TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

TOPLU ÖĞRENME İLE İLAÇ KOMBİNASYONLARININ SİNERJİ SKOR TAHMİNİ

YÜKSEK LİSANS TEZİ

Işıksu EKŞİOĞLU

Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Mehmet TAN

NİSAN 2020

Fen Bilimleri Enstitüsü Onayı

.....
Prof. Dr. Osman EROĞUL
Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onaylarım.

.....
Prof. Dr. Oğuz ERGİN
Anabilimdalı Başkanı

TOBB ETÜ, Fen Bilimleri Enstitüsü'nün 171111009 numaralı Yüksek Lisans Öğrencisi **Işıksu EKŞİOĞLU** 'in ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı **"TOPLU ÖĞRENME İLE İLAÇ KOMBİNASYONLARININ SİNERJİ SKOR TAHMİNİ"** başlıklı tezi **20.04.2020** tarihinde aşağıda imzaları olan jüri tarafından , kabul edilmiştir.

Tez Danışmanı:	**Doç. Dr. Mehmet TAN** TOBB Ekonomi ve Teknoloji Üniversitesi	.....
Jüri Üyeleri:	**Prof. Dr. Tolga CAN (Başkan)** Orta Doğu Teknik Üniversitesi	.....
	Doç. Dr. Osman ABUL TOBB Ekonomi ve Teknoloji Üniversitesi	.....

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, alıntı yapılan kaynaklara eksiksiz atıf yapıldığını, referansların tam olarak belirtildiğini ve ayrıca bu tezin TOBB ETÜ Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırlandığını bildiririm.

Işıksu EKŞİOĞLU

ÖZET

Yüksek Lisans Tezi

TOPLU ÖĞRENME İLE İLAÇ KOMBİNASYONLARININ SİNERJİ SKOR TAHMİNİ

Işıksu EKŞİOĞLU

TOBB Ekonomi ve Teknoloji Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Mehmet TAN

Tarih: Nisan 2020

Kanser gibi ortaya çıkış sebebi birden fazla genetik ve çevresel nedene bağlı olan kompleks hastalıkların tedavisinde son zamanlarda en çok tercih edilen yöntem; birden fazla ilacın birarada kullanıldığı politerapi (kombinasyonel terapi) yöntemidir. Eğer bir ilaç kombinasyonunun, herhangi bir hastalığa sahip hücre hattına olan etkisi, kombinasyondaki ilaçların tek başına uygulanmasıyla elde edilen etkilerin toplamından fazlaysa, bu ilaç kombinasyonuna sinerjik ilaç kombinasyonu denir. Son zamanlarda bu alanda yapılan çalışmalarda, yapay öğrenme yöntemlerinin sinerjik ilaç kombinasyonları nı belirlemede zaman,kaynak kullanımı vs. gibi birçok açıdan verimlilik sağladıkları gözlemlenmştir.

Bu tez çalışması iki bölümden oluşmaktadır. İlk bölümde farklı ilaç gösterimleriyle oluşturduğumuz veri kümelerinin, ilaç kombinasyonlarının sinerjilerinin derecelerini gösteren sinerji skorlarının tahminine olan etkileri incelendi. Kullandığımız ilaç gösterim lerinden bazıları sinerji skoru tahmini için ilk defa kullanılan verilerdir. Bu aşamada oluşturduğumuz veri kümeleri ile yapay öğrenme modellerinden elde edilen tahminler birleştirilerek kapsamlı bir onkoloji veri kümesindeki sinerji skorlarının tahmini için literatürdeki en iyi sonuçlar elde edildi.

İkinci bülümde, ilaç-kanserli hücre hattı ikilileri için bir yapay öğrenme modelinin tahmin ettiği sinerji skorlarını en iyileyecek ikinci ilaçlar (moleküller) oluşturulmaya

çalışıldı. Bu amaç için varyasyonel oto kodlayıcı ve gradyan çıkış yapay öğrenme yöntemlerinden yararlanıldı. Bu çalışmanın sonucunda en iyilenen sinerji skoruna yakın skorlar veren moleküllere, belirli bir oranın üzerinde benzeyen moleküllerin oluşturulduğu gözlemlendi.

Anahtar Kelimeler: Çizge sinir ağı, Oto-kodlayıcı, Makine öğrenmesi, Derin öğrenme, İlaç kombinasyonları sinerji skoru tahmini, Molekül tasarımı, Öznitelik önem analizi.

ABSTRACT

Master of Science

DRUG COMBINATIONS' SYNERGY SCORES PREDICTİON BY ENSEMBLE LEARNING

Işıksu EKŞİOĞLU

TOBB University of Economics and Technology
Institute of Natural and Applied Sciences
Department of Computer Engineering

Supervisor: Assoc. Prof. Mehmet TAN

Date: April 2020

Recently, the most preferred method in the treatment of complex diseases such as cancer, the origin of which is due to more than one genetic and environmental causes, is polytherapy (combination therapy). It is a method of where more than one drug is used together. If the effect of a drug combination on the cell line with any disease is greater than the sum of the effects achieved by applying the drugs in the combination alone, this drug combination is called a synergistic drug combination. In recent studies in this field, It has been observed that machine learning methods provide efficiency for determining synergistic drug combinations in many aspects such as time, resources, etc. This thesis consists of two parts. In the first part, the effects of data sets that we created with different drug representations on the estimation of synergy scores which show the degree of synergism of drug combinations were examined. Some of the drug representations used for the first time for synergy score estimation. The best results in the literature were obtained for the estimation of synergy scores in a comprehensive oncology dataset by combining machine learning predictions' for these datasets.
In the second part, we tried to create second drugs (molecules) for drug-cancer cell line pairs that would optimize synergy scores predicted by an artificial learning model. For this purpose, variational autocoder and gradient ascent methods were used. As a result of this study, it has been observed that, machine learning methods can create molecules

that are similar with the molecules that give scores close to the synergy scores that are optimized.

Keywords: Graph neural network, Autoencoder, Machine learning, Deep learning, Drug combinations' synergy scores prediction, Molecule generation, Feature importance analysis.

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren hocam Doç. Dr. Mehmet TAN'a, kıymetli tecrübelerinden faydalandığım TOBB Ekonomi ve Teknoloji Üniversitesi Bilgiayar Mühendisliği Bölümü öğretim üyelerine, eğitimim boyunca bana burs veren TOBB Ekonomi ve Teknoloji Üniversitesi'ne ve destekleriyle her zaman yanımda olan aileme ve arkadaşlarıma çok teşekkür ederim.

İÇİNDEKİLER

Sayfa

ÖZET . . . **iv**
ABSTRACT . . . **vi**
TEŞEKKÜR . . . **viii**
İÇİNDEKİLER . . . **ix**
ŞEKİL LİSTESİ . . . **xi**
ÇİZELGE LİSTESİ . . . **xiii**
KISALTMALAR . . . **xv**
1. GİRİŞ . . . **1**
2. ÖN BİLGİ . . . **5**
2.1 Tam bağlı yapay sinir ağları . . . 5
2.2 Oto-kodlayıcılar . . . 7
2.3 Gradyan arttırma . . . 7
2.4 Gradyan iniş ve çıkış . . . 8
2.5 Elastik ağ . . . 10
2.6 Rastgele ağaç . . . 10
2.7 SHAP(SHapley Additive exPlanations) değerleri . . . 11
3. İLAÇ KOMBİNASYONLARININ SİNERJİ SKORU TAHMİNİ . . . **13**
3.1 İlgili çalışmalar . . . 13
3.2 Veri kümeleri . . . 16
3.2.1 Öznitelikler . . . 18
3.2.2 Veri kümelerilerine uygulanan ön işlemler . . . 24
3.3 Sinerji skoru tahmini . . . 24
4. SİNERJİ SKORU OPTİMİZASYONU . . . **27**
4.1 İlgili çalışmalar . . . 27
4.2 Veri kümesi . . . 28
4.2.1 Öznitelikler . . . 29
4.2.2 Veri kümelerilerine uygulanan ön işlemler . . . 35
4.3 Sinerji skoru optimizasyonu . . . 35
4.3.1 Sinerji skoru optimizasyonu amacı, girdi ve çıktıları . . . 35
4.3.2 Sinerji skoru optimizasyonu için izlenilen yöntem . . . 36
5. DENEY SONUÇLARI . . . **39**
5.1 Çapraz doğrulama ve istatistiksel testler . . . 39
5.2 Sinerji skoru tahmin deneyleri . . . 40
5.2.1 Model tahminlerinin birleştirilmesi . . . 41
5.2.2 Sinerji skoru tahmin deneyleri sonuçları . . . 42
5.3 Karakteristik yönelim ilaç gösteriminin öznitelik analizi . . . 44

5.4 Sinerji skoru optimizasyonu sonuçları 46
6. DEĞERLENDİRME VE GELECEK ÇALIŞMALAR 55
KAYNAKLAR 57
ÖZGEÇMİŞ 63

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1: Tam bağlı yapay sinir ağı örneği 5
Şekil 2.2: Bir yapay nöronun girdi vektörüyle beslenmesi[14] 6
Şekil 2.3: Gradyan arttırma çalışma akışı: mavi noktalar mode tarafından tahmin edilmeye çalışılan değerleri, kırmızı grafik gradyan arttırma modelinin tahminini, yeşil noktalar ise herbir iterasyondaki basit yapıdak yapay öğrenme modellerinin hatasını gösterir. [17] 9
Şekil 2.4: Rastgele ağaç modelinin eğitimi [20][21] 11
Şekil 3.1: Veri Kümelerinin Yapısı 17
Şekil 3.2: Birinci Aşamada İzlenilen Genel Yöntem Örneği[20][7][44] 18
Şekil 3.3: Molekül vektörlerinin oluşturulması[43] 22
Şekil 3.4: Çizge yapay sinir ağının uçtan uca öğrenme ile ilaç gösterimi oluşturması 23
Şekil 4.1: İlaçların JTVAE Gösteriminin Oluşturulması 29
Şekil 4.2: JTVAE İlaç Gösterimleriyle Sinerji Skoru Optimizasyonu 36
Şekil 5.1: İlaç Kombinasyonları ile Çapraz Doğrulama[7] 40
Şekil 5.2: En iyi performans gösteren ilk beş modelin birleşimi 42
Şekil 5.3: CD ilaç gösterimi için TBYSA tarafından belirlenen genlerin analizi . 45
Şekil 5.4: CD ilaç gösterimi için Gradyan Arttırma tarafından belirlenen genlerin analizi 46
Şekil 5.5: Sinerji skoru en iyileme izlenilen yöntem özeti 52
Şekil 5.6: Gradyan çıkış sonucunda oluşan ve yakınlaşılan SMILES dizileri . . 53
Şekil 5.7: İlaç-hücre hattı ikilileri için kullanılan veri kümesinden daha sinerjik skorlar veren ve gradyan çıkarma işlemiyle oluşturulan SMILES dizileri . 54

ÇİZELGE LİSTESİ

Sayfa

Çizelge 5.1: Veri kümesi-model kombinasyonlarının parametre optimizasyonu sonuçları 41
Çizelge 5.2: Veri kümesi-model kombinasyonlarının ortalama hata karesine göre çapraz doğrulama sonuçları 44
Çizelge 5.3: Veri kümesi-model kombinasyonlarının Pearson korelasyonuna göre çapraz doğrulama sonuçları 44
Çizelge 5.4: Gürültü eklenerek istenilen SMILES dizilerine yaklaşılan kombinasyonlar 51
Çizelge 5.5: Benzer ilaçlarla istenilen SMILES dizilerine yaklaşılan kombinasyonlar 51

KISALTMALAR

GRU : Gated Recurrent Unit
JTVAE : Junctional Tree Variational Autoencoder
GNN : çizge yapay sinir ağı kullanılarak oluşturulan ilaç gösterimi
CD : Karakteristik Yönelim
CHEM : ilaçlarım fiziksel ve kimyasal özelliklerini gösteren tanımlayıcılar
GNNR : GNN ilaç gösterimleriyle oluşturulan veri kümesinin adı
CDR : CD ilaç gösterimleriyle oluşturulan veri kümesinin adı
CHEMR : CHEM ilaç gösterimleriyle oluşturulan veri kümesinin adı
RA : Rastgele Ağaç(Random Forest)
GA : Gradyan Arttırma(Gradient Boosting)
ELAS.A : Elastik Ağ(Elastic Net)
MOE : Molecular Operating Environment
SHAP : Shapley Additive Explanations
SMILES : simplified molecular-input line-entry system
ECFP : Extended-Connectivity Fingerprints
ZIP : Zero Interaction Potency

1. GİRİŞ

Kanser; belirli doku ve organlarda hücrelerin normalden hızlı ve fazla bir şekilde bölünerek, vücuttaki işleyişi bozmasından kaynaklanan hastalıktır. Bu hastalığın çevre, genetik, yaşam tarzı (obezite, hareketsizlik, alkol, sigara vs.) gibi birçok farklı sebebi bulunmaktadır.Kanserin, kontrol dışı hücre bölünmesinin başladığı doku ve organa göre değişen yüzden fazla farklı çeşidi vardır.

Yüz doksan beş ülkeden 1980'den 2015'e kadar toplanan verilere göre kanserin sebep olduğu ölümler, tüm ölümlerin %15.7'sini oluşturmaktadır[1]. Dolayısıyla, bu kadar yaygın olup, farklı birçok sebebi ve çeşidi olan bir hastalığıa her açıdan verimli bir çözüm bulmak önemli bir problemdir. Kanseri tedavi etmek için ameliyat, radyoterapi, hormon tedavisi gibi farklı yöntemler kullanılmaktadır. Bu tedavi çeşitlerinden politerapi (kombinasyonel terapi), kanser gibi birçok sebebi olan bir hastalığın oluşma sebeplerine farklı ilaçları bir araya getirerek aynı anda çözebildiğinden, diğer yöntemlere göre daha verimli bir çözüm sunmaktadır. Aynı zamanda bu tedavi yönteminin kanserli hücrelerin, ilaçlara olan dirençlerini azalttığı gözlemlenmiştir[2].

Bir ilaç kombinasyonunun, bir kanserli hücre için ne kadar etkili olacağı sinerji skoruna bakılarak anlaşılabilir. Bu skor şu şekilde hesaplanır; kombinasyondaki ilaçların farklı dozlarda bir arada kullanılarak, kanserli hücre hattından alınacak tepki (ölüm oranı) bazı matematiksel modeller kullanılarak tahmin edilir; daha sonra laboratuvar ortamında kombinasyondaki ilaçların farklı dozlarda bir arada kullanılarak, kanserli hücre hattından alınan tepkiler elde edilir. Tahmin edilen tepkiler ile deneysel ortamdan alınan tepkiler arasındaki fark bize sinerji skorunu verir. Bir ilaç kombinasyonu için alınan deneysel tepkiler, tahmin edilen tepkilerden fazlaysa ilaç kombinasyonuna sinerjik denir, tam tersi bir durum geçerliyse; ilaç kombinasyonuna antagonostik denir[3]. (Loewe[4], Bliss[5], ZIP[6] hücre hattının ilaç kombinasyonuna tepkisini tahmin etmek için kullanılan matematiksel modellere örnek olarak verilebilir.)

Yukarıda bahsedilen prosedürden anlaşılabileceği gibi, bir ilaç kombinasyonunun sinerjik olup olmadığını anlamak veya bir kombinasyonun sinerji skorunu hesaplamak zaman ve kaynak tüketen bir işlemlerdir. Bu sebepten dolayı yapay öğrenme yöntemleri, ilaç kombinasyonlarının sinerjisi ile ilgili çalışmalar için; herhangi bir laboratuvar deneyi gerektirmedikleri ve matematiksel modeller gibi herhangi bir hipoteze dayanarak

tahmin yapmadık için zaman, kaynak ve doğruluk açısından verimlilik sağlarlar. Son beş yılda DeepSynergy[7] ve TreeCombo[8] gibi yapay öğrenme yöntemleri kullanılarak yapılan çalışmalarla sinerji skoru tahmini için literatürdeki en iyi sonuçlar elde edilmiştir.

Bu tez çalışmasında ilk olarak, DeepSynergy[7] ve TreeCombo[8] çalışmalarıyla aynı onkoloji veri kümesi kullanılarak sinerji skoru tahmini yapıldı. DeepSynergy[7] ve TreeCombo[8] çalışmalarında olduğu gibi sinerji skoru tahmini için gene yapay öğrenme modellerinden faydalanıldı fakat, bu çalışmalardan farklı olarak onkoloji veri kümesindeki ilaç kombinasyonlarının öznitelikleri için, daha önce sinerji skoru tahmini için kullanılmamış farklı ilaç gösterimleriyle çalışıldı. Bu ilaç gösterimleri biraraya getirilerek [7] ve [8] çalışmalarından alınan sonuçlar geliştirilmeye çalışıldı. Buna ek olarak kullandığımız bazı ilaç gösterimleri için SHAP(SHapley Additive exPlanations)[9] değerleri kullanılarak öznitelik analizi uygulandı. Yapılan öznitelik analizi sonucunda önemli bulunan öznitelikler literatürde araştırılarak, kullandığımız ilaç gösteriminin sinerji skoru tahmini için ne kadar uygun bir gösterim olduğu belirlenip, yapılan sinerji skor tahminlerinin öznitelik değerlerine göre değişimi gözlemlendi.

Bu tez çalışmasında ikinci olarak sinerji skor tahmini için eğitilmiş bir yapay öğrenme modelinin tahminini en iyileyen moleküller (ilaçlar) oluşturuldu. İlaç oluşturulması veya tasarımı; belirli hastalığa sebep olabilecek proteinleri hedef alabilecek ilaçların yapısını belirleyip, bu yapılara sahip ilaçları (organik molekülleri) oluşturmaktır[10]. Tasarlanacak ilaçların yapısı belirlenirken; hedef alınan proteinlerle etkileştiği bilinen ilaçların ortak yapılarına bakılır veya hedef alınan proteinlerin üç boyutlu yapısına göre, bu proteinle etkileşime geçecek ilaç yapıları çıkarılır[11][12]. Yapay öğrenme modelleri, sinerji skoru tahmininde olduğu gibi, bu alanda da klasik yöntemlere göre zaman ve kaynak açısından daha verimli bir çözüm sağlıyor. Aynı zamanda diğer yöntemlere göre daha çeşitli moleküller oluşturup, daha geniş bir uzayı keşfedebilirler. Son beş yılda, özellikle bir yapay öğrenme çeşidi olan derin öğrenme teknikleri kullanılarak, herhangi bir moleküler özelliği en iyileyen moleküller oluşturulmaktadır[13]. Bahsedilen derin öğrenme yöntemlerini referans alarak yaptığımız çalışmalar ile literatürde ilk defa yapay öğrenme modelleriyle sinerji skorunu en iyileyen moleküller oluşturuldu. Daha sonra, oluşturulan moleküllerin en iyilenen sinerji skorunu verip veremeyeceği doğrulanmaya çalışıldı.

