kiba.csv)
#        - HPO ile bulunan KIBA hyperparametreleri
##########################################

######################
# [BOLUM 1: Ortam Kurulumu, Drive, Paketler]
######################
print("===== 1) Ortam Kurulumu ve Drive Baglantisi =====")

from google.colab import drive
drive.mount('/content/drive')

!pip install rdkit-pypi -q
!pip install torch -q
!pip install transformers -q
!pip install scikit-learn -q
!pip install pandas -q
!pip install numpy -q
# !pip install scipy -q

import pandas as pd
import numpy as np
import torch
import torch.nn as nn
import torch.optim as optim
import torch.nn.functional as F
import random
import re
import time
from torch.utils.data import Dataset, DataLoader
```

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, average_precision_score
from math import sqrt
from scipy.stats import pearsonr

def set_seed(seed=42):
    random.seed(seed)
    np.random.seed(seed)
    torch.manual_seed(seed)
    torch.cuda.manual_seed_all(seed)

set_seed(42)

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
print("Kullanilan cihaz:", device)

######################
# [BOLUM 2: KIBA Veri Yukleme ve Inceleme]
######################
print("\n===== 2) KIBA Veri Okuma ve Inceleme =====")

# KIBA CSV yolu
file_path = '/content/drive/MyDrive/TransformDTA/kiba.csv'

# CSV dosyasinin ilk satirinda sutun isimleri varsayiliyor
df = pd.read_csv(file_path, sep=',', header=0)

print("Orijinal sutunlar:", df.columns.tolist())

# KIBA label sutunu tespiti
label_candidates = ['KIBA_Score', 'KIBA score', 'KIBA_score', 'KIBA', 'kiba',
    'Label']
label_col = None

# Once direk eslesen isimlere bak
for col in label_candidates:
    if col in df.columns:
```

```
        label_col = col
        break

# Hala bulunamadiysa icinde "KIBA Score" gecen sutunu ara
if label_col is None:
    for col in df.columns:
        if ('kiba' in col.lower()) and ('score' in col.lower()):
            label_col = col
            break

if label_col is None:
    raise ValueError(
        "KIBA skor sutunu bulunamadi. Lutfen CSV sutun adlarini kontrol edip "
        "label_candidates ya da substring mantigini guncelle."
    )

print("Bulunan label sutunu:", label_col)

# Diger sutunlari ortak forma cevir
rename_dict = {
    "CHEMBLID": "ID",
    "ProteinID": "Target",
    "compound_iso_smiles": "SMILES",
    "target_sequence": "Sequence",
}
if label_col != "Label":
    rename_dict[label_col] = "Label"

df.rename(columns=rename_dict, inplace=True)

print("\nRename sonrasi sutunlar:", df.columns.tolist())
print("Label sutunu:", "Label")
print("\nDataFrame ilk 5 satir:")
print(df.head())
print("\nVeri boyutu:", df.shape)
print("Label istatistikleri:")
print(df['Label'].describe())
```

```
108
109 ######################
110 # [BOLUM 3: Train / Val / Test Split]
111 ######################
112 print("\n===== 3) Train-Val-Test Split (KIBA) =====")
113
114 # %80 train, %10 val, %10 test
115 train_df, temp_df = train_test_split(df, test_size=0.2, random_state=42)
116 val_df, test_df   = train_test_split(temp_df, test_size=0.5, random_state=42)
117
118 print("Train:", train_df.shape, "Val:", val_df.shape, "Test:", test_df.shape)
119
120 ######################
121 # [BOLUM 4: Tokenizasyon ve PyTorch Dataset/DataLoader]
122 ######################
123 print("\n===== 4) Tokenizasyon, Dataset, DataLoader =====")
124