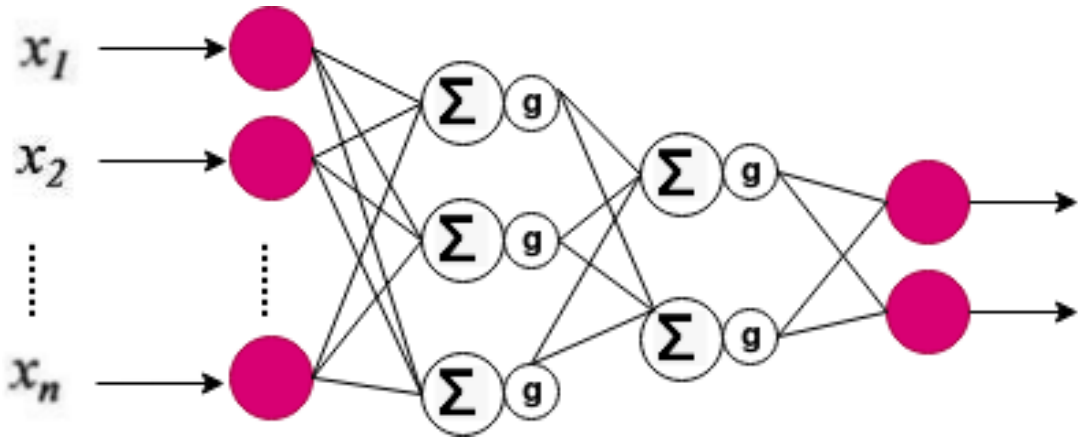
Bahsedilen bu çalışmalarımızı daha detaylı anlatacağımız tez çalışması şu şekilde düzenlendi; Bölüm 2'de tez çalışmasında kullanılan yapay öğrenme yöntemleri hakkında temel bilgiler verilmiştir. Bölüm 3'de ilaç kombinasyonlarının sinerji skor tahmini çalışmalarında, literatürde izlenilen yöntemlerden ve bizim geliştirdiğimiz yöntemin çalışma akışı ile yöntemimizle kullanmak için oluşturulan veri kümelerinden bahsedilmiş tir. Bölüm 4'te sinerji skoru en iyileme ve molekül oluşturma çalışmalarımız için

kullanılan veri kümelerinden ve geliştirilen yöntemden söz edilmiştir. Bölüm 5'te tezin iki aşaması için de yapılan deney ve analizlerin detayları, herbir deney ve analiz sonuçları ile gelecek çalışmalara yer verilmiştir.

2. ÖN BİLGİ

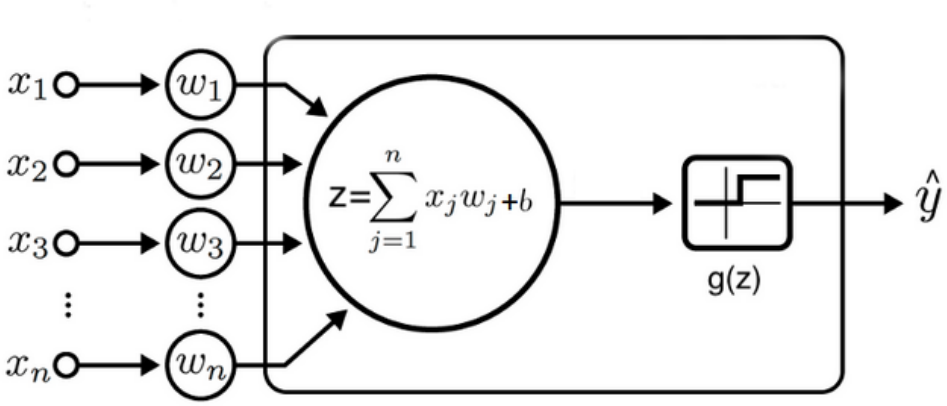
2.1 Tam bağlı yapay sinir ağları

Beyin hücrelerine nöron denir. Nöronlar arasındaki bilgi alışverişi elektrik sinyalleriyle ve sinaps denilen uzantılarla yapılır. Bir nöron, diğer nöronlardan belirli bir değerin üzerinde sinyal aldığı zaman, topladığı bilgileri diğer nöronlara yollar. Hayvan beyinlerin de gerçekleşen bu olaydan esinlenerek Warren McCulloch ve Walter Pitts, 1943 yılında ilk yapay sinir modelini geliştirmişlerdir. Bu yapay sinir modelinin, ikilik tabanda değer alan bir veya birden fazla girdisi olur. Bu modelin, gene ikilik tabanda değer alabilen bir çıktısı bulunmaktadır. Bu yapay sinir modeli, biraz daha geliştirilerek 1957 yılında Frank Rosenblatt en basit yapay sinir ağı modelini (doğru algoritması) geliştirmiştir. İlk yapay sinir modelinden farklı olarak, bu yapay sinir ağında kullanılan sinir modelinin girdileri herhangi bir sayısal değer alabilirler. Aynı zamanda bu yapay sinirin çıktısı, girdilerin ağırlıklı toplamıdır. Geliştirilen bu modelin basit bir XOR problemi çözememe gibi eksiklikleri vardır. Fakat kullanılan nöron ve katman sayısı arttırılarak, doğru algoritmasının bu kısıtlarının kolayca aşılabileceği görülmüştür. Katman sayısı ve nöron sayısını arttırmak derin yapay sinir ağı modellerini ortaya çıkarmıştır. Tam bağlı yapay sinir ağı, bir derin yapay sinir ağı çeşididir. Bu yapay sinir ağı türünde herbir katmanda bulunan herbir nöron, bir sonraki katmanda bulunan nöronların tamamına bağlıdır (Şekil 2.1).



Şekil 2.1: Tam bağlı yapay sinir ağı örneği

Tam bağlı yapay sinir ağının eğitim aşamasında, yapay sinir ağı girdi olarak gelen veri ile beslenir (Şekil 2.2). Daha sonra yapay sinir ağı, bu veri için yaptığı hataya göre geri beslenir. Geri beslenme aşamasında herbir nöronun yapılan hataya ne kadar katkı sağladığı hesaplanır. Bu hesaplamalara göre, en son katmandan başlanarak, girdi katmanına kadar, tüm nöronların ağırlıkları güncellenir. Aşağıda tek bir yapay nöronun nasıl beslenip, yapılan hataya göre ağırlıkların, geri beslenme ile nasıl güncellendiği gösterilmiştir.



Şekil 2.2: Bir yapay nöronun girdi vektörüyle beslenmesi[14]

Şekil 2.2 ve 2.1'deki değişkenlerin tanımları:

* X : girdi vektörü
* W : ağırlık matrisi
* z : girdi vektörünün ağırlıklı toplamı
* b : bayes terimi
* g : aktivasyon fonksiyonu
* $\hat{y}$: nöron çıktısı

Şekil 2.2'de gösterilen yapay nöronun, yaptığı hataya göre ağırlıklarını geri beslemeyle nasıl güncellendiği aşağıdaki formüllerle gösterilmiştir [15].

$$L(y,\hat{y}) = \frac{1}{2}(y-\hat{y})^2 \tag{2.1}$$

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial g(z)} \frac{\partial g(z)}{\partial z} \frac{\partial z}{\partial w_i} \tag{2.2}$$

$$w_i = w_i + \alpha \frac{\partial L}{\partial w_i} \tag{2.3}$$

2.2 Oto-kodlayıcılar

Oto kodlayıcılar, gözetimsiz bir şekilde eğitilrn ve girdi olarak aldıkları vektörleri daha küçük boyutlu yeni bir vektöre kodlayıp, girdi vektörünü bu kodlanılan vektörden yeniden üretmeye çalışan yapay sinir aplarıdır. Bu yapay sinir ağı, evrişimsel, özyineli, tam bağlı gibi farklı yapıdaki sinir ağları biraraya getirilerek oluşturulabilirler. Bu yapay sinir ağları ile yapılan girdi olarak gelen herhangi bir vektörü yeniden oluşturulmasını sağlayacak, gizli vektörler öğrenmektir. Bu sebepten oto-kodlayıcılar iki birleşenden oluşur. Bunlardan ilki kodlayıcı yapay sinir ağıdır. Kodlayıcı; girdi olarak alınan girdi vektörünü, öğrenilen gizli vektöre kodlar. İkinci birleşen ise kod çözücüdür. Kod çözücü; girdi vektörünün kodlandığı gizli vektörden tekrardan girdi vektörünü oluşturmaya çalışır. Kod çözücünün girdi vektörünü yeniden oluşturma hatasına göre tüm sistem geri beslenir.

Girdi vektörünü, daha sınırlı boyuttaki bir vektöre kodlayarak, girdi vektörlerini yeniden oluşturma gibi bir görev daha zorlu hale getirilmektedir. Fakat bu yöntem aynı zamanda istenilen kalitede gizli vektörler öğrenilmesini sağlamaktadır. Oto-kodlayıcılar, gösterim öğrenimi, yeni ögeler oluşturma vs. gibi farklı birçok amaç için kullanılabilir.

Bu tez çalışmasında varyasyonel oto-kodlayıcılar kullanılmıştır. Bu oto-kodlayıcı çeşidi eğitilirken, kodlayıcının oluşturduğu gizli vektöre belirli bir dağılıma sahip gürültü eklenerek, kod çözücü tarafından oluşturulan vektörler çeşitlendirilmeye çalışılmıştır. Bu sistem geri beslenirken, kod çözücünün girdi vektörüne ne kadar yaklaştığını gösteren hataya ek olarak, oluşturulan gizli vektörün belirli bir dağılıma ne kadar uzak yakın olduğunu gösteren başka bir skor daha hesaplanır. Tüm sistem bu iki skorun toplamından hesaplanan hataya göre geri beslenir [15].

2.3 Gradyan arttırma

Arttırma (Boosting) birden fazla yapay öğrenme modelinin birarada kullanılmasını sağlayan yöntemşlerden biridir. Bu yöntemde, öğrenilecek olan değerler için birden fazla basit yapıdaki yapay öğrenme modelleri birlikte kullanılır. Arttırma yönteminde, basit yapıdaki yapay öğrenme modelleri öğrenme işlemine tekrarlı bir şekilde dahil olurlar. Herbir tekrarda, bir önceki tekrardaki basit yapay öğrenme modellerinin yaptığı hatalar düzeltilir. Gradyan arttırma da bir arttırma yöntemidir ve herbir tekrarlamadaki basit

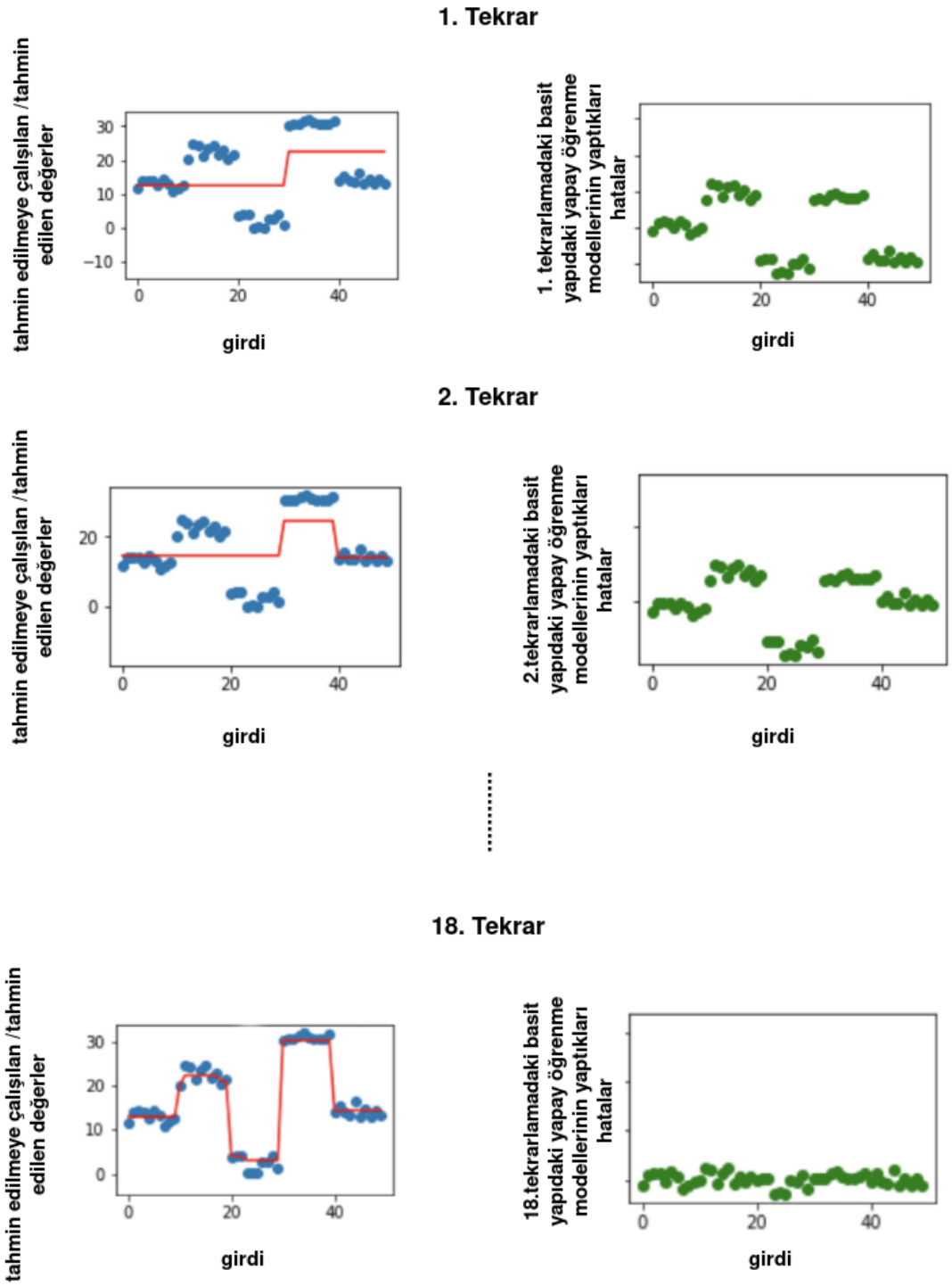
yapay öğrenme yöntemlerinin hatalarını öğrenmeye çalışarak toplam hatayı en iyilemeye çalışır. Gradyan arttırma modeliyle bir tahmin yapılırken, herbir tekrarlamada eğitilen tüm modellerin tahminleri birletirilir. Şekil 2.3'de Gradyan arttırmanın nasıl çalıştığı ayrıntılı bir şekilde görselleştirilmiştir. Çalışmalarımızda, bu modelle kullandığımız basit yapıdaki yapay öğrenme modelleri, karar ağaçlarıdır [16].

2.4 Gradyan iniş ve çıkış

Yapay öğrenme yöntemlerinde, gelen girdi vektöründeki öznitelikler belirli ağırlıklarla birleştirilerek bir tahmin yapılır. Bu tahminlerin, asıl değerlere ne kadar yaklaştığını anlamak için bir hata veya bir benzerlik fonksiyonu kullanırız. Yapay öğrenmedeki amaç, hesaplanan skoru (hata veya benzerlik fonksiyonu değerini) en iyileyecek değişken (ağırlık) değerlerini bulmaktır. Bu değerleri bulmayı sağlayan yöntemlerden ikisi gradyan iniş ve çıkıştır.
Bu yöntemler ile bir fonksiyonu en iyileyen değerin nasıl bulunduğu aşağıdaki formül ile gösterilmiştir. Aşağıdaki formül gradyan iniş işleminin nasıl yapıldığını göstermektedir, bu formüldeki gradyan, değişkene ekleniyor olsaydı, formül gradyan çıkış işleminin nasıl yapıldığını gösterirdi.

$$\Theta_{t+1} = \Theta_t - \alpha \nabla J(\Theta) \tag{2.4}$$

Güncellenecek değişkenin (Θ) t zamandaki değerine, en iyilenecek fonksiyonun (J) o parametreye göre gradyanı ($\nabla J(\Theta)$) belirli bir katsayı (öğrenme katsayısı(α)) ile çarpılarak çıkarırılır (veya eklenir.). Bu sayede değişkenin (t+1). zamandaki değeri hesaplanır. Gradyan bize en optimum değere ulaşmak için elimizdeki değeri nasıl güncelleyeceğimizi gösterir. Öğrenme katsayısı ise herbir iterasyonda değişkenimizi ne kadar güncelleyeceğimizi belirler.
Anlatılan güncelleme işlemi tüm değişkenler için belirli bir iterasyon sayısına ulaşıncaya ve optimum noktaya istenilen yakınlığa ulaşılıncaya kadar devam eder.



Şekil 2.3: Gradyan arttırma çalışma akışı: mavi noktalar mode tarafından tahmin edilmeye çalışılan değerleri, kırmızı grafik gradyan arttırma modelinin tahminini, yeşil noktalar ise herbir iterasyondaki basit yapıdak yapay öğrenme modellerinin hatasını gösterir. [17]

2.5 Elastik ağ

Bir yapay öğrenme modelinin, eğitim verisini ezberlemesini engellemek için modelin kompleksi azaltılır. Modelin kompleksini azaltan yöntemlerden biri düzenleştirici (regularization) kullanmaktır. Düzenleştiriciler, modelin ağırlıklarını kısıtlayarak, eğtim verisini ezberlememesini sağlarlar. Doğrusal yapay öğrenme modellerinde, düzenleştirme ridge, lasso ve elastik ağ yöntemleriyle yapılır.
Ridge düzenleştiricisinde, modelin en iyilemeye çalıştığı hata fonksiyonuna, $\alpha \sum_i w_i^2$ terimi eklenir. Örneğin modelimizin en iyilemeye çalıştığı hata fonksiyonu hataların karesi ortalaması (formül 2.5) ise; ridge düzenleştiricisi eklenmiş hata fonksiyonu formül 2.6'de gösterilmiştir. $\alpha \sum_i w_i^2$ terimindeki, α düzenleştirmeyi, modele ne kadar uygulayacağımızı belirleyen bir parametredir. Formül 2.5'deki n ise tahmin edilen öge sayısını gösterir [18].

$$L(y,\hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.5}$$

$$L(y,\hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \alpha\sum_{i} w_i^2 \tag{2.6}$$

Lasso düzenleştiricisinde, modelin en iyilemeye çalıştığı hata fonksiyonuna eklenen terim $\alpha \sum_i \mid w_i \mid$'dir. Lasso düzenleştiricisi eklenmiş hata fonksiyonu formül 2.7'de verilmiştir.

$$L(y,\hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \alpha\sum_{i} \mid w_i \mid \tag{2.7}$$

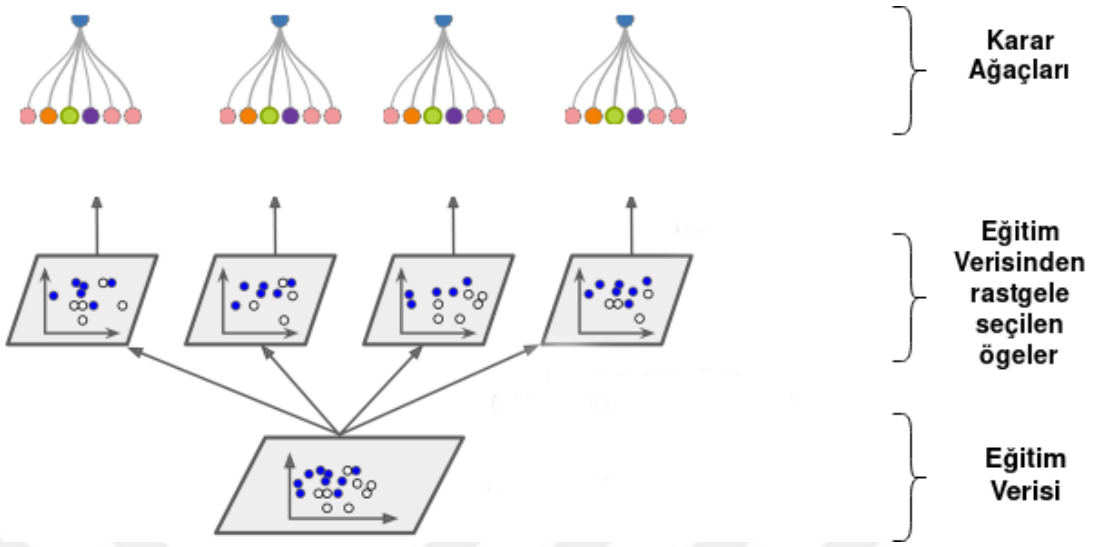
Elastik ağ ise lasso ve ridge düzenleştirme yöntemlerinin, belirli bir r oranına göre birleştirilmesidir (formül 2.8).

$$L(y,\hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + r\alpha\sum_{i} \mid w_i \mid + (1-r)\alpha\sum_{i} w_i^2 \tag{2.8}$$

2.6 Rastgele ağaç

Rastgele ağaç yöntemi, birden fazla karar ağacının torbalama (bagging) yöntemi ile biraraya getirildiği bir algoritmadır. Torbalama (bagging) yönteminde, birden fazla aynı çeşit yapay öğrenme modeli, eğitim verisinden rastgele seçilen ögelerle seçilir. Tahmin edilecek yeni bir öge geldiğinde, regresyon problemi için rastgele ögelerle eğitilen yapay öğrenme modellerinin tahminlerinin ortalaması alınır, sınıflandırmada ise, yeni öge için belirlenen (tahmin edilen) sınıf, yapay öğrenme modellerinin çoğunlukla tahmin ettiği sınıftır [19]. Şekil 2.4'de Rastgele Ağaç yönteminin nasıl çalıştığı ayrıntılı bir şekilde

görselleştirilmiştir.



Şekil 2.4: Rastgele ağaç modelinin eğitimi [20][21]

2.7 SHAP(SHapley Additive exPlanations) değerleri

Yapay öğrenme çalışmlarında, eldeki verilerle yapılabilecek en iyi tahminleri yapmak dışında, yapılan tahminleri hangi özniteliğin nasıl etkilediğinin analizi dikkat edilmesi gereken bir başka aşamadır. Literatürde, öznitelik önem analizi için kullanılan farklı, birçok yöntem vardır. Fakat [22]'a göre, bu yöntemlerden sadece SHAP değerleriyle tutarlı bir öznitelik analizi yapılabilir.

SHAP değerlerinin nasıl hesaplandığını anlamak için; ilk olarak bir oyun kuramı terimi olan Shapley değerlerinin nasıl hesaplandığı açıklanmalıdır. Shapley değerleri, bir özniteliğin aldığı değerin, yapay öğrenme modelinin tahminini nasıl etkilediğini gözlemlemek için kolayca kullanılabilir. Shapley değerlerini hesaplarken, herbir öznitelik bir oyundaki oyuncular, yapılan tahmin oynanan oyun ve Shapley değerleri, oyuncuların (özniteliklerin) aldıkları puan olarak düşünülebilir. Bir ögenin, bir özniteliğinin Shapley değerleri hesaplanırken; incelenen öznitelik dışındaki tüm özniteliklere alabilecekleri farklı değerler verilerek ve bu öznitelikleri farklı şekillerde biraraya getirerek yeni öznitelik vektörleri oluşturulur. Oluşturulan herbir farklı öznitelik vektörü için; bu vektöre hem incelenen öznitelik dahil edilerek hem de dahil öznitelik dahil edilmeden iki farklı tahmin yapılır. Herbir vektör için elde edilen iki tahmin arasındaki farklarının toplamının ortalaması, incelenen özniteliğin Shapley değeridir. (Formül 2.9)

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(M-|S|-1)!}{M!} [F(S \cup i) - F(S)] \tag{2.9}$$

Shapley değerleri, öznitelik analizinde, eklenebilirlik, verimlilik, etkisizlik, simetri

özelliklerini birarada sağlayan literatürdeki tek yöntemdir. [23]'da formül 2.9'deki S alt kümelerini oluştururken dahil edilmeyen özniteliklere sıfır değeri verilerek, tüm veri kümesi için yapılan öznitelik analizinin , oluşabilecek herhangi bir S alt kümesi için de tutarlı olması sağlanmıştır. Bunun için Shapley değerlerini toplanır bir fonksiyon(additive function) yardımıyla hesaplamışlardır. Bu şekilde hesapladıkları değerlere SHAP ismini vermişlerdir.

Yukarıda anlatıldığı gibi SHAP ve Shapley değerlerini hesaplamak için tüm öznitelik alt kümelerini kullanmamız gerekir. Dolayısıyla, bu değerleri hesaplamak üssel bir zaman karmaşıklığına sahiptir. [23]'da SHAP değerlerini polinomsal zamanda hesaplayan yöntemler geliştirilmiştir ve çalışmalarımızda bu yöntemlerden faydanılmıştır.

3. İLAÇ KOMBİNASYONLARININ SİNERJİ SKORU TAHMİNİ

Bu bölümde ilaç kombinasyonlarının sinerjisi ile ilgili literatürdeki çalışmalardan söz edilip, tez çalışmamızın ilk aşamasında kullanılan veri kümelerini nasıl elde ettiğimiz ve yapay öğrenme yöntemlerini sinerji skoru tahmini problemi için nasıl kullandığımız anlatılmıştır.

3.1 İlgili çalışmalar

İlaç kombinasyonlarının biraraya getirildiği zaman, oluşacak etkileri yapay öğrenme yöntemleriyle tahmin eden bir çok çalışma vardır[24]. Bu çalışmalardaki amaç sadece ilaç kombinasyonlarının sinerji skorunu tahmin etmek değildir. Bu bölümde anlatılan çalışmalardan anlaşılabileceği gibi, yapay öğrenme, sinerjik ilaç kombinasyonlarını bulma, ilaç kombinasyonlarının yan etkilerini tahmin etme gibi farklı birçok problemi çözmek için kullanılabilir.

[25] çalışmasında, mantarlar için sinerjik olabilecek yeni ilaç kombinasyonları tespit edilmeye çalışılmıştır. Bu amaç için üç farklı anti-mantar ilaç kombinasyonu veri kümesi birleştirilmiştir. Bu veri kümelerinde sinerjik olan ve olmayan ilaç kombinasyonları ile sinerjik olup olmadığı bilinmeyen ilaç kombinasyonları bulunur. Veri kümelerindeki herbir kombinasyon için ilaçlar arasındaki yapısal benzerlikler, hedef aldıkları ortak protein sayısı ve iki ilacın sinerjik kombinasyon oluşturduğu ortak ilaç sayısı doğrusal bir şekilde biraraya getirilir. Herbir ilaç kombinasyonu için bu şekilde biraraya getirilen bu bilgiler, LaplacianRLS(Laplacian Regularized Least Square) hata fonksiyonuna parametre olarak verilip, sınıflandırma fonksiyonu oluşturulur. Oluşturulan sınıflandırma fonksiyonu, bir kombinasyonun sinerjik olup olmadığını, hesapladığı olasılıklara göre tahmin eder. Geliştirilen bu yöntemle daha önce bilinmeyen yedi anti-mantar ilaç kombinasyonu elde edilmiştir.

[26] çalışmasındaki amaç sinerjik anti kanser ilaç kombinasyonlarını tahmin etmektir. Bu çalışma için rastgele ağaç yapay öğrenme modelinden yararlanılmıştır. Bu modeli eğitmek için kullanılan ilaç kombinasyonları, ilaç kombinasyonlarının aktivasyonlarını tahmin etmek için düzenlenen bir DREAM yarışmasından[27] alınmıştır. Belirli öznitelik lerle eğitilen bu rastgele ağaç modeli daha sonra, Connectivy Map[28] veri tabanında gen anlatımı öznitelikleri bulunan ilaçlarla oluşturulan kombinasyonların sinerjik olup

olmadığını test etmek için kullanılmıştır. Bu çalışma için çıkarılan öznitelikler, herbir kombinasyon için; iki ilacın(kombinasyondaki iki ilacın) hedef aldığı proteinlere göre oluşturulan Jaccard benzerliğini, iki ilacın(gene kombinasyondaki iki ilacın) hedef aldığı proteinlerin, protein-protein etkileşim ağındaki yakınlıklarını, kimyasal yapı benzerliklerini ve belirli kanser hücre hattında oluşturdukları farklı anlatımlı genleri göstereriler. Bu öznitelikler farklı kombinasyonlarla biraraya getirilerek, rastgele ağaç modelini eğitmek ve test etmek için kullanılmışlardır. Bu çalışmada yapılan deneylere göre, sinerjik kombinasyonları tahmin etmek için gen anlatım özniteliklerinin daha önemli olduğu görülmüştür. Aynı zamanda, Connectivy Map'ten[28] gen anlatımı öznitelikleri alınan 17 anti-kanser ialcıyla 187 ilaç kombinasyonu oluşturulmuştur. Bahsedilen özniteliklerle eğitilen rastgele ağaç modeliyle, bu kombinasyonlardan 28 tanesi sinerjik kombinasyon olarak belirlenmiştir.Belirlenen 28 sinerjik kombinasyondan üç tanesi literatürde etkili olarak bilinen anti-kanser ilaç kombinasyonu çıkmıştır.

İlaç kombinasyonlarında, ilaçların birarada kullanılmasının beklenmedik yan etkileri olabiliyor. [29] çalışmasında, çizge evrişimsel yapay sinir ağı kullanılarak, bu yan etkiler tahmin edilmeye çalışılmıştır. Bu amaç için protein-protein etkileşim, ilaç-protein etkileşim ve ilaç-ilaç etkileşim ağları birleştirilerek çoklu çizge(multigraph) oluşturulmuştur. Bu çoklu çizgede, iki ilaç köşesi arasında bulunan kenarlar yan etki çeşidini göstermektedir. Dolayısıyla ele aldıkları problemi, bir çizgede kenar çeşidi tahmini problemine çevirmişlerdir. Geliştirdikleri çizge evrişimsel yapay sinir ağının adı Decagon'dur. Decagon bir kodlayıcı ve bir kod çözücüden oluşur. Kodlayıcı, çoklu çizgedeki herbir köşe için bir gömülüm üretir. Kod çözücü, herbir iki köşe gömülümü kombinasyonu için, aralarında olabilecek tüm kenar çeşitlerinin olasılığını çıkarır. Çapraz entropi kaybına göre tüm sistem geri beslenir. Bu sistemle, literatürde bulunan tüm kenar tahmini yöntemlerinden daha başarılı sonuçlar elde edilmiştir.

[30] çalışmasında, sinerji skoru tahmini yapmak için oluşturulan medikal veri, rastgele ağaç, ANFIS(Adaptive-Network-Based Fuzzy Inference System), DENFIS(Dynamic Evolving Neural-Fuzzy Inference System), GFS.GCCL(Fuzzy Rules Using Genetic Cooperative-Competitive Learning) yapay öğrenme yöntemleriyle kullanılmıştır. Daha sonra, bu modellerden elde edilen tahminler belirli ağırlıklarla birleştirilmiştir. Bu yöntem, üzerinde çalıştıkları veri kümesi için karşılaştırdıkları diğer yöntemlerden daha başarılı olmuştur.

[31]'de geliştirilen çalışmada gene bir DREAM yarışması[27] verisi üzerinde test edilmiştir ve bu veri kümesi ile diğer yöntemlerle alınan en iyi PC-indeks %61 iken, bu yöntemle bu %78' yükselmiştir. Bu yöntem kullanılarak bir ilaç kombinasyonunun sinerjik olup olmadığını anlamak için, ilaç kombinasyonun iki aşamadan geçmesi gerekmektedir. İlk olarak, bilinmeyen ilaç kombinasyonun, sinerjik olup olmadığı

bilinen ilaç kombinasyonlarına benzerliği hesaplanır. Bu benzerlik kimyasal yapıları, ilacın kimyasal bazı özelliklerini, ilaç-protein etkileşim ağını gösteren yedi tane özniteliğe göre yapılır. İlaçların benzerlikleri hesaplanıp, bu benzerliklere göre sıralanırlarken bir yarı-gözetimli öğrenme yöntemi kullanılmıştır. Belirli bir benzerliğe sahip olan kombinasyonlardaki ilaçların, belirli hücre hatları üzerine uygulanması sonucu elde edilen farklı anlatımlı genleri, Permutation istatistiksel testine sokulur. Bu test sonucu hesaplanan p değeri 0.05'ten küçükse ilaç kombinasyonu sinerjik olarak belirlenir.

[32] çalışmasında sıtma hastalığı için yapay öğrenme modelleriyle hastalığı için sinerjik ilaç kombinasyonları tespit edilmeye çalışılmıştır. Bu çalışmada geliştirilen yöntem, 'bir hastalık sonucu ortaya çıkan farklı anlatımlı genleri, ters yönde etkileyen ilaçlar hastalığın çözümü için etkilidir' hipotezine göre şekillendirilmiştir. Bu sebepten test verisi oluşturmak için, ilk olarak sıtma hastası olan çocukların kan örnekleri alınmıştır. Bu örneklere göre, sıtmanın gen anlatımı imzaları çıkarılmıştır. Daha sonra çıkarılan gen imzalarını negatif yönde etkileyen ilaçlar LINCS[33] veri tabanından yararlanarak bulunmuştur. Eğitim verisi ise NCATS[34] kullanılarak 56 ilaçla oluşturulan 1540 kombinasyondan oluşur. [35] ve [36]'da anlatılan sistemler kullanılarak, eğitim ve test verisindeki her bir ilacın hedef aldığı proteinler belirlenmiştir. Belirlenen hedeflerin, Biosystems veri tabanı[37] kullanılarak insan vücudundaki 2010 metabolizmik gidişatta ne durumda olduğu belirlenmiştir. Eğitim ve test verisindeki herbir kombinasyondaki ilaçlar için çıkarılan 2010 uzunluğundaki bu vektörler çarpılarak herbir kombinasyon için birleştirilir. Herbir kombinasyon için oluşan bu vektörler, rastgele ağaç modeline girdi olarak verilmiştir. Eğitilen rasgele ağaç, LINCS[33] veri tabanından çıkarılan ilaçlarla oluşturulan kombinasyonların sinerjik olup olmadığını tahmin etmek için kullanıldı. Bu tahmin sonucu, sinerjik olduğu bilinen kombinasyonlar tespit edilmiştir. Bu da bu yöntemin yeni sinerjik kombinasyonlar bulmak için kullanışlı bir yöntem olduğunu gösteriyor.

DeepSynergy[7], ikili ilaç kombinasyonlarının, otuz dokuz tane kanserli hücre hattına uygulanması sonucu elde edilen sinerji skorları tahmin edilmeye çalışılmıştır. Bu sinerji skorları çıkarılırken Loewe matematiksel modeli[4] kullanılmıştır. Bu çalışmada sinerji skorlarını tahmin etmek için, sinerji skorunu elde etmek için kullanılan iki ilaç ve hücre hattının öznitelikleri birleştirilip tam bağlı yapay sinir ağına verilmiştir. Uygulanan bu yöntem ile sinerji skoru tahmini için literatürdeki en başarılı sonuçlar elde edilmiştir. DeepSynergy[7] ile alınan sonuçlar, aynı veri kümesi için, TreeCombo[8] çalışmasıyla daha iyi hale getirilmiştir. Bu çalışmada gradyan arttırma algoritmasından yararlanılmştır. TreeCombo[8] çalışmasında aynı zamanda, gradyan arttırma ve SHAP değerleri[9] kullanılarak özniteliklerin önemi hesaplanıp, bu özniteliklerin değerlerine göre gradyan arttırma modelinin performansının nasıl değiştiği gözlemlenmiştir.