125 # Protein tokenizasyonu (karakter duzeyinde)
126 def tokenize_protein(seq):
127     return list(str(seq).strip())
128
129 # SMILES tokenizasyonu (Cl, Br vb. iki harfli ogelerle)
130 _SMILES_REGEX = re.compile(
131     r"\[\w+\]|"       # [NH3+], [O-]
132     r"\%\d{2}|"       # yuzuk kapama %12
133     r"Br|Cl|"         # iki harfli elementler
134     r"Si|Se|Na|Li|Mg|Ca|Zn|Fe|Cu|Mn|Al|Ag|Sn|Hg|Pb|Bi|Ne|He|Ar|Kr|Xe|"
135     r"@@?|"           # stereo
136     r"=|#|"           # baglar
137     r"\(|\)|"         # parantez
138     r"\.|"            # nokta
139     r"\d|"            # rakam
140     r"[A-Za-z]|"      # tek harf element
141     r"\+|\-|\*|\/|\\" # isaretler
142 )
143
144 def tokenize_smiles(smi: str):
```

```
    return _SMILES_REGEX.findall(str(smi).strip())

# Tokenizer test
print(f"\nSMILES Tokenizer Testi:")
test_smiles = "CCCCl"
print(f"  Input: {test_smiles}")
print(f"  Tokens: {tokenize_smiles(test_smiles)}  (Beklenen: ['C','C','C','Cl
    '])")
test_smiles2 = "c1ccccc1Br"
print(f"  Input: {test_smiles2}")
print(f"  Tokens: {tokenize_smiles(test_smiles2)}")

# Vocab olusturma (sadece TRAIN'den)
all_prot_tokens = set()
for seq in train_df['Sequence']:
    all_prot_tokens.update(tokenize_protein(seq))

all_smi_tokens = set()
for smi in train_df['SMILES']:
    all_smi_tokens.update(tokenize_smiles(smi))

prot_special_tokens = ['<pad>', '<unk>', '<cls>', '<sep>']
smi_special_tokens  = ['<pad>', '<unk>', '<cls>', '<sep>']

prot_vocab_list = prot_special_tokens + sorted(list(all_prot_tokens))
smi_vocab_list  = smi_special_tokens + sorted(list(all_smi_tokens))

protein_vocab = {token: idx for idx, token in enumerate(prot_vocab_list)}
smiles_vocab  = {token: idx for idx, token in enumerate(smi_vocab_list)}

print(f"\ n Protein vocab size: {len(protein_vocab)}")
print(f"  SMILES vocab size:  {len(smiles_vocab)}")
print(f"   SMILES vocab'da 'Cl' var mi? {'Cl' in smiles_vocab}")
print(f"   SMILES vocab'da 'Br' var mi? {'Br' in smiles_vocab}")

class DTADataset(Dataset):
    def __init__(self, df, protein_vocab, smiles_vocab,
```

```
                 max_prot_len=1000, max_smi_len=100):
        self.df = df.reset_index(drop=True)
        self.protein_vocab = protein_vocab
        self.smiles_vocab  = smiles_vocab
        self.max_prot_len  = max_prot_len
        self.max_smi_len   = max_smi_len

    def __len__(self):
        return len(self.df)

    def __getitem__(self, idx):
        row = self.df.iloc[idx]
        seq = row['Sequence']
        smi = row['SMILES']
        label = float(row['Label'])

        prot_toks = tokenize_protein(seq)[:self.max_prot_len]
        smi_toks  = tokenize_smiles(smi)[:self.max_smi_len]

        prot_ids = [self.protein_vocab.get(t, self.protein_vocab['<unk>']) for
     t in prot_toks]
        smi_ids  = [self.smiles_vocab.get(t, self.smiles_vocab['<unk>']) for t
     in smi_toks]

        prot_pad_len = self.max_prot_len - len(prot_ids)
        smi_pad_len  = self.max_smi_len - len(smi_ids)

        prot_ids += [self.protein_vocab['<pad>']] * prot_pad_len
        smi_ids  += [self.smiles_vocab['<pad>']] * smi_pad_len

        return {
            'protein_input': torch.tensor(prot_ids, dtype=torch.long),
            'smiles_input':  torch.tensor(smi_ids,  dtype=torch.long),
            'label':         torch.tensor(label,   dtype=torch.float)
        }

# === Optuna'nin KIBA icin en iyi denemesi: batch size 32 ===
```