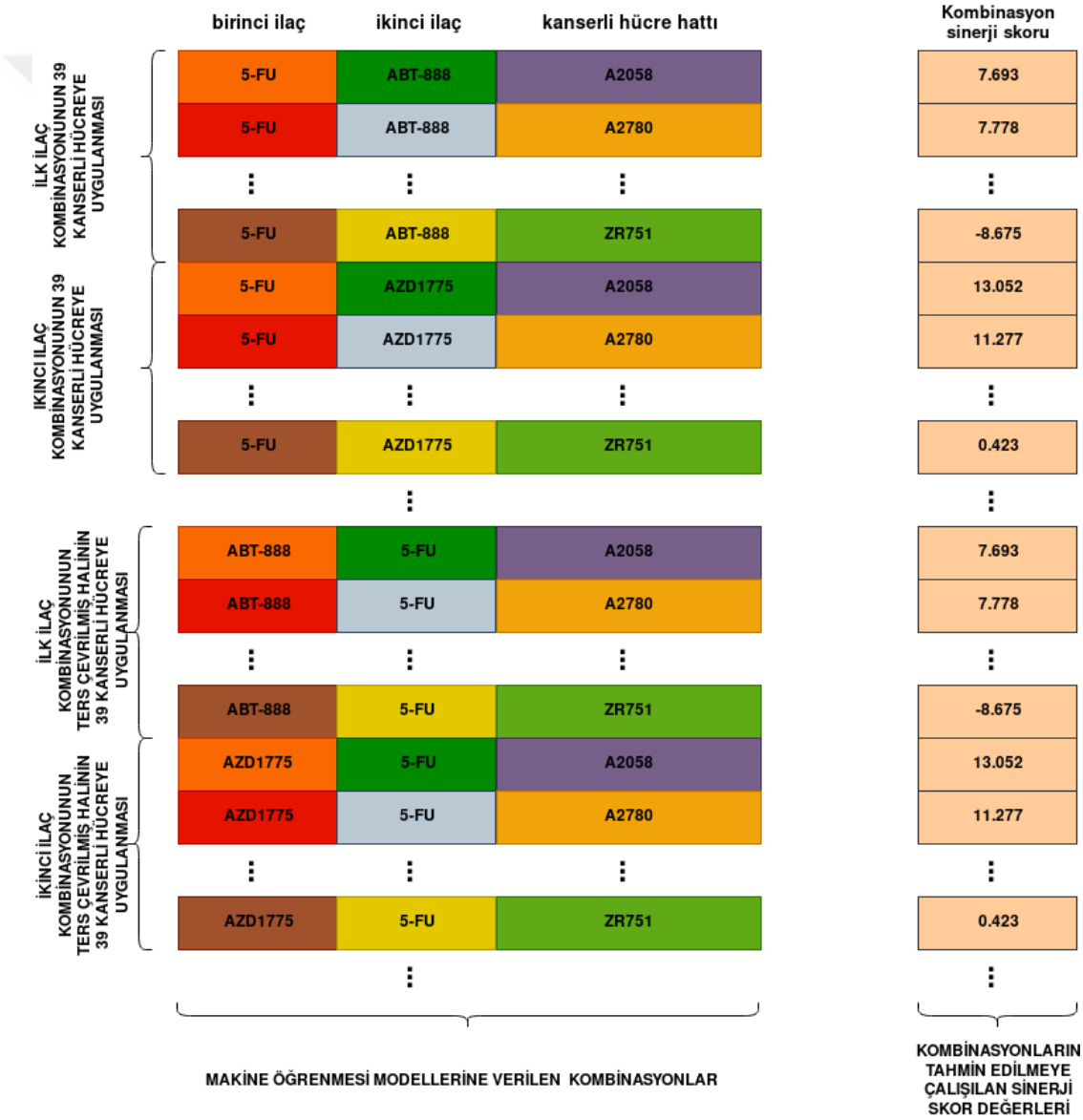
[38] çalışmasında, DeepSynergy[7] ve TreeCombo[8] çalışmalarındaki aynı veri kümesi kullanılmıştır. Bu çalışmada kullanılan öznitelikler, [7] ve [8] çalışmalarından farklı olarak gen anlatımını, ilaçların hedef proteinlerini, ilaçların kimyasal özelliğini ve sentetik öldürücülüğünü gösterir. Bu özniteliklerle beş gruplu çapraz doğrulamadan en yüksek sonucu aşırı rastgele ağaç ile almışlardır. Daha sonra aynı öznitelikler ve rastgele ağaç kullanarak sinerjik ve antagonistik ilaç kombinasyonlarda özniteliklerin nasıl değiştiğini gözlemlemişlerdir. Bu amaç için problemi regresyondan sınıflandırmaya çevirmişlerdir. Maalesef sinerjik ilaç kombinasyonlarını belirlediklerini düşündükleri öznitelikler için literatürde herhangi bir kanıt bulamamışlardır.

3.2 Veri kümeleri

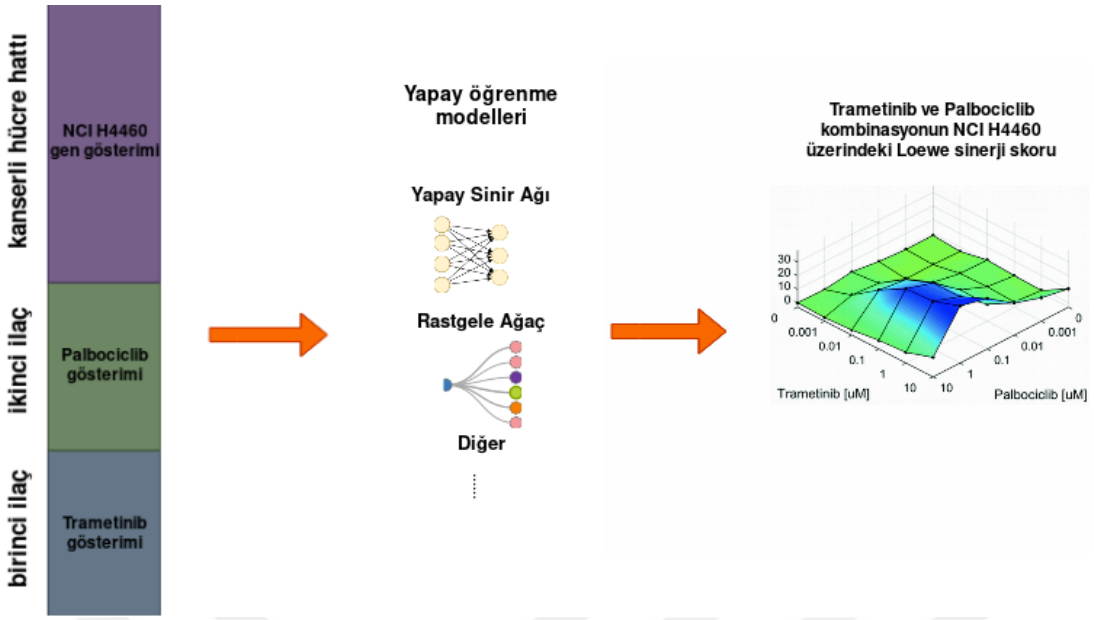
Giriş kısmında da bahsedildiği gibi, tez çalışması; ilaç gösterimlerinin belirli yapay öğrenme modellerine etkisini sinerji skor problemi için gözlemlemek ve yapay öğrenme yöntemleriyle, bir ilaç ve bir kanserli hücre hattı için istenilen sinerji skorunu verecek olan ilacı üretmek olarak iki bölümden oluşuyor. Bu çalışmaları yapmak için dört farklı ilaç gösterimi kullanılarak, dört farklı veri kümesi oluşturuldu. Bütün veri kümelerindeki, herbir ögede; iki ilaç ve bir hücre hattının öznitelikleri bulunmaktadır. Herbir öge için öznitelikler, birinci ilacın öznitelikleri-ikinci ilacın öznitelikleri-hücre hattının öznitelikleri şeklinde sıralanmıştır. Buna ek olarak, tüm veri kümelerinde yapay öğrenme yöntemlerinin ilaçların sırasından etkilenmemesi için, herbir öge için öznitelikler, ikinci ilacın öznitelikleri-birinci ilacın öznitelikleri-hücre hattının öznitelikleri şeklinde yeniden sıralanıp veri kümelerine eklendi. Gözetimli yapay öğrenme modelleri ile tahmin edilmeye çalışılan değerler; bir veride bulunan iki ilacın birarada kullanılıp, gene aynı veride bulunan hücre hattına uygulanmasıyla elde edilen sinerji skorlarıdır.
İlaç kombinasyonları, hücre hatları ve tahmin etmeye çalıştığımız sinerji skor değerleri, [39]'den alındı. Bu veri kümesinde otuz sekiz anti kanser ilacı ile oluşturulan, beş yüz seksen üç tane farklı ilaç kombinasyonunun, otuz dokuz tane farklı kanserli hücre hattına uygulanmasıyla hesaplanan sinerji skorları bulunur. Bu veri kümesindeki sinerji skorları Loewe yöntemi kullanılarak hesaplanmıştır. Sinerji skoru tahmini için şimdiye kadarki en iyi metodlar olan DeepSynergy[7] ve TreeCombo'da da[8] aynı onkoloji veri kümesinden yararlanıldı. Giriş kısmında da bahsedildiği üzere, ilk aşamadaki amaçlarımızdan biri; bu aşamada elde edilen sonuçları şimdiye kadarki en iyi yöntemlerle karşılaştırmak olduğu için, bu çalışmalarda kullanılan kombinasyonlardan yararlanıldı.
Bu çalışmanın ilk kısmı için kullanılan ilaç gösterimleri : belirli ilaçlar uygulandıktan sonra etkilenen genleri gösteren karakteristik yönelim verisi[40], ilaçların topolojik ve fiziksel özelliklerini gösteren Chemopy[41], jCompoundMapper[42] kütüphenelerinden elde edilen vektörler, [43]'deki gözetimli çizge yapay sinir ağı kullanılarak elde edilen

molekül çizge gömülümleridir. Bu üç farklı ilaç gösterimi kullanılarak, üç farklı veri kümesi oluşturulmuştur. Tez çalışmasının ilk kısmındaki amaç; ilaç gösterimlerinin, sinerji skoru tahmini üzerindeki etkisini incelemek olduğu için, oluşturulan üç farklı veri kümesinde değişen tek özniteliklerin ilaç gösterimleri olması gerekmektedir. Bu sebepten tüm veri kümelerindeki hücre hattı öznitelikleri ve ögelerin gösterdiği kombinasyonlar aynıdır. Veri kümelerini bu şekilde oluşturarak ilaç gösterimleri arasında, uygun ve doğru bir karşılaştırma yapabildik.

Şekil 3.1'de oluşturduğumuz veri kümelerinin genel yapısı gösterilmiştir. Buna ek olarak Şekil 3.2'de ise birinci aşamada oluşturduğumuz veri kümelerindeki herbir ögenin sinerji skorlarını nasıl tahmin ettiğimiz görselleştirilmiştir.



Şekil 3.1: Veri Kümelerinin Yapısı



Şekil 3.2: Birinci Aşamada İzlenilen Genel Yöntem Örneği[20][7][44]

3.2.1 Öznitelikler

Bu bölümde, hücre hattı ve ilaç özniteliklerini çıkarmak için kullanılan yöntemler ve prosedürler anlatılmıştır.

1. **İlaç Öznitelikleri**:

   Çalışmalarımızda kullanılan ilaç öznitelikleri aşağıdaki başlıklar altında şu şekilde açıklanmıştır; Karakteristik Yönelim(CD), ilaçların kimyasal özelliklerini gösteren tanımlayıcılar(Chem), [43]'deki kullanılan yöntem referans alınarak oluşturulan ilaç gösterimleri(GNN).

   (a) **CD**:

   Gen anlatımı; bir protein enzim vs. gibi ürünler oluşturmak için gendeki bilginin sentezlenmesi olayıdır. Farklı anlatımlı gen (DEG); bir gendeki bilginin sentezlenme miktarının (gen anlatımı miktarı), iki farklı deneysel ortam (durum, koşul vs.) arasında, istatistiksel olarak farklı olmasıdır. Bu tür genler, özellikle biyolojik ve fizyopatolojik alanlardaki çalışmalar için önemli veri kaynaklarıdır. Normal ve hastalıklı insanlardaki farklı anlatımlı genlerin belirlenip, hastalığın nedenlerinin anlaşılması ve buna göre bir tedavi geliştirilmesi, önemli veri kaynakları oldukları durumlara örnek verilebilir. Bu genleri belirlemek için literatürde birden fazla farklı yöntem bulunmaktadır. Karakteristik Yönelim (CD)[40] bu yöntemlerden biridir.

Karakteristik Yönelim (CD)'de, doğrusal sınıflandırma yöntemi kullanılarak farklı anlatımlı genler bulunmaya çalışılır. Bu sınıflandırma için bir hiper düzlem belirlenir. Bu hiper düzlemin normalinin yönü farklı anlatımlı genleri belirlemek için kullanılır.

$$\log \frac{Pr(G=k|X=x)}{Pr(G=l|X=x)} = \log \frac{\pi_k}{\pi_l} - \frac{1}{2}(\mu_k-\mu_l)^T\Sigma^{-1}(\mu_k-\mu_l)+x^T\Sigma^{-1}(\mu_k-\mu_l) \tag{3.1}$$

Bir gen anlatımının k ve l sınıfına ait olma olasılığı, aslında bu iki sınıf arasında, düzlemin yaptığı oryantasyona göre belirlenir. Yapılan oryantasyon, yukarıdaki formülde $b = \Sigma^{-1}(\mu_k - \mu_l)$ terimidir. Şuan b sadece düzlemin oryantasyonunu göstermektedir. Bu değişkenden, her bir genin yönelimi; yön kosinüslerine göre b değişkenini birleşenlerine ayırıp, bu birleşenlerin büyüklükleri alınarak çıkarılır.

Anlatılan bu yöntem, farklı anlatımlı gen belirlemede t-test, SAM ve Limma gibi yöntemlerle AUC metriğine göre karşılaştırılmıştır. Bu yöntem, diğer yöntemlere göre daha iyi bir performans sergilemiştir. Bu sonuç Karakteristik Yönelim (CD)'in farklı anlatımlı genleri belirlemek için daha hassas bir yöntem olduğunu gösteriyor.

Deneylerimizde kullanılan CDR veri kümesini oluştururken kullanılan ilaç gösterimleri; LINCS L1000[33] gen anlatımı verisine, Karakteristik Yönelim (CD)[40] uygulanarak belirlenen gen anlatım imzalarıdır.

Deneylerimizde kullanılan bu ilaç gösterimleri [45]'den alınmıştır.

(b) **Chem**:

DeepSynergy[7] ve TreeCombo[8] çalışmalarında kullanılan ilaç gösterimidir. Bu gösterim, 1309 uzunluğundaki ECFP_6 vektörlerinden, molekülün fiziksel ve kimyasal özelliklerini gösteren 802 uzunluğundaki öznitelik vektörlerinden ve belirli zehirli moleküler alt-yapılara(Toxicophore) sahip olunup olunmadığı nı gösteren 2276 uzunluğundaki iki tabanındaki vektörlerden oluşur.

i. **ECFP_6:** Bu öznitelikler jCompoundMapper kütüphanesi kullanılarak oluşturulmuştur. Extended-connectivity fingerprint (ECFP)[46] üretmek için, ilk iterasyonda her bir atoma birbirinden farklı olmak üzere tam sayı değerleri verilir. Daha sonra her bir atom için, bitişiğinde bulunan komşularının tamsayı değerleri bir araya getirilerek bir dizi oluşturulur. Oluşturulan bu diziler, bir özetleme fonksiyonundan geçirilerek tekrar bir tam sayıya çevrilirler. Oluşturulan yeni tam sayılar, atomların yeni

değerleridir. Her iterasyondan sonra güncellenen değerler başka bir dizide kaydedilir. Belirli iterasyondan sonra bu işlemler sonlandırılır. Her bir işlemden sonra oluşan tamsayı değerlerini kaydettiğimiz dizi, molekülün ECFP parmakizidir. ECFP isminin sonuna eklenen rakam dönülecek iterasyon sayısının iki katıdır. Çünkü her bir iterasyonda, o iterasyon sayısının iki katı uzaklığındaki alt-çizgeler güncelleme işlemine dahil ediliyor. Dolayısıyla ECFP, bir molekülün alt çizgelerinin topolojisinin bir tam sayı vektörüne çevrilmiş halidir.
Rdkit ve jCompoundMapper gibi kütüphaneler bu tamsayı dizisini (vektörünü) tekrar belirli uzunluktaki bitlere özetlerler.

ii. **Fiziksel ve Kimyasal Özellikler:** Bu özellikler Chemopy kütüphanesi kullanılarak çıkarılmıştır. Bu kütüphane kullanılarak çıkarılan 802 özniteliği, gösterdikleri özelliklere göre 9 başlık altında ifade edebiliriz.
   A. **CPSA Tanımlayıcıları:** Molekülün polar bağ yağma isteğiyle alakalı özniteliklerdir.
   B. **WHIM, MOE, Geometrik Tanımlayıcıları:** Molekülün şekli ve büyüklüğüyle alakalı özniteliklerdir.
   C. **Gary ve Monan Korelasyonları:** Verilen ağırlık, Van der Waals, polarizasyon gibi özelliklere göre, bir moleküldeki atomların ne kadar korelasyon halde bulunduklarını gösteren özelliklerdir.
   D. **Yük Tanımlayıcıları:** Molekülün yaptığı hidrojen bağlarını, atomla rın yük durumlarını gösteren niteliklerdir.
   E. **Morse Tanımlayıcıları:** Elektronların dalga yapısını gösteren tanım layıcılardır.
   F. **Moleküler Bağlantı Endeksleri:** Alt-çizgelerin ve atomların nasıl bağlı olduğu ve ulaşılabilirlik bilgilerini gösterirler.
   G. **Moleküler Yapısal Tanımlayıcılar:** Molekül ile ilgili herhangi bir geometrik ve komşuluk bilgisi vermeden, oksijen atom sayısı, hidrojen atom sayısı, molekül ağırlığı gibi bilgilerle yapısal açıdan özetleyen bilgilerdir.
   H. **RDF Tanımlayıcıları:** Bir atomun , belirli bir yarıçaplı kürede rastlanılma olasığını gösterir. Bu tanımlayıcılar, molekülün tüm atomları için olasılık hesaplandıktan sonra bir değer alırlar.
   I. **Moleküler Özellikler:** Bu özellikler bir molekülün; bir molünün elektron verme isteğini, çözünürlüğünü (kalıcılığını), yüzeyindeki polar atomlar toplamını, çember (ring) ve π bağları toplamını

ve suyla etkileşmekten kaçınma direnci olmak üzere toplam 5 özelliğini gösterirler.

iii. **Toxicophore Öznitelikleri:** Bir molekülün zehirli olmasına sebep olan alt yapılara toxicophore denir. Deneylerimizde bir molekül için olup olmadığı kontrol edilen toxicophore alt yapıları, OCHEM[47] veri kümesi sayesinde çıkarılmıştır.

(c) **GNN**:

[43]'da ilaç-protein etkileşimini tahmin etmek amacıyla (ikili sınıflandırma problemi), derin yapay sinir ağları kullanarak, ilaç ve protein dizilimleri için gösterim öğrenimi gerçekleştirmişlerdir. Gösterim öğrenimi için uçtan uca öğrenme tekniği kullanılmıştır. Bu öğrenme yönteminde, ayrık girdi vektörleri, belirli uzunluktaki sürekli vektörlere gömülür ve diğer yapay sinir ağı katmanları bu gömülümü girdi alarak bir çıktı üretir. Asıl yapay sinir ağının yaptığı hataya göre, tüm sistem baştan geri beslenirken gömülüm vektörü de güncellenir. Bu şekilde, yapay sinir ağının tahmin etmeye çalıştığı değerler için en optimal gösterimler öğrenilir.

[43]'da ilaç-protein etkileşimini tahmin edecekleri ilaç çizgesini ve protein dizilimini girdi olarak almıştır. İlaç çizgelerinin, gösterimi çizge yapay sinir ağı kullanarak elde edilirken; protein dizilimlerinin gösterimleri, evrişimsel yapay sinir ağları kullanarak elde edilir. Çizge yapay sinir ağı ve evrişimsel yapay sinir ağı çıktıları birleştirilerek, ilaç protein etkileşimi tahmini yapmak üzere tam bağlı yapay sinir ağına verilir. Bu yapay sinir ağının hatasına göre tüm sistem (çizge yapay sinir ağının ve evrişimsel sinir ağının başından itibaren) geri beslenir.

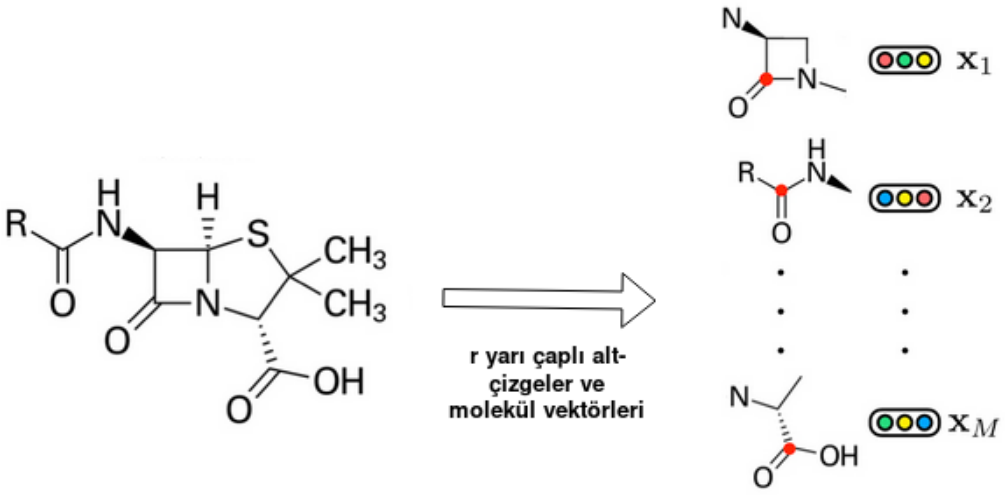
[43]'daki çalışmaya göre bu modelleri kullanarak, belirli verilerde daha iyi bir performans sergilenebiliyor. Aynı zamanda çalışılan veri düzensiz olsa bile, literatürdeki diğer yapay öğrenme yöntemlerine göre daha kararlı bir performans göstermiştir.

Biz bu çalışmada kullanılan çizge yapay sinir ağları ve uçtan uca öğrenme tekniğini; tez çalışmasının ilk aşaması olan sinerji skoru tahmini problemine uyarladık. Bu sayede, çizge yapay sinir ağları ile oluşturulan vektörler, sinerji skoru tahmini için incelenen ilaç gösterimlerinden biri olmuştur.

Uçtan uca öğrenme ve çizge yapay sinir ağı kullanarak ilaç gösterimi oluşturma aşamaları şunlardır:

i. Çizge yapay sinir ağı girdi olarak bir molekül vektörü ve molekül çizgesinin komşuluk matrisini alır. Molekül vektörü oluşturulurken;

molekül r yarı çaplı alt-çizgelere ayrılır. Her bir alt-çizgedeki farklı iki atom arasında bulunan kenar, bir sözlük veri yapısında tutulur. Molekül vektörü de, her bir r yarı çaplı alt-çizgedeki kenarların, sözlükte bulunma sırasını gösterir. Molekülün r yarı çaplı alt çizgelere bölünmesinin sebebi; moleküllerdeki farklı çeşit atom sayısının, gösterim öğrenimi için çok az olmasıdır. Dolayısıyla bu işlem girdi vektörlerini, daha yoğun bir hale getirmek için yapılmıştır.



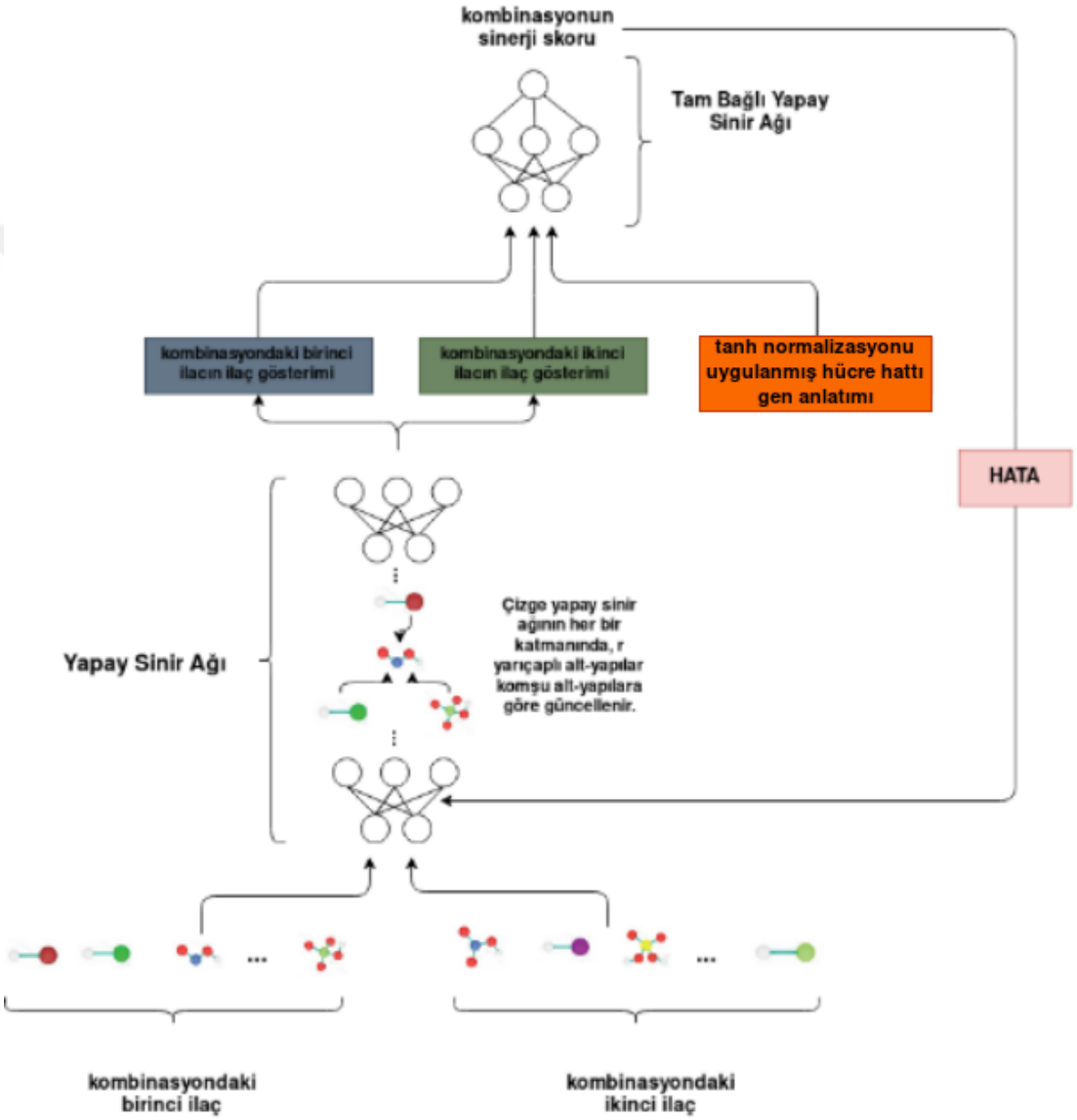
Şekil 3.3: Molekül vektörlerinin oluşturulması[43]

ii. Oluşturulan molekül vektörleri, çizge yapay sinir ağında, ilk olarak gömme katmanından geçilir. Bu şekilde sürekli hale getirilen vektör 3.2. formülde gösterildiği gibi, çizge yapay sinir ağının diğer katmanlarında; vektör, ağırlık matrisleriyle çarpılıp, ReLU aktivasyon fonksiyonundan geçirilip güncellenir. Güncellenen vektör, molekülün komşuluk matrisi ile çarpılır ve güncellenmeden önceki haliyle toplanır. Bu sayede, girdi molekül vektörü, belirli uzaklıktaki komşularının topolojisine göre güncellenmiş olur.

$$x_i^{(l+1)} = x_i^{(l)} + \sum_j f(x_j^{(l)}) \tag{3.2}$$

iii. Çizge yapay sinir ağının her bir katmanından geçirildikten sonra, girdi vektörü farklı bir vektör haline getirilir. Önceki kısımlarda anlatıldığı gibi, sinerji skoru tahmin ederken ilaç kombinasyonları ile çalıştığımız için, yapay sinir ağına, bir kombinasyon için iki ilacın molekül çizgeleri verilir. Kombinasyonlardaki ilaçlar, birbirinden bağımsız ve her biri yirmi beş uzunluğunda iki farklı vektöre çevrilir. Çevrilen bu vektörler,

tanh normalizasyonundan geçirilmiş hücre hattı öznitelikleriyle birleştiri lir. Bu birleştirmeyle oluşan girdi vektörü, girdi olarak verilen birinci ilaç-ikinci ilaç-hücre hattı kombinasyonunun sinerji skor tahminini yapan bir tam bağlı yapay sinir ağına bağlanır. Bu tam bağlı yapay sinir ağının yaptığı hataya göre, tüm sistem çizge yapay sinir ağından başlanarak geri beslenir. Bu sayede, tam bağlı yapay sinir ağının hatasını en aza düşüren, ilaç gösterimleri öğrenilmiş olur.



Şekil 3.4: Çizge yapay sinir ağının uçtan uca öğrenme ile ilaç gösterimi oluşturması

Yukarıda anlatılan sistem belirli bir iterasyon sayısına kadar eğitilir. Eğitim tamamlandıktan sonra, bir ilaç için, yukarıdaki sistemdeki çizge yapay sinir ağı sonucunda oluşan vektör, o ilacın GNNR verisetinde kullanılan

gösterimidir.

2. **Gen Öznitelikleri**: Mikrodizin ve FARMS, gen anlatımı özniteliklerini çıkarmak için yaygın olarak kullanılan yöntemlerdir. Deneylerimizde kullanılan ve kanserli hücre hattının gen anlatımları mikrodizi yöntemiyle üretilmiştir (E-MTAB-3610 veri kümesi). Bu gen anlatımlarındaki sinyal ölçümleri Factor Analysis for Robust Microarray Summarization (FARMS) yöntemiyle bir araya getirilmiştir. FARMS, bu ölçümleri birleştirirken faktör analizi modeli kullanmaktadır. Bu faktör analizi modelinin parametreleri, Bayesian maksimum soncul yöntemiyle optimize edilir. Bu da Gauss dağılımının dışındaki (ilginç) sinyallerin daha kolay tespit edilmesini sağlar (özetlenen sinyallerin sonuçlarının daha başarılı olması için zayıf sinyaller bu aşamaya dahil edilmemiştir.).

3.2.2 Veri kümelerilerine uygulanan ön işlemler

İlk olarak, [39]'daki veri kümesinde bulunan otuz sekiz ilaçtan sadece yirmi dokuz tanesinin karakteristik yönelim verisi bulunmaktadır. Çalışmanın ilk kısmında, farklı ilaç gösterimlerini doğru bir şekilde karşılaştırmak için, geriye kalan dokuz ilacın bulunduğu ögeler veri kümelerinden çıkarılıp çalışmamıza dahil edilmedi. Ortak olan ögelere göre veri kümeleri düzenlendikten sonra, ilk aşama için ChemR ve CDR veri kümelerine tanh normalizasyonu uygulandı. Uygulanan tanh normalizasyonu DeepSynergy[7] ve TreeCombo[8]'da uygulanan normalizasyon işleminin aynısıdır. ChemR ve CDR'ın aksine, GNNR veri kümesindeki ilaç gösterimlerine herhangi bir normalizasyon işlemi uygulanmadı. 1c kısmında anlatıldığı gibi çizge gömülümlerini öğrenen gözetimli tam bağlı yapay sinir ağı[43]; molekül çizge gömülümleri herhangi bir ön işlemden geçirilmeden, tanh normalizasyonu uygulanmış hücre hattı öznitelikleriyle bağlananan vektörü girdi olarak alarak bir sinerji skoru tahmini yapıyor. Model performanslarını, bu ilaç gösterimi için doğru bir şekilde karşılaştırmak amacıyla, diğer modellere verilen GNNR veri kümesi, tam bağlı yapay sinir ağı tarafından öğrenilen, herhangi bir ön işlemden geçirilmeyen ilaç gösterimleriyle tanh normalizasyonundan geçirilen hücre hattı özniteliklerinden oluşmalıdır.

3.3 Sinerji skoru tahmini

Tez çalışmasının birinci aşamasında, CD, Chem ve GNN ilaç gösterimleriyle oluşturulan veri kümeleri, 3.2.2 kısmında anlatılan aşamalardan geçirilmiştir. Bu veri kümelerine daha sonra 5.1'da anlatılan şekilde gruplara ayrıldı. Ayrılan bu gruplar üzerinde, elastik ağ, tam bağlı yapay sinir ağı(TBYSA), gradyan artırma(GA) ve rastgele ağaç(RA) yapay öğrenme modelleri çalıştırıldı. Bu modellere parametre optimizasyonunun nasıl

uygulandığı ve bu parametre optimizasyonunun sonuçları 5.2'da verilmiştir. Bütün bu işlemlerden sonra, herbir farklı veri kümesi üzerinde çalıştırılan yapay öğrenme modellerinden, sinerji skoru tahmini için, ortalama hata karesi ve Pearson korelasyon metrikleri elde edildi. Farklı veri kümeleri üzerinde çalıştırılan bu yapay öğrenme modelleri, 5.2.1 bölümünde anlatıldığı gibi ağırlıklı ortalama yöntemiyle birleştirildi. Ağırlıklı ortalama yöntemi, ortalama alma, istifleme gibi diğer biraraya getirme yöntemle rinden, bizim problemimiz için, daha iyi bir performans sergilemiştir.

4. SİNERJİ SKORU OPTİMİZASYONU

Bu bölümde bir yapay öğrenme modelinin tahminini en iyileyen ve molekül oluşturma için kullanılan yapay öğrenme modelleriyle ile ilgili literatürdeki bazı çalışmalardan söz edilip, tez çalışmasının ikinci kısmında üzerinde çalıştığımız sinerji skoru tahmini yapan bir yapay öğrenme modelinin tahminini en iyileyecek molekülü üretme problemini nasıl çözdüğümüz anlatılmıştır.

4.1 İlgili çalışmalar

Bir yapay öğrenme modelinin çıktısını en iyileyecek girdiyi veya molekülü bulmak için, özellikle son iki yılda özellikle derin öğrenme mimarileriyle birçok çalışma yapılmıştır [13]. Bu bölümde literatür araştırmamız sonucunda bulduğumuz ve bu aşamada izlediğimiz yönteme en benzer çalışmalar özetlenmiştir.

[48]'de oto-kodlayıcı ile öğrenilen bir dizi gösterimleri, gözetimli bir iş için yapay öğrenme modellerine girdi olarak verilir. Bu gösterimlerle eğitilen yapay öğrenme modelleri, gradyan çıkış işlemi uygulanarak, yapay öğrenme yöntemlerinin çıktılarını en yüksek değere çıkaran girdi bulunmaya çalışılmıştır. Gradyan çıkış işleminde kullandıkları, gradyan adımlarda kullandıkları ceza fonksiyonu, problemleri için uygun bir girdi bulmalarını sağlamaktadır. Bu sayede karşılaştırdıkları diğer yöntemlerden daha doğru sekanslar elde etmişler. Yaptıkları bu çalışma molekül SMILES dizileri dahil her türlü sekansa uyarlanabilir.

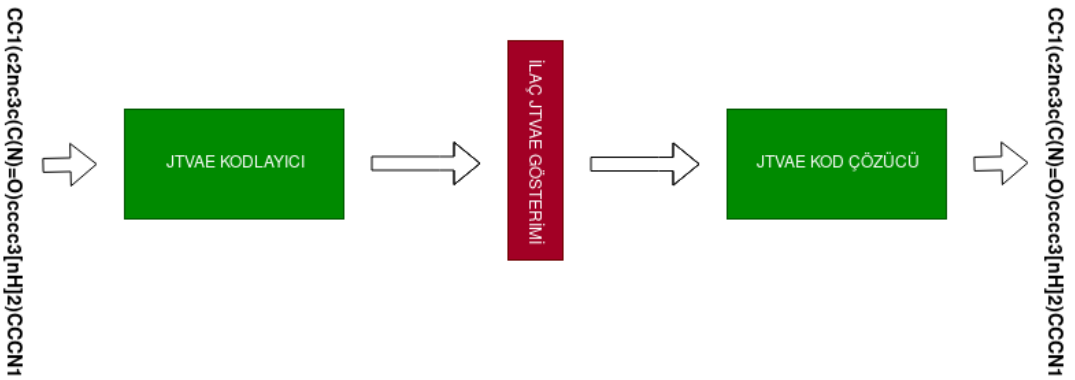
[49] çalışmasında, geliştirdikleri sistemde ZINC veri tabanından[50] 250000 moleküle eğitilen oto-kodlayıcının gizli vektörleri moleküler bir özelliği tahmin etmek için kullanılmıştır. Oluşturdukları oto-kodlayıcı özyineli yapay sinir ağlarından oluşmaktadır. Bu oto-kodlayıcı girdi olarak aldığı molekülleri bir girdi vektörüne kodlar. Bu kodlanan vektör(gizli vektör), daha sonra oto-kodlayıcının mimarisinde bulunan kod çözücülerle tekrar girdi olarak alınan vektöre çevrilmeye çalışılır. Bu çalışmada ise gizli vektör, moleküler bir özelliği tahmin etmek için kullanılırken bir Gaussian yöntemden[51] yararlanılmıştır. Bu Gaussian yöntemde, özelliği nasıl tahmin edileceği öğrenilirken, aynı zamanda bu yönteme girdi olarak verilen vektör güncellenmiştir. Bu güncellenen vektörler daha sonra kod çözücülere girdi olarak verilip molekül oluşturulmuştur. Bu

sayede kod çözücülerin oluşturduğu uygun moleküllerin sayısının arttığı gözlemlenmiştir. [52] çalışmasında yarı gözetimli bir oto-kodlayıcı kullanılarak, belirli bir moleküler özelliği sağlayan moleküller üretilmeye çalışılmıştır. Oluşturdukları oto-kodlayıcı özyineli yapay sinir ağlarından oluşmaktadır.Kodlayıcı tarafında kullanılan özyineli yapay sinir ağındaki ağlar iki taraflı iken, kod çözücü tarafındaki ağlar tek taraflıdır. Oto-kodlayıcı sayesinde oluşturulan gizli vektörler, molekül özelliğini tahmin etmek için kullanılır. Daha sonra özelliği tahmin ederken alınan hata ile kod çözücünün, girdi olarak alınan molekülü tekrar oluştururken aldığı hata birleştirilir ve bütün sistem buna göre geri beslenir. Bu çalışmada ZINC veri tabanından[50] alınan 310000 molekül kullanılmıştır. Kod çözücüde uygun molekül oluşumunu arttırmak için ışın arama (beam search) kullanılmıştır.