```
batch_size = 32

# === Optuna'nin KIBA icin en iyi denemesi: max len 800 / 80 ===
train_dataset = DTADataset(train_df, protein_vocab, smiles_vocab,
                           max_prot_len=800, max_smi_len=80)
val_dataset   = DTADataset(val_df,   protein_vocab, smiles_vocab,
                           max_prot_len=800, max_smi_len=80)
test_dataset  = DTADataset(test_df,  protein_vocab, smiles_vocab,
                           max_prot_len=800, max_smi_len=80)

train_loader = DataLoader(train_dataset, batch_size=batch_size, shuffle=True)
val_loader   = DataLoader(val_dataset,   batch_size=batch_size, shuffle=False)
test_loader  = DataLoader(test_dataset,  batch_size=batch_size, shuffle=False)

print(f"train_loader: {len(train_dataset)} ornek, "
      f"val_loader: {len(val_dataset)}, test_loader: {len(test_dataset)}")

######################
# [BOLUM 5: MODEL - Cross-Attention ve FFN (GeGLU)]
######################
print("\n===== 5) Model ve FFN (GeGLU) =====")

class TransformerEncoderBlock(nn.Module):
    """
    Minimal Transformer Encoder Blogu
    - self-attn
    - FFN (GeGLU)
    - key_padding_mask destekli
    """
    def __init__(self, d_model=256, nhead=8, dim_feedforward=1024, dropout
    =0.1):
        super().__init__()
        self.self_attn = nn.MultiheadAttention(embed_dim=d_model, num_heads=
    nhead, batch_first=True)
        self.dropout = nn.Dropout(dropout)
        self.norm1 = nn.LayerNorm(d_model)
        self.norm2 = nn.LayerNorm(d_model)
```

```
251
252         # GeGLU: (XW_v)  GELU(XW_g)  -> out_proj
253         # Parametre esitligi icin h_glu = (2/3) * dim_feedforward
254         h_glu = int(dim_feedforward * 2 / 3)
255         self.linear_v = nn.Linear(d_model, h_glu)
256         self.linear_g = nn.Linear(d_model, h_glu)
257         self.out_proj = nn.Linear(h_glu, d_model)
258         self.gelu = nn.GELU()
259
260     def forward(self, x, key_padding_mask=None):
261         # Self-Attention
262         attn_out, _ = self.self_attn(x, x, x, key_padding_mask=
    key_padding_mask)
263         x = self.norm1(x + attn_out)
264
265         # GeGLU FFN
266         v = self.linear_v(x)
267         g = self.gelu(self.linear_g(x))
268         ff = self.out_proj(self.dropout(v * g))
269
270         x = self.norm2(x + ff)
271         return x
272
273 class CrossAttentionBlock(nn.Module):
274     """ Protein ve SMILES arasinda karsilikli dikkat (mask destekli) """
275     def __init__(self, d_model=256, nhead=8, dropout=0.1):
276         super().__init__()
277         self.cross_attn_p2s = nn.MultiheadAttention(d_model, nhead,
    batch_first=True)
278         self.cross_attn_s2p = nn.MultiheadAttention(d_model, nhead,
    batch_first=True)
279         self.dropout = nn.Dropout(dropout)
280         self.norm_p = nn.LayerNorm(d_model)
281         self.norm_s = nn.LayerNorm(d_model)
282
283     def forward(self, p, s, p_mask=None, s_mask=None):
284         # p -> s
```

```
285         p2, _ = self.cross_attn_p2s(query=p, key=s, value=s, key_padding_mask=
    s_mask)
286         p = self.norm_p(p + self.dropout(p2))
287         # s -> p
288         s2, _ = self.cross_attn_s2p(query=s, key=p, value=p, key_padding_mask=
    p_mask)
289         s = self.norm_s(s + self.dropout(s2))
290         return p, s
291 
292 class CrossAttentionFusionModel(nn.Module):
293     """
294     Cross-Attention tabanli DTA modeli (KIBA).
295     Not: Encoder katmanlarinda FFN = GeGLU (sabit).
296     """
297     def __init__(self,
298                  prot_vocab_size,
299                  smi_vocab_size,
300                  prot_max_len=1000,
301                  smi_max_len=100,
302                  d_model=256,
303                  nhead=8,
304                  num_encoder_layers=2,
305                  num_cross_layers=1,
306                  dim_feedforward=1024,
307                  dropout=0.1,
308                  padding_idx=0):
309         super().__init__()
310 
311         self.padding_idx = padding_idx
312 
313         # Embeddingler
314         self.prot_token_emb = nn.Embedding(prot_vocab_size, d_model,
    padding_idx=padding_idx)
315         self.prot_pos_emb   = nn.Embedding(prot_max_len, d_model)
316 
317         self.smi_token_emb  = nn.Embedding(smi_vocab_size, d_model,
    padding_idx=padding_idx)
```

```
318         self.smi_pos_emb    = nn.Embedding(smi_max_len, d_model)
319
320         # Encoder bloklari (GeGLU FFN iceriyor)
321         self.prot_encoders = nn.ModuleList([
322             TransformerEncoderBlock(d_model, nhead, dim_feedforward, dropout)
323             for _ in range(num_encoder_layers)
324         ])
325         self.smi_encoders = nn.ModuleList([
326             TransformerEncoderBlock(d_model, nhead, dim_feedforward, dropout)
327             for _ in range(num_encoder_layers)
328         ])
329
330         # Cross-Attention katmanlari
331         self.cross_layers = nn.ModuleList([
332             CrossAttentionBlock(d_model, nhead, dropout)
333             for _ in range(num_cross_layers)
334         ])
335
336         # Cikis basligi: ablasyonda sabit tutuyoruz (adil karsilastirma icin)
337         self.fc = nn.Sequential(
338             nn.Linear(d_model*2, 512),
339             nn.ReLU(),
340             nn.Dropout(dropout),
341             nn.Linear(512, 256),
342             nn.ReLU(),
343             nn.Dropout(dropout),
344             nn.Linear(256, 1)
345         )
346
347     def forward(self, prot_in, smi_in):
348         prot_mask = (prot_in == self.padding_idx)
349         smi_mask  = (smi_in  == self.padding_idx)
350
351         p_positions = torch.arange(prot_in.size(1), device=prot_in.device).
    unsqueeze(0)
352         s_positions = torch.arange(smi_in.size(1),  device=smi_in.device).
    unsqueeze(0)
```

```
353
354         p_embed = self.prot_token_emb(prot_in) + self.prot_pos_emb(p_positions
    )
355         s_embed = self.smi_token_emb(smi_in)  + self.smi_pos_emb(s_positions)
356
357         # Encoderlar
358         for enc in self.prot_encoders:
359             p_embed = enc(p_embed, key_padding_mask=prot_mask)
360         for enc in self.smi_encoders:
361             s_embed = enc(s_embed, key_padding_mask=smi_mask)
362
363         # Cross-attention
364         for cross in self.cross_layers:
365             p_embed, s_embed = cross(p_embed, s_embed, p_mask=prot_mask,
    s_mask=smi_mask)
366
367         # Masked mean pooling
368         prot_mask_expanded = prot_mask.unsqueeze(-1).expand_as(p_embed)
369         smi_mask_expanded  = smi_mask.unsqueeze(-1).expand_as(s_embed)
370
371         p_sum = (p_embed * (~prot_mask_expanded)).sum(dim=1)
372         p_cnt = (~prot_mask).sum(dim=1, keepdim=True).clamp(min=1)
373         p_vec = p_sum / p_cnt
374
375         s_sum = (s_embed * (~smi_mask_expanded)).sum(dim=1)
376         s_cnt = (~smi_mask).sum(dim=1, keepdim=True).clamp(min=1)
377         s_vec = s_sum / s_cnt
378
379         out = self.fc(torch.cat([p_vec, s_vec], dim=1)).squeeze(-1)
380         return out
381
382 ######################
383 # [Egitim / Degerlendirme Fonksiyonlari ve Metrikler]
384 ######################
385 def train_one_epoch(model, dataloader, optimizer, criterion, device):
386     model.train()
387     total_loss = 0
```

```
    for batch in dataloader:
        prot_in = batch['protein_input'].to(device)
        smi_in  = batch['smiles_input'].to(device)