[53] çalışmasında koşullu oto-kodlayıcı kullanılmıştır. Bu çalışmada, oto-kodlayıcı ile öğrenilen gizli vektörle, beş tane moleküler özellik tahmin edilmeye çalışılıyor. Moleküler özellikler tahmin edilirken, bir Gaussian yöntem kullanılmıştır. Kullanılan bu Gaussian yöntemde, özellikleri nasıl tahmin edileceği öğrenilirken, aynı zamanda bu yönteme girdi olarak verilen vektör güncellenmiştir. Bu güncellenen vektörler daha sonra kod çözücülere girdi olarak verilip molekül oluşturulmuştur. Bu çalışmada, tahmin edilmeye çalışılan moleküler özellikler oto-kodlayıcının, kodlayıcısına verilen girdi vektörleriyle birleştiriliyorlar. Dolayısıyla, bu moleküler özellikler gizli vektör oluşumunu etkilemektedir ve tüm sistem moleküler özellik tahmininden alınan hatalar ile kod çözücüden alınan hatalara göre geri beslenmektedir.

4.2 Veri kümesi

3.2 kısmında belirtildiği gibi, tezin ikinci aşaması için oluşturulan veri kümesinde ilk aşamada olduğu gibi, herbir öge için öznitelikler, birinci ilacın öznitelikleri-ikinci ilacın öznitelikleri-hücre hattının öznitelikleri şeklinde sıralandı. Tekrar ilk aşamada olduğu gibi, yapay öğrenme yöntemlerinin ilaçların sırasından etkilenmemesi için, herbir öge için öznitelikler, ikinci ilacın öznitelikleri-birinci ilacın öznitelikleri-hücre hattının öznitelikleri şeklinde yeniden sıralanıp veri kümesine eklendi. Bu çalışmanın ikinci kısmı için kullanılan ilaç gösterimleri : [54]'da anlatılan ve gözetimsiz bir şekilde eğitilen junction tree variational autoencoder (JTVAE) modelindeki kodlayıcılar tarafından oluşturulan gizli vektörlerdir (Şekil 4.1). İlk kısımdan farklı olarak JTVAE tarafından oluşturulan ilaç gösterimlerini kullanmamızın sebebi, bu ilaç gösteriminin, diğer gösterimlerden farklı olarak, her ilaç için herhangi bir kısıt olmadan kolayca elde edilebiliyor olmasıdır. Bu aşamadaki sinerji skoru tahmini yapan yapay öğrenme yöntemi için gerekli veri kümeleri oluşturulurken, JTVAE kullanılarak elde edilen ilaç gösterimleri, ilk aşamada kullanılan hücre hattı öznitelikleri ile birleştirildi.



Şekil 4.1: İlaçların JTVAE Gösteriminin Oluşturulması

4.2.1 Öznitelikler

Bu bölümde, ikinci aşamada kullanılan ilaç özniteliklerini çıkarmak için kullanılan yöntemler ve prosedürler anlatılmıştır. Yukarıda belirtildiği gibi bu aşamada oluşturulan veri kümesindeki hücre hattı öznitelikleri, ilk aşamadaki hücre hattı öznitelikleriyle aynıdır. Hücre hattı özniteliklerinin nasıl oluşturulduğu 2 kısmında anlatılmıştır.

1. **İlaç Öznitelikleri**:

   Çalışmalarımızın ikinci kısmında ilaç gösterimleri için kullanılan JTVAE Gösterim lerinin nasıl oluşturulduğu bu bölümde detaylı bir şekilde açıklanmıştır.

   (a) **JTVAE Gösterimi**:

   [54]'da oto-kodlayıcı kullanarak, girdi olarak verilen çizgeleri, belirli uzunluk taki sürekli vektörlere gömüp, gömülen bu vektörlerden geçerli moleküller üretmeyi amaçlamışlardır.

   Bu amaç için ilk olarak, molekül çizgelerini, nodal (junctional) ağaçlara çevirmişlerdir. Nodal ağaç kullanmalarının sebebi, molekül oluşturulurken, oluşan molekülün geçerli olup olmadığının aşamalı bir şekilde kontrol edilebiliyor olmasıdır.

   Nodal ağaç oluşturma: Molekül çizgelerini, nodal ağaçlara çevirirken, ilk olarak bir sözlük oluşturulur. Bu sözlük, girdi molekül çizgelerinde bulunan ve farklı çember(ring), atom, kenarları kapsayan öbekleri içerir (kısacası öbekler molekül çizgesinin alt-çizgeleridir.). Bir molekül çizgesini, öbeklere ayırırken, öbekler arasında iki atomdan fazla kesişen atom olmamasına dikkat edilmiştir. Herhangi bir öbeğe dahil olmayan ve çember oluşturan kenarlar silinmiştir. Öbeklere ayrılan çizgede, kesişen öbekler arasına kenar eklenip, tüm öbekleri kapsayan ağaç çıkarılır ve nodal ağaç oluşturulur.

Kodlayıcılar: JTVAE modelinde, ağaç gömülümünü ve çizge gömülümünü öğrenen iki kodlayıcı bulunur.

i. **Çizge Kodlayıcı**: Girdi olarak alınan molekül çizgelerinde, herbir düğümün (v), x_v olarak gösterilen bir öznitelik vektörü bulunmaktadır. Aynı zamanda her bir kenarın da, x_{uv} olarak gösterilen bir öznitelik vektörü bulunmaktadır. Kenar öznitelikleri, kenar çeşidi ile o kenarın düğümlerinin birbirine gönderdiği mesajlardan oluşur (bu mesajları genel olarak v_{uv} ve v_{vu} şeklinde gösterebiliriz.).
Molekül çizgesinin gömülümü bir mesajlaşma prosedürü ile öğrenilir. Bir düğümün t zamanında yolladığı mesaj 4.1. formülle gösterilmiştir.

$$v_{uv}{}^{(t)} = \tau(W_1^g x_u + W_2^g x_{uv} + W_3^g \sum_{w \in N(u) \setminus v} v_{wu}^{(t-1)}) \tag{4.1}$$

Bu formülde τ Relu aktivasyon fonksiyonunu, $N(u)$ ise u düğümünün komşularını gösterir.
Yukarıdaki formülden anlaşılacağı gibi bir düğümden yollanan mesajı güncellemek için, diğer komşularından gelen mesajlar, üzerinde bulundu ğu kenarın özniteliği ve kendi öznitelikleri (düğüm öznitelik vektörü) bir araya getirilmiştir. Belirli sayıda iterasyondan sonra her bir düğümün en son öznitelik vektörü 4.2. formülle hesaplanır.

$$h_u = \tau(U_1^g x_u + \sum_{v \in N(u)} U_2^g \mathbf{v}_{vu}^{(T)}) \tag{4.2}$$

Yukarıdaki formülde u düğümü için komşularından gelen mesajlar toplanıp, düğüm öznitelikleriyle birleştirilmiştir. Relu fonksiyonundan geçirildikten sonra u düğümü için gösterim (latent vector) elde edilmiştir. Tüm çizgenin gömülümü; tüm düğümlerin gizli vektörlerinin (düğüm gösterimleri) ortalaması alınıp, ortalama sonucunda oluşan sürekli vektöre tüm düğümlerin log varyansı eklenerek elde edilir. Herhangi bir çizgenin gömülümü z_G şeklinde ifade edilir.

ii. **Ağaç Kodlayıcı**: Çizge kodlayıcıda olduğu gibi, girdi olarak verilen ağacın tüm düğümlerinin öznitelik vektörleri vardır. Herhangi bir i düğümünün öznitelik vektörünü x_i olarak gösterebiliriz. Nodal ağaç oluşturma bölümünde anlatıldığı gibi; nodal ağaç, öbeklerden oluşur. Bir i düğümünün x_i vektörü; bizim sözlüğümüzdeki hangi öbek olduğunu gösteren iki tabanında vektörlerdir.
Ağaç kenarlarının öznitelikleri, bir kenarın iki ucundaki düğümlerin

birbirine yolladığı mesajlardan oluşur.
Ağaç kodlayıcıda, çizge kodlayıcıdan farklı olarak, düğümlerin birbirine mesaj göndermesi, aşağıdan yukarı olacak şekilde tek bir iterasyonda gerçekleşir. Mesajlar, GRU fonksiyonu ile oluşturulur.

$$m_{ij} = GRU(x_i, \{m_{ki}\}_{k \in N(i) \setminus j}) \tag{4.3}$$

Aşağıdaki formüllerde bu GRU fonksiyonunda izlenilen prosedürünün nasıl gerçekleştiği verilmiştir.

$$s_{ij} = \sum_{k \in N(i) \setminus j} m_{ki} \tag{4.4}$$

$$z_{ij} = \sigma(W^z x_i + U^z s_{ij} + b^z) \tag{4.5}$$

$$r_{ij} = \sigma(W^T x_i + U^T m_{ki} + b^T) \tag{4.6}$$

$$\widetilde{m_{ij}} = tanh(W x_i + U \sum_{k \in N(i) \setminus j} r_{ki} \odot m_{ki}) \tag{4.7}$$

$$m_{ij} = (1 - z_{ij}) \odot s_{ij} + z_{ij} \odot \widetilde{m_{ij}} \tag{4.8}$$

Yukarıdaki formülde, i ve j düğümünleri arasındaki mesaj m_{ij} ile gösterilir. Fark edileceği gibi, herhangi bir bir i düğümünün, bütün çocuklarından mesaj gelmeden, i düğümünün mesajı yollanmaz.
Çizge kodlayıcıda izlenilen yönteme benzer şekilde; her bir i öbeğinin gizli vektörü (gösterimi) 4.9. formülle son haline getirilmiştir.

$$h_i = \tau(W^o x_i + \sum_{k \in N(i)} U^o m_{ki}) \tag{4.9}$$

Daha sonra tüm ağacın gömülümü, gene çizge kodlayıcıda olduğu gibi, tüm öbeklerin gizli vektörlerinin (gösterimlerinin) ortalaması alınıp, ortalama sonucunda oluşan sürekli vektöre tüm düğümlerin log varyansı eklenerek elde edilir. Herhangi bir nodal ağacın gömülümü z_T şeklinde ifade edilir.

Tez çalışmasının ikinci aşamasında, sinerji skoru tahmini yapmak için kullanılan yapay öğrenme modelleri ile kullanılan veri kümesindeki ilaç gösterimleri ; JTVAE modelinin ağaç ve çizge kodlayıcılarının ürettiği z_G ve z_T vektörlerinin birleşimidir. Kod çözücünün hatasına göre, eğitim

boyunca parametreleri güncellenen kodlayıcılar tarafından oluşturulan z_G ve z_T birleştir ve belirli normalizasyon işlemleri uygulandıktan sonra, bu gösterim yapay öğrenme modelleri ile çalışmaya uygun hale getirilir.

Tezin ikinci aşamasında, JTVAE'nin birleşenleri olan kod çözücülerden yararlanarak kendi oluşturduğumuz ilaç gösterimlerinden ilaç SMILES dizileri üretilmiştir. Bu kod çözücülerin nasıl çalıştığı aşağıda anlatılmıştır.

Kod Çözücüler: Kodlayıcı kısmında olduğu gibi, JTVAE modelinde, ağaç kod çözücü ve çizge kod çözücü olmak üzere iki farklı kod çözücü bulunmaktadır.

(a) **Ağaç Kod Çözücü**: Ağaç kod çözücü yukarıdan aşağıya doğru bir akış ile çalışır. Eğitim aşamasında; girdi olarak gelen ağaç derin öncelikli arama ile gezilir. Bu derin öncelikli arama ile her bir düğümün tüm çocukları ve bu çocuklara uğrama sırası belirlenir. Bu belirlemeden sonra girdi olarak verilen ağacın derin öncelikli arama yolları için aşağıdaki GRU fonksiyonu çalıştırılır.

$$h_{i_t,j_t} = GRU(x_{i_t}, \{h_{k,i_t}\}_{k,i_t} \in \tilde{\varepsilon}_t, k \neq j_t) \tag{4.10}$$

Bu GRU fonksiyonu, ağaç kodlayıcıda kullanılan GRU fonksiyonunun aynısıdır. x_{i_t}, GRU fonksiyonuna girdi olarak verilen derin öncelikli arama yolunun her bir düğümün sözlükte bulunma sırasının gömülümüdür. h_{k,i_t} ise, derin öncelikli arama yolundaki her bir düğümün, bu yoldaki çocuğu olmayan diğer komşularının sözlükte bulunma sırasını gösterir.

Ağaç kod çözücünün akışı sırasında yapılan iki tahmin vardır. Bunlardan ilki; herhangi bir düğümün çocuğu olup olmadığıdır, ikincisi ise; herhangi bir düğümün çocuğu olan öbeği(daha önce belirtildiği gibi ağaçların her bir düğümü sözlükte bulunan bir öbektir) tahmin etmektir.

İlk tahmin için aşağıdaki formül kullanılır;

$$p_t = \sigma(u^d.\tau(W_1^d x_{i_t} + W_2^d z_T + W_3^d \sum_{(k,i_t)\in\tilde{\varepsilon}} h_{k,i_t})) \tag{4.11}$$

bu formülde; z_T ağaç kodlayıcı çıktısıdır ve aynı GRU fonksiyonundaki gibi x_{i_t}, girdi olarak verilen derin öncelikli arama yolunun her bir düğümün sözlükte bulunma sırasının gömülümüdür. h_{k,i_t} ise, derin öncelikli arama yolundaki her bir düğümün, bu yoldaki çocuğu olmayan diğer komşularının sözlükte bulunma sırasını gösterir. Girdi olarak gelen derin öncelikli arama yolundaki her bir ana-çocuk düğümler için bu fonksiyonun çıktısı, yolun

ana düğümden mi çocuk düğüme, yoksa çocuk düğümden mi ana düğüme olduğunu gösterir. Dolayısıyla bir ikili sınıflandırma problemi çözer. Bu fonksiyonun hatası(topolojik hata), JTVAE'yi geri besleme için kullanılan hata fonksiyonuna eklenir.

İkinci tahmin için aşağıdaki formül kullanılır;

$$q_j = softmax(U_T^l(W_1^l z_T + W_2^l h_{ij})) \tag{4.12}$$

bu formülde: z_T ağaç kodlayıcı çıktısıdır, h_{i_t,j_t} GRU fonksiyonunun çıktısıdır. Topoloji tahmininde olduğu gibi, öbek tahmini için kullanılan bu fonksiyonun da hatası(öbek hatası) da, JTVAE'yi geri besleme için kullanılan hata fonksiyonuna eklenir.

Geri besleme ile GRU'daki, topoloji ve öbek tahminindeki ağırlık matrisleri öğrenilir. Ağaç oluşturma aşamasında öğrenilen parametreler ile kodlayıcı çıktısı kullanılarak, ağacın kökü oluşturulur ve iteratif bir şekilde öbek tahmini yapılarak(4.11. formül kullanılarak) çocuk öbekler oluşturulur. Çocuk öbekler oluşturulurken geçerli bir ağaç oluşup oluşmadığı kontrol edilir. Öbek tahmini için kullanılan fonksiyon, JTVAE için oluşturulan sözlükteki bütün öbeklere bir olasılık atar. Bu olasılıklara göre, iterasyondaki ana düğümün çocuğu olup olmayacağına karar verilir. Eğer çocuğu olacaksa, en yüksek olasılıktaki beş öbek için geçerli bir ağaç oluştup oluşmadığına bakılır. Eğer geçerli ağaç oluşursa bir sonraki iterasyondaki ana düğüm, belirlenen çocuk düğüm olur. Eğer geçerli ağaç oluşmazsa veya düğüm için çocuğu olmaz şeklinde bir tahmin yapılırsa, iterasyona ana düğümün diğer bir çocuğuyla devam edilir.

(b) **Çizge Kod Çözücü**: JTVAE'de molekül çizgesi oluşturmak için en son olarak, ağaç kod çözücü tarafından oluşturulan nodsal ağacı, çizge haline çevirmek için çizge kod çözücü kullanılır. Çizge kod çözücü, JTVAE'nin diğer birleşenleri gibi kararlı(deterministic) değildir. Bunun sebebi birden fazla farklı molekül çizgesinin, aynı nodsal ağaca sahip olabilmesidir. Çizge kod çözücüde de ağaç kod çözücüde olduğu gibi yukarıdan aşağıya bir akış izlenir (öbekleri birleştirmeye ilk kök ve onun çocuklarından başlanır.). Çizge çözücünü temel olarak yaptığı, nodsal ağaçtan oluşabilecek, her türlü farklı çizgeye bir olasılık vermesidir.

Ağaç kod çözücünün oluşturduğu nodsal ağacın kökünden başlanarak, ağaç kod çözücünün çıktısı olan ağaçtaki her bir düğümün çocukları çıkarılır. Daha sonra kökten başlanarak, her bir düğüm, çocuklarından biriyle birleştiril

meye başlanır (düğümlerin birleşme sırası derin öncelikli aramadaki gibidir.). Ağaçtaki herhangi bir düğümün, ilk *a* çocuğuyla birleşip, daha sonra *b* çocuğuyla birleşmesiyle; ilk *b* çocuğuyla birleşip daha sonra *a* çocuğuyla birleşmesi farklı çizgeler oluşturur. Bir ana düğümün hangi çocuğuyla birleşeceğine karar vermek için, her bir çocuk için bir olasılık elde edilir ve en yüksek olasılıklı çocukla ana düğüm birleştirilir. Bu olasılığı elde etmek için; aşağıdaki formül hesaplanarak elde edilen h_{G_i} sürekli vektörü ile çizge kodlayıcının çıktısı olan z_G vektörü çarpılır. Bu çarpım sonucu oluşan yeni vektörden, ana düğümün her bir çocuğu için istenilen olasılıklar çekilebilir.

Belirli bir aşamaya kadar öbeklerin birleştirilmesiyle oluşan G_i alt-çizgesinde, bir sonraki aşamada hangi öbek ile birleşeceğine karar vermek için, bir mesajlaşma prosedürü izlenir.Bu mesajların biraraya getirilmesiyle h_{G_i} vektörü oluşturulur. Mesajların nasıl oluşturulduğu aşağıdaki formüllerle gösterilmiştir:

$$\mu_{uv}^{(t)} = \tau(W_1^a x_u + W_2^a x_{uv} + W_3^a \tilde{\mu}_{uv}^{(t-1)}) \tag{4.13}$$

$$\tilde{\mu}_{uv}^{(t-1)} = \begin{cases} \sum_{w \in N(u) \setminus v} \mu_{wu}^{(t-1)}, & \alpha_u \alpha_v \\ \widehat{m}_{a_u,a_v} + \sum_{w \in N(u) \setminus v} \mu_{wu}^{(t-1)}, & \alpha_u \neq \alpha_v \end{cases} \tag{4.14}$$

Bu formülde *u* ile *v* farklı öbeklerde bulunan ve *u* dan *v* ye, ağaç kod çözücü aşamasında m_{α_u,α_v} mesajı yollanmış iki atomu gösterir. Çizge kodlayıcı bölümünde belirtildiği gibi x_u düğüm özniteliği, x_{uv} ise kenar özniteliğidir.

Eğitim aşamasında, birleştirilen düğümün, nodsal ağacın asıl derin öncelikli arama dolaşımında uğradığı asıl düğüm olup olmamasına göre bir hata değeri hesaplanır. Bu hata değeri JTVAE'yi geri besleme için kullanılan hata fonksiyonuna eklenir. Geri besleme ile h_{G_i} hesaplamada kullanılan ağırlık matrisleri öğrenilir.