        labels  = batch['label'].to(device)

        optimizer.zero_grad()
        outputs = model(prot_in, smi_in)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()

        total_loss += loss.item()
    return total_loss / len(dataloader)

@torch.no_grad()
def evaluate(model, dataloader, criterion, device):
    model.eval()
    total_loss = 0
    preds, trues = [], []
    for batch in dataloader:
        prot_in = batch['protein_input'].to(device)
        smi_in  = batch['smiles_input'].to(device)
        labels  = batch['label'].to(device)
        outputs = model(prot_in, smi_in)
        loss = criterion(outputs, labels)
        total_loss += loss.item()
        preds.extend(outputs.detach().cpu().numpy())
        trues.extend(labels.detach().cpu().numpy())
    avg_loss = total_loss / len(dataloader)
    return avg_loss, np.array(preds), np.array(trues)

def concordance_index(y_true, y_pred):
    n, n_conc = 0, 0
    for i in range(len(y_true)):
        for j in range(i+1, len(y_true)):
            if y_true[i] != y_true[j]:
                n += 1
```

```
                if (y_true[i] > y_true[j] and y_pred[i] > y_pred[j]) or \
                   (y_true[i] < y_true[j] and y_pred[i] < y_pred[j]):
                    n_conc += 1
    return 0 if n == 0 else n_conc / n



# KIBA icin tipik aktiflik esigi  12.1 (DeepDTA vb. calismalarda)
def aupr_score(y_true, y_pred, threshold=12.1):
    labels_bin = np.array([1 if val > threshold else 0 for val in y_true])
    return average_precision_score(labels_bin, y_pred)

######################
# [BOLUM 6: I_GeGLU - KIBA Egitim & Degerlendirme]
######################
print("\n===== 6) I_GeGLU (KIBA): Egitim & Degerlendirme =====")

# === Optuna EN IYI KIBA yapilandirmasi ===
common_cfg = dict(
    prot_vocab_size=len(protein_vocab),
    smi_vocab_size=len(smiles_vocab),
    prot_max_len=800,      # Optuna
    smi_max_len=80,        # Optuna
    d_model=320,           # Optuna
    nhead=4,               # Optuna
    num_encoder_layers=2,  # Optuna
    num_cross_layers=2,    # Optuna
    dim_feedforward=1536,  # Optuna
    dropout=0.00015872938836575257,  # Optuna
    padding_idx=0
)

EPOCHS = 120
LR = 5.217430696340404e-05  # Optuna en iyi deneme
criterion = nn.MSELoss()

name = "I_GeGLU_KIBA"
```

```
462 print(f"\n--- {name} (GEGLU, KIBA) ---")
463 set_seed(42)  # tekrarlanabilirlik
464 model = CrossAttentionFusionModel(**common_cfg).to(device)
465 optimizer = optim.Adam(model.parameters(), lr=LR)
466
467 best_val_loss = float('inf')
468 best_path = f"best_{name}.pt"
469
470 for epoch in range(1, EPOCHS+1):
471     t0 = time.time()
472     train_loss = train_one_epoch(model, train_loader, optimizer, criterion,
    device)
473     val_loss, val_preds, val_trues = evaluate(model, val_loader, criterion,
    device)
474     val_rmse = sqrt(mean_squared_error(val_trues, val_preds))
475     dt = time.time() - t0
476     print(f"[{name}] Epoch {epoch:03d}/{EPOCHS} "
477           f"TrainLoss={train_loss:.4f} ValLoss={val_loss:.4f} ValRMSE={
    val_rmse:.4f} ({dt:.1f}s)")
478     if val_loss < best_val_loss:
479         best_val_loss = val_loss
480         torch.save(model.state_dict(), best_path)
481         print(f"  -> Best model saved to {best_path}")
482
483 # Test: en iyi checkpoint ile
484 print("\nEn iyi validasyon loss'a sahip model yukleniyor ve TEST setinde
    degerlendiriliyor...")
485 model.load_state_dict(torch.load(best_path, map_location=device))
486 test_loss, test_preds, test_trues = evaluate(model, test_loader, criterion,
    device)
487 test_rmse = sqrt(mean_squared_error(test_trues, test_preds))
488 test_ci   = concordance_index(test_trues, test_preds)
489 test_aupr = aupr_score(test_trues, test_preds, threshold=12.1)
490
491 print(f"\n[{name}] Test MSE:  {test_loss:.4f}")
492 print(f"[{name}] Test RMSE: {test_rmse:.4f}")
493 print(f"[{name}] Test CI:   {test_ci:.4f}")
```

```
print(f"[{name}] Test AUPR: {test_aupr:.4f}")

print("TransformGeGLU (KIBA) degerlendirmesi tamamlandi.")
```

Kod 1: TransformGeGLU Modeli Kodu