Test aşamasında, ağaç kod çözücünün oluşturduğu nodsal ağacın kökünden başlanır ve yukarıda anlatılan olasılık hesabı kullanılarak derin öncelikli aramadaki sırayla düğümler birleştiril. Bu şekilde oluşturulan çizgenin molü hesaplanır. Rdkit kütüphanesi kullanılarak, bu mole sahip isomerik 3D SMILES dizileri elde edilir. Elde edilen bu dizilerin, çizge kodlayıcı kullanarak z_G vektörleri elde edilir. Asıl çizgenin z_G vektörüne, kosinüs benzerliği en fazla olan 3D SMILES, JTVAE modelinin çıktı SMILES dizisidir.

4.2.2 Veri kümelerilerine uygulanan ön işlemler

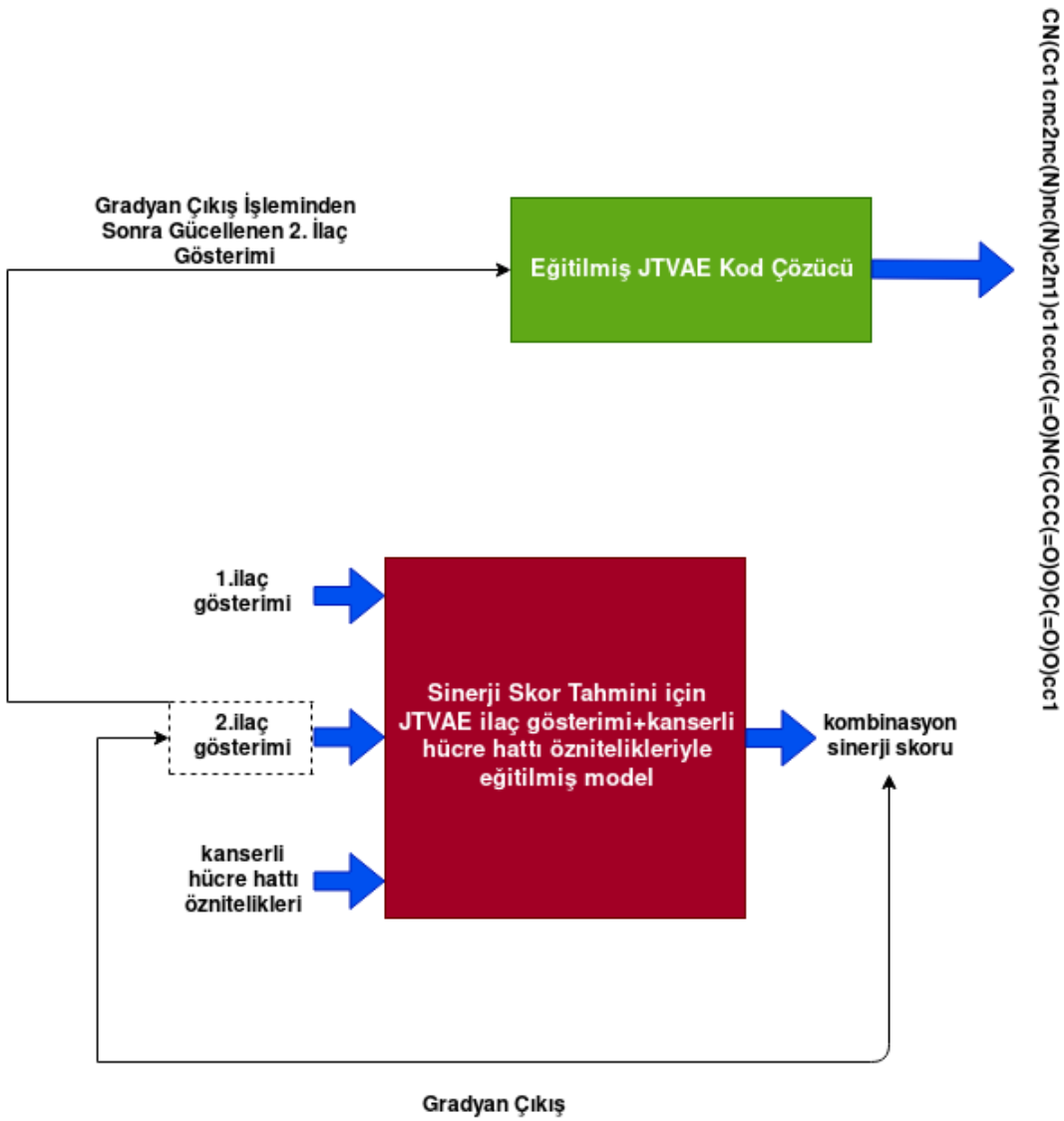
Çalışmanın ikinci kısmında, ilk çalışmadaki kısıtlarımız olmadığı için [39]'daki veri kümesindeki tüm ögeler kullanıldı. Ögelerin nasıl kullanıldığı 4.3 bölümünde ayrıntılı bir şekilde anlatıldı. JTVAE modelinin kodlayıcı çıktısı, sinerji skoru tahmini yapmak için normalize edilip yapay öğrenme modeline verildi.
İkinci aşamadaki ilaç üretimi kısmı olarak, JTVAE modelinin kod çözücü kısımları kullanıldı. Kod çözücünün doğru bir şekilde çalışabilmesi için, kod çözücüye girdi olarak ilaçların, yapay öğrenme modelini eğitmek için kullanılan normalize edilmiş hallerini değil; orijinal hallerini vermek gerekir. Normalizasyon işleminde, ilk aşamadan farklı olarak, her sürekli (continous) değerin arctanh değeri olmadığı için tanh normalizas yonu yerine, standardizasyon yöntemi kullanmanın daha uygun olacağı sonucuna varıldı. Dolayısı ile, ilaç üretmek için, sinerji skoru tahmini yapan yapay öğrenme modeliyle bulunan sürekli(continous) JTVAE gösterimlerilerine, standardizasyon işlemi tersten uygulanarak JTVAE vektörleri doğru aralıklara(scale) çekilip, kod çözücü ile çalışmaya uygun hale getirildi. Uygun hale getirilen sürekli vektörler ve JTVAE modelinin kod çözücüleri kullanılarak, bu aşamadaki amacımıza uygun molekülleri oluşturduk.

4.3 Sinerji skoru optimizasyonu

Yukarıda tezimizin ikinci aşamasında kullanılan ilaç gösterimlerinden, bu ilaç gösterimle rini oluşturmak için kullandığımız yöntemlerden ve ikinci aşamada kullanmak için düzenlediğimiz veri kümesinden söz ettik. Bu bölümde ise ikinici aşamadaki amacımız ve kullandığımız birleşenler detaylıca anlatıldı. Aynı zamanda bu bölümde amacımızı gerçekleştirmek için belirlediğimiz birleşenleri ve oluşturduğumuz veri kümelerini nasil biraraya getirdiğimiz açıklandı. Sinerji Skoru optimizasyonu için geliştirdiğimiz yöntem Şekil 4.2'de görselleştirilmiştir.

4.3.1 Sinerji skoru optimizasyonu amacı, girdi ve çıktıları

Bu aşamadaki amacımız,sinerji skoru tahmini için eğitilmiş bir yapay öğrenme modelinin tahminini en iyileyecek girdi vektörünü bulmaktır.
Sinerji skorunu en iyileyecek girdi vektörü oluşturulurken; 4.2 kısmında bahsedildiği gibi yapay öğrenme modellerine verilen girdi vektörleri; birinci ilacın öznitelikleri-ikinci ilacın öznitelikleri-hücre hattı öznitelikleriindan oluşur. Bu aşamada yaptığımız en iyileme işleminde; birinci ilaç ve hücre hattı öznitelikleri sabit tutulup, sadece kombinasyondaki ikinci ilacın öznitelikleri güncellendi.
Dolayısıyla bu aşamanın sonucunda, bir ilaç-hücre hattı ikilisi için; eğitilen yapay öğrenme modelinin tahmin ettiği skoru en iyileyen ikinci ilaçlar bulunmaktadır.



Şekil 4.2: JTVAE İlaç Gösterimleriyle Sinerji Skoru Optimizasyonu

4.3.2 Sinerji skoru optimizasyonu için izlenilen yöntem

Bu iş için Şekil 4.2'da gösterilen akış izlendi. Temel olarak uyguladığımız yöntem; elimizde belirli ilaç gösterimleriyle eğitilmiş ve sinerji skoru tahmini yapan bir yapay öğrenme modeli bulunmaktadır. Bu modele, tezin ilk çalışmasında olduğu gibi, birinci ilaç gösterimi-ikinci ilaç gösterimi-hücre hattı gen anlatımı öznitelikleri şeklinde bir girdi vektörü verildi. Bu vektördeki ikinci ilaç gösterimini, gradyan çıkış yöntemiyle güncelleyerek yapay öğrenme modelinin verebileceği maksimum çıktıya ulaşılmaya çalışıldı. Gradyan çıkış işleminden sonra elde edilen ilaç gösterimini verebilecek SMILES dizini belirlenilmeye çalışıldı.

Şekil 4.2'da anlaşılacağı gibi gradyan çıktı sonunda oluşan ilaç gösterimlerinin kolayca SMILES dizilerine çevrilebiliyor olması gerekmektedir. Bu sebepten, daha önce 4.2

bölümünde belirtildiği gibi, ilk aşamadan farklı olarak, bu aşamadaki yapay öğrenme modelleri ile kullanılan veri kümesindeki ilaç gösterimleri JTVAE modeli kullanılarak elde edildi. Bu ilaç gösterimlerinin nasıl elde edildiği 1a bölümünde anlatılmıştır. Bu aşamada CD,Chem, GNN ilaç gösterimlerinin kullanılmamasının sebebi; bu ilaç gösterimlerinin özniteliklerinden oluşan herhangi bir vektörün, hangi SMILES dizisine ait olduğunu bulmak masraflı ve belirsiz (ambigous) bir işlemdir (Bir CD vektörünü veren SMILES dizini ancak [45]'da varsa bulunabilir eğer yoksa aranan ilaç için tüm laboratuvar deneyleri tekrarlanmalıdır. Chem gösterimindeki ECFP vektörleri ve GNN gösterimindeki molekül vektörleri, molekülün rastgele bir atomundan başlanarak oluşturulduğu için tekrar SMILES dizisine çevirmeye uygun öznitelikler değildir.).Ancak herhangi bir JTVAE gizli vektörünü, JTVAE modelinin kod çözücülerine verdiğimizde %100 olasılıkla uygun bir SMILES dizini elde edilir.

Çalışmalarımızda kullanılan JTVAE modeli, [55]'daki ve [45]'den alınan, LINCS L1000 veri kümesine CD yöntemi uygulanarak çıkarılmış gen imzaları veri kümelerindeki SMILES dizinleriyle eğitilmiştir.

Bu aşamadaki yapay öğrenme modelleri için eğitim ve test verileri oluşturulurken; 3.2 bölümünün başında anlatıldığı gibi [39]'daki veri kümesindeki ögeler birinci ilaç-ikinci ilaç-hücre hattı kombinasyonlarından oluşuyordu. Bir ögedeki kombinasyonda, eğer her iki ilacın da CD gösterimi bulunuyorsa, bu öge eğitim verisine dahil edildi. Diğer taraftan, bir ögedeki kombinasyonda, eğer her iki ilacın da CD gösterimi bulunmuyorsa, bu öge test verisine dahil edildi. Test verisindeki herhangi bir ilacın, eğitim verisi tarafından görülmemesi için test ve eğitim verileri bu şekilde oluşturuldu. İki veri de standardizasyon yöntemiyle normalize edilmiştir. Yapay öğrenme modellerinin parametre optimizasyonu için, eğitim verisinden rastgele seçilen ve eğitim verisinin %10'unu oluşturan ögeler değerlendirme verileri olarak kullanıldı. Her bir yapay öğrenme modeli için test verisinden alınan ortalama karesel hata değerine göre gradyan artırma, JTVAE ilaç gösterimleriyle oluşturulan veri kümesi için en iyi performans gösteren modeldir. Dolayısıyla gradyan çıkış aşamasında, çıktısını en iyileyeceğimiz model gradyan artırmadır.

Gradyan çıkış aşaması için kullanılacak kombinasyonlar belirlenirken; test verisindeki gradyan artırma modelinin hatasının en az olduğu ilk beş yüz öge tespit edilmiştir. Şekil 4.2'te de gösterildiği gibi, yapay öğrenme modelinin çıktısını en iyilemek amacıyla, kombinasyondaki sadece ikinci molekülü güncelledik. Daha kararlı bir sonuç elde etmek için, hatası en az olan ilk beş yüz kombinasyondan, sadece, ikinci molekülü, JTVAE tarafından %100 yeniden oluşturulabilen ögeler dikkate alındı. Belirlenen ilaç kombinasyonlarıyla deneyler yapılırken, çeşitli başlangıç ilaç gösterimleri ve benzerlik fonksiyonları kullanıldı.

Gradyan çıkış aşamasında; eğitilmiş yapay öğrenme modellerinin çıktısını en iyileyecek, girdi molekülü bulunmaya çalışılır. Dolayısıyla ikinci ilaç gösterimi değişkendir. Bu gösterim, belirli değerlerle başlatıldı ve belirli sayıda iterasyon boyunca, gradyan adımlarla güncellendi. Çalışmamızdaki gradyan adımların nasıl hesaplandığı 4.15. formülle verilmiştir.

$$z_{gradyan} = \frac{F(z(t) + \Delta z) - F(z(t))}{\Delta z} \tag{4.15}$$

Bu gradyanlarla kombinasyonlardaki ikinci ilaç gösterimlerinin nasıl güncellendiği 4.16. formülle gösterilmiştir.

$$z(t+1) = z(t) + \alpha z_{gradyan} \tag{4.16}$$

Yukarıdaki formüllerde F gradyan artırma modelini gösterir. $z(t)$ ikinci molekülün, güncellemeden önceki gösterimi, $z(t+1)$ aynı molekülün, güncellemeden sonraki gösterimidir. Δz ; ilaç gösterimindeki değişim, α ise öğrenme katsayısıdır. Bu aşamadan sonra güncellenen ilaç gösterimleri, JTVAE modelindeki kod çözücülere verilerek SMILES dizileri elde edilir.

Bu aşamada, anlatılan şekilde molekül oluşturarak, JTVAE, gradyan artırma ve gradyan çıkış yöntemlerinin bu amacımız için uygun olup olmağı test edildi. Yaptığımız bu sisteme göre, bir modelin çıktısını en iyileyen molekül ve bu molekülün analizi, (veya istenilen sinerji skorunu sağlayan molekül) JTVAE modelinin eğitildiği SMILES dizilerine, gradyan çıkarma fonksiyonun parametreleri vs. gibi değişkenlere göre çeşitlilik gösterebilir. Bu sebepten gradyan çıkış aşamasında, her iterasyondan sonra oluşan ilaç gösterimi kaydedildi. Bu aşamadaki sonuçlar, her aşamada kaydedilen SMILES dizinlerine ve ilaç gösterimlerine göre analiz edildi.

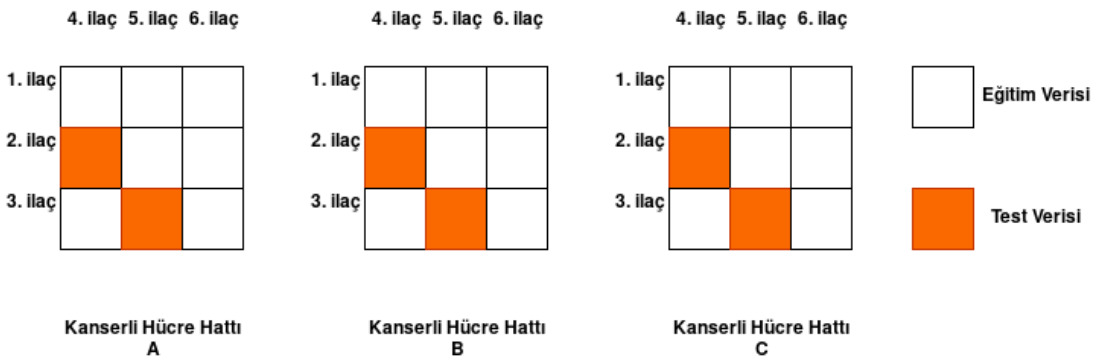
5. DENEY SONUÇLARI

5.1 Çapraz doğrulama ve istatistiksel testler

Tezin ilk aşamasında tahmin etmeye çalıştığımız sinerji skorları, ikili ilaç kombinasyonla rı ve hücre hatları gibi değişkenlere bağlı olduğu için, çapraz doğrulama bu değişkenleri göz önünde bulundurarak düzenli hale getirildi. Şekil 3.1'de de görüleceği gibi test grubundaki ögeler; bazı ilaç kombinasyonlarının, onkoloji veri kümesindeki tüm hücre hatlarına etkisini gösteren verilerdir. Dolayısıyla, çapraz doğrulamada ögeler gruplandırı lırken, eğitim verisinin, test verisindeki ilaç kombinasyonlarını görmemesine dikkat edildi. Belirlenen çapraz doğrulamanın nasıl yapıldığı Şekil 5.1'de görselleştirilmiştir. Bu sayede, üzerinde çalıştığımız modeller, görmedikleri ilaç kombinasyonlarına, belirli hücre hatlarının verecekleri tepkileri tahmin etme performanslarına göre karşılaştırılır.

Bu şekilde bir çapraz doğrulama yapılmasının bir başka sebebi de çalışmamızdan alınan sonuçların karşılaştırıldığı ve şimdiye kadarki en iyi yöntemler olan DeepSynergy[7] ve TreeCombo'nun[8] da bu şekilde çapraz doğrulama yapmasıdır. Bu şekilde yapılan düzenli çapraz doğrulama ile ilk aşamada kullanılan veri kümelerileri beş gruba(folda) bölündü. Bu üç veri kümesi ile çalıştırılan yapay öğrenme modellerinin performansları beş gruptan alınan değerlerin ortalamasına ve standard sapmasına göre karşılaştırıldı. Buna ek olarak herbir gruptan alınan değerleri Wilcoxon Signed-Rank istatistiksel testine sokup, yapay öğrenme modellerinin performansları arasında belirli bir fark olup olmadığını incelendi. Bu testi kullanmamızın sebebi grup başına alınan değerlerin belirli bir dağılımın olmaması ve üç veri kümesindeki test gruplarının, aynı kombinasyonlardan oluşuyor olmasıdır.

Tezin ikinci aşamasında çıktıyı optimize eden, girdi bulmaya çalışırken, herhangi bir modelle performans karşılaştırması yapılmadı ve bu aşamada kullanılan bazı modeller gözetimsiz şekilde eğitildi. Aynı zamanda bu aşamada kullanılan yöntemler, daha fazla veri ile eğitilince, bu aşamadaki amacımız için daha kullanışlı hale gelmiştir. Bu sebeplerden dolayı, ikinci aşama için çapraz doğrulama ve istatistiksel testlerin kullanılmasına gerek kalmamıştır.



Şekil 5.1: İlaç Kombinasyonları ile Çapraz Doğrulama[7]

5.2 Sinerji skoru tahmin deneyleri

Parametre optimizasyonu, referans aldığımız çalışmalardan afarklı yapılmıştır. [7] ve [8] çalışmalarında, çapraz doğrulamdaki herbir grupta, bütün veri kümesi eğitim, test ve değerlendirme olarak üç kola ayrılmıştır. Eğitim verisi ile farklı parametrelerle eğitilen model, değerlendirme verisi üzerinde denenip, en iyi parametreler bulunmuştur. Bulunan bu parametrelerle eğitilen model, test verisi için tahminler yaparak, çapraz doğrulamada çalışılan grup için sonuçlar elde edilir. Fakat bu şekilde yapılan parametre optimizasyonu çok vakit ve kaynak tükettiği için, bizim deneylerimizde farklı bir yöntem izlenmiştir.

Deneylerimizde, bütün modellerin parametre optimizasyonu yapılırken, birinci ilaç-ikinci ilaç-kanserli hücre hattı kombinasyonlarından rastgele seçilen kombinasyonlar, parametre optimizasyonu için değerlendirme verisi olarak kullanıldı. Değerlendirme verisini oluşturan ögelerin sayısı, tüm kombinasyonların %10'udur. Deneylerimizdeki tutarlılığı korumak için, rastgele seçilen kombinasyonlar tüm model-veri kümesi ikilileri için aynıdır. Gradyan artırma, rastgele ağaç ve elastik ağ modellerinin parametre arama uzayı, TreeCombo[8] çalışmasına göre oluşturuldu. Yapay sinir ağlarında ise parametre uzayı belirlenirken DeepSynergy[7] çalışması referans alındı. İlk aşamadaki herbir veri kümesi-model kombinasyonu için en iyi skorları veren parametreler Çizelge 5.1'de verilmiştir. Elde ettiğimiz en iyi parametreler, referans aldığımız çalışmalardaki en iyi parametreler ile tutarlıdır.

Tezin ikinci aşamasınada, daha önceki kısımlarda söylenildiği gibi, ilk aşamadan farklı ögeler üzerinde çalışılmıştır. Test ve eğitim verileri 4.3 kısmında anlatılan şekilde belirlenmiştir. Bu aşamada sinerji skoru tahmini yapan modelin parametre optimizasyonu için; eğitim verisinden rastgele seçilen ögeler değerlendirme verisi için kullanılmıştır. Seçilen ögeler, tüm eğitim verisi ögelerinin %10'unudur. İlk aşamada olduğu gibi gene tüm eğitim verisinin %10'ununu kullanmamızın sebebi, deneylerimizi hızlandırmaktır. Değerlendirme verilerinin nasıl seçildiğini ve bu aşamadaki parametre optimizasyonu

sonuçlarını 5.4 ve 4.3 bölümlerinde detaylı bir şekilde anlatılmıştır.

Çizelge 5.1: Veri kümesi-model kombinasyonlarının parametre optimizasyonu sonuçları

veri kümesi-model komb.	en iyi parametreler
CDR-yapay sinir ağı	**seyreltme oranı:**0.4,**ilk katman nöron sayısı:**3000, **ikinci katman nöron sayısı:**1500,**iterasyon:**455
ChemR-yapay sinir ağı	**öğrenme katsayısı:**0.4,**ilk katman nöron sayısı:**3000, **ikinci katman nöron sayısı:**1500,**iterasyon:**455
GNNR-yapay sinir ağı	**iterasyon:**1000, **alt-çizgelerin yarı çapı:**2, **gömülüm vektörlerinin uzunluğu:**25, **çizge yapay sinir ağının katman sayısı:**3, **TBYSA ilk katman nöron sayısı:**3000, **TBYSA ikinci katman nöron sayısı:**1500
CDR-gradyan artırma	**maksimum derinlik:**6,**öğrenme katsayısı:**0.05, **iterasyon durdurumu kontrolü:**15
ChemR-gradyan artırma	**maksimum derinlik:**6,**öğrenme katsayısı:**0.05, **iterasyon durdurumu kontrolü:**15
GNNR-gradyan artırma	**maksimum derinlik:**2,**öğrenme katsayısı:** 0.01, **iterasyon durdurumu kontrolü:**15
CDR-rastgele ağaç	**maksimum derinlik:**8
ChemR-rastgele ağaç	**maksimum derinlik:**8
GNNR-rastgele ağaç	**maksimum derinlik:**4
CDR-elastik ağ	**alpha:**0.25
ChemR-elastik ağ	**alpha:**0.25
GNNR-elastik ağ	**alpha:**0.25

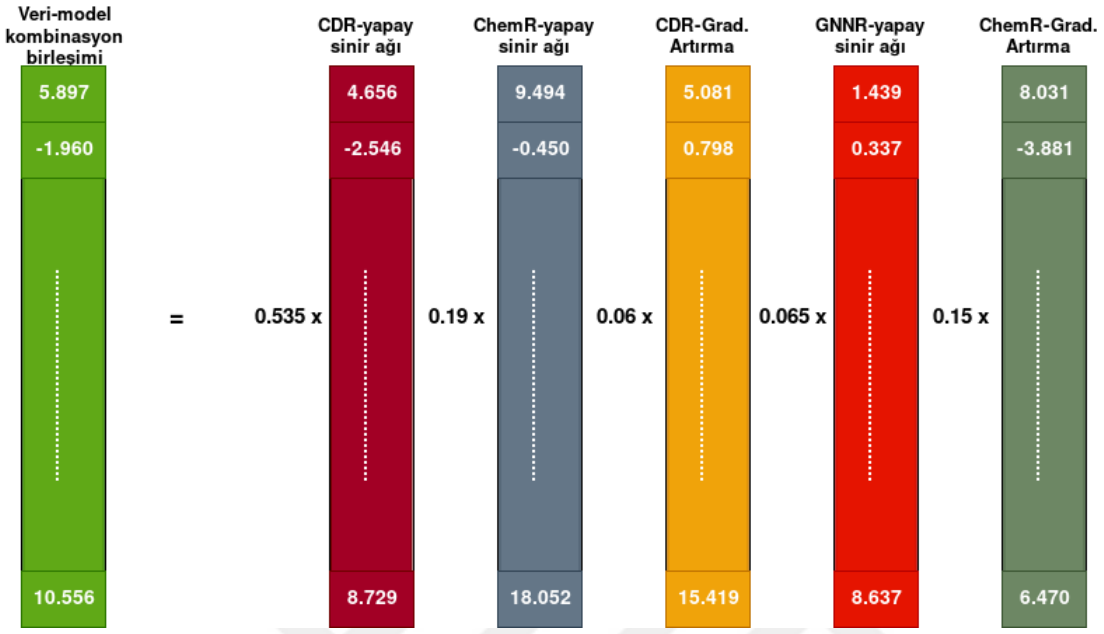
5.2.1 Model tahminlerinin birleştirilmesi

Herbir veri kümesi-model kombinasyonu için çapraz doğrulama sonuçları elde edildikten sonra en iyi performans gösteren ilk beş modelin tahminleri ağırlıklı ortalama yöntemi ile birleştirildi.

Ağırlıklı ortalama yöntemindeki herbir birleşenin ağırlıkları belirlenirken, birleştirme aşamasına dahil edilen herbir birleşen için $\{0.75, 0.25, 0.6, 0.4, 0.5, 0.15, 0.45, 0.1, 0.3, 0.2, 0.05, 0.06, 0.065, 0.535, 0.19, 0.22, 0.075, 0.125, 0.475\}$ ağırlık kümesinde baştan sona arama yapılmıştır. Birleşenlerin ağırlıkları toplamı 1.0 olan kombinasyonlar kaydedilmiştir. Veri kümesi-model kombinasyonları birleştirme işlemine dahil edilirken aç gözlü bir yaklaşım izlenmiştir. Ağırlık belirleme yönteminde, oluşan kombinasyon lardan daha kötü sonuç alınmaya başlandığında, birleştirme işlemine, daha fazla veri kümesi-model kombinasyonu tahmininin dahil edilmesi durdurulmuştur.

Veri kümesi-model kombinasyonları, 5.1 kısmında anlatılan çapraz doğrulama sonuçları na göre azalan sırada sıralanıp, bu sıraya göre birleştirme işlemine dahil edildi. Bu

adımlar izlendikten sonra en iyi sonuçları veren ilk beş model-veri kümesi kombinasyonu Şekil 5.2'de gösterilen ağırlıklarla birleştirilince, sinerji skoru tahmini için yaptığımız tüm deneyler arasındaki en iyi sonuçlar elde edildi.



Şekil 5.2: En iyi performans gösteren ilk beş modelin birleşimi

5.2.2 Sinerji skoru tahmin deneyleri sonuçları

CD, Chem, GNN ilaç gösterimleri ve gen anlatımı verileri [39]'daki onkoloji veri kümesine göre birleştirilip ,yapay öğrenme modelleriyle sinerji skoru tahmini için üç farklı veri kümesi oluşturuldu. Bu veri kümeleri 5.1 kısmında anlatılan şekilde beş farklı gruba bölündü. Bu bölünmeye göre, veri kümeleri 3.2.2 anlatılan ön işlemlerden geçirildi. Gruplara ayrılan ve ön işlemlerden geçirilen veri kümeleri ile elastik ağ(elas.a.), tam bağlı yapay sinir ağı(TBYSA), rastgele ağaç(RA), gradyan artırma(GA) modelleri sinerji skor tahmini yapmak için çalıştırıldı. Çizelge 5.2 ve 5.3'de, herbir veri kümesi-yapay öğrenme modeli kombinasyonunun, beş grup çapraz doğrulama işleminden sonra elde edilen beş ortalama hata karesi ve Pearson korelasyon değerlerinin ortalaması ve standart sapması gösterilmiştir. Bu sonuçlar elde edildikten sonra, herbir veri kümesi-model kombinasyonunun tahminleri 5.2.1 kısmında anlatılan yönteme göre birleştirildi. Bu işlem sonucunda herbir grup başına oluşan tahminlerle de Pearson korelasyon ve ortalama hata karesi hesaplanıp, bu değerlerin ortalaması ve standart sapması elde edilir. Veri kümeleri için, Çizelge 5.2 ve 5.3'de görülebileceği gibi CD ilaç gösterimiyle ortalama olarak en başarılı sonuçları aldık. Fakat, Wilcoxon Signed-Rank istatistiksel testi sonuçlarına göre, Chem ve CD gösterimleriyle herbir yapay öğrenme modelinden alınan sonuçlar arasında istatistiksel bir fark yoktur (İki ilaç gösterimi için herbir gruptan

alınan sonuçlar için Wilcoxon Signed-Rank testin sonucunda $p > 0.05$ çıkmıştır). İki ilaç gösteriminin de tercih edilebilirlik açısından negatif ve pozitif yönleri vardır; CD, Chem gösterimine göre daha kısa vektörlerden oluşur. Bu sebepten dolayı bu gösterimle çalışmak zaman ve alan(space) açısından daha verimlidir. Fakat, CD vektörlerini çıkardığımız [45]'de sadece 20339 ilacın CD vektörü bulunmaktadır, bunlardan farklı bir ilacın CD gösterimlerini elde etmek masraflı bir işlemdir (tüm laboratuvar deneyleri ve hiper düzlem hesaplamaları baştan yapılmalıdır.). Chem ise daha uzun vektörlerden oluşan bir ilaç gösterimi olmasına rağmen yeni bir ilaç için bu gösterimi elde etmek jCompound ve Chemopy kütüphaneleri kullanılarak yapılabileceği için CD'ye göre daha kolaydır. En kötü performans gösteren ilaç gösterimi GNN olmuştur. Bu ilaç gösteriminin CD ve Chem'e göre daha başarısız olmasının sebebinin sadece çizge topolojisi kullanmasından kaynaklandığı düşünüyoruz. CD ve Chem, ilacın hastalıklı hücre hatlarına etkisini, molekülün tepkimeye girme isteği, şekli vs. gibi sinerji skoru tahmini ile daha alakalı öznitelikler içerirken, GNN sadece rastgele oluşturulan r yarıçaplı altçizgelerin, 1c bölümünde anlatılan yöntemle elde edilen gömülümlerinden oluşur. GNN ilaç gösterimleri, tam bağlı yapay sinir ağı ile kullanılınca diğer modellere göre daha başarılı sonuçlar elde edilmiştir. Bunun sebebi diğer modellerden farklı olarak, yapay sinir ağı uçtan uca öğrenme sayesinde GNN ilaç gösterimlerini kendi hatasına göre düzeltebilmiştir. Diğer modeller ise, tam bağlı yapay sinir ağının hatasına göre düzeltilmiş GNN ilaç gösterimleriyle çalıştırıldı.

Modeller arasından tam bağlı yapay sinir ağının, diğer modellere göre, sinerji skor tahmini için belirli bir şekilde daha başarılı olduğu Çizelge 5.2 ve 5.3'de gözlemlenebilir. Kullandığımız tam bağlı yapay sinir ağının özellikleri göz önüne alındığında, oluşturdu ğumuz veri kümeleri ve tahmin etmeye çalıştığımız sinerji skorları arasında doğrusal olmayan bir ilişki olduğu anlaşılıyor.

Yukarıdaki analiz edilen veri kümesi-model sonuçlarına ek olarak, 5.2.1 bölümünde anlatıldığı gibi ağırlıklı ortalama yöntemiyle birleştirilen tahminlerle deneylerimizdeki en başarılı sonuçlar elde edildi. Aynı zamanda bu sonuçlar, bütün veri kümesi-model kombinasyonlarından alınan sonuçlardan istatistiksel olarak farklıdır (Herbir gruptan alınan değerler için Wilcoxon Signed-Rank testin sonucunda $p < 0.05$ çıkmıştır). Şekil 5.2'de görülebileceği gibi en iyi sonuçları elde ettiğimiz birleştirme kombinasyonunda CD, Chem ve GNN ilaç gösterimleriyle alınan tahminler bulunmaktadır. Bu sonuç incelediğimiz üç ilaç gösteriminin de sinerji skor tahminine pozitif yönde etki eden örüntüleri kapsadıklarını gösterir. Bu birleştirme yöntemi ile sinerji skoru tahmini için literatürdeki en iyi yöntemlerden biri olan DeepSynergy'den[7] daha başarılı sonuçlar elde edilmiştir.

Çizelge 5.2: Veri kümesi-model kombinasyonlarının ortalama hata karesine göre çapraz doğrulama sonuçları

	TBYSA	GA	RA	Elas. A.
CDR	266.0 ± 57.9	295.8 ± 61.3	405.1 ± 76.6	451.4 ± 76.6
ChemR	273.7 ± 53.7	295.2 ± 55.9	410.9 ± 63.5	452.0 ± 77.4
GNNR	306.4 ± 55.9	572.5 ± 105.9	578.2 ± 101.8	583.4 ± 103.8
Ağırlık. Ort.	260.112 ± 57.144			

Çizelge 5.3: Veri kümesi-model kombinasyonlarının Pearson korelasyonuna göre çapraz doğrulama sonuçları

	TBYSA	GA	RA	Elas. A.
CDR	0.74 ± 0.04	0.69 ± 0.03	0.56 ± 0.03	0.47 ± 0.03
ChemR	0.72 ± 0.03	0.7 ± 0.03	0.54 ± 0.05	0.47 ± 0.03
GNNR	0.69 ± 0.03	0.15 ± 0.01	0.14 ± 0.01	0.11 ± 0.02
Ağırlık. Ort.	0.745 ± 0.035			

5.3 Karakteristik yönelim ilaç gösteriminin öznitelik analizi

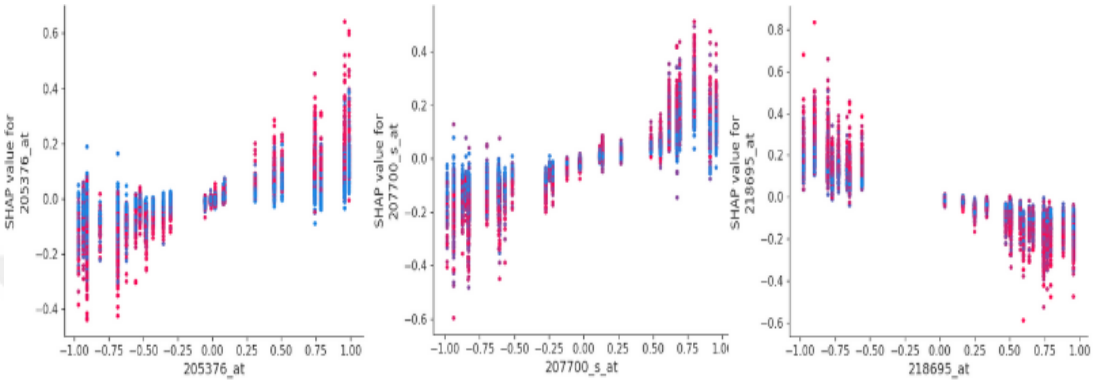
5.2.2 bölümünde anlatıldığı gibi, elimizdeki CD, Chem ve GNN ilaç gösterimleri sinerji skoru tahmininde literatürde bilinen en başarılı yöntemlerden daha başarılı sonuçlar elde edilmesini sağlamışlardır. Bu ilaç gösterimleriyle yapay öğrenme modellerini eğittikten sonra gösterimlerdeki özniteliklerin, modellerin sinerji skoru tahminini nasıl etkilediğini görmek istedik. Aynı zamanda belirlenen özniteliklerin, literatürdeki sinerji skoruna etkisi olduğu bilinen özellikler olup olmadığı araştırıldır.

Yapmak istediğimiz bu analizler için kullanılan tek ilaç gösterimi CD'dir. GNN ve Chem gösterimlerindeki öznitelikler; moleküldeki atomları rastgele gezerek oluşturulması ve özniteliğin gösterdiği değerin molekülün hangi altyapısından (veya özelliğinden) kaynaklı olduğunun tam olarak bulunamaması nedenlerinden dolayı yapılmak istenilen analizler için kullanılmaları uygun görülmedi.

Analizler, SHAP değerleri kullanılarak yapıldı. Bu değerler herbir özniteliğin, modelin tahminine etkisini gösterir. Eğer bu değerler bir öznitelik için pozitifse, bu öznitelik modelin tahmin ettiği değeri arttırmıştır; negatifse azaltmıştır. SHAP değerleri, CDR verisi ile çalıştırılan gradyan arttırma ve tam bağlı yapay sinir ağı modellerine göre çıkarıldı. Bu değerler çıkarılırken, herbir özniteliğin değeri çapraz doğrulamadaki beş gruba göre çıkarılıp ortalaması alındı.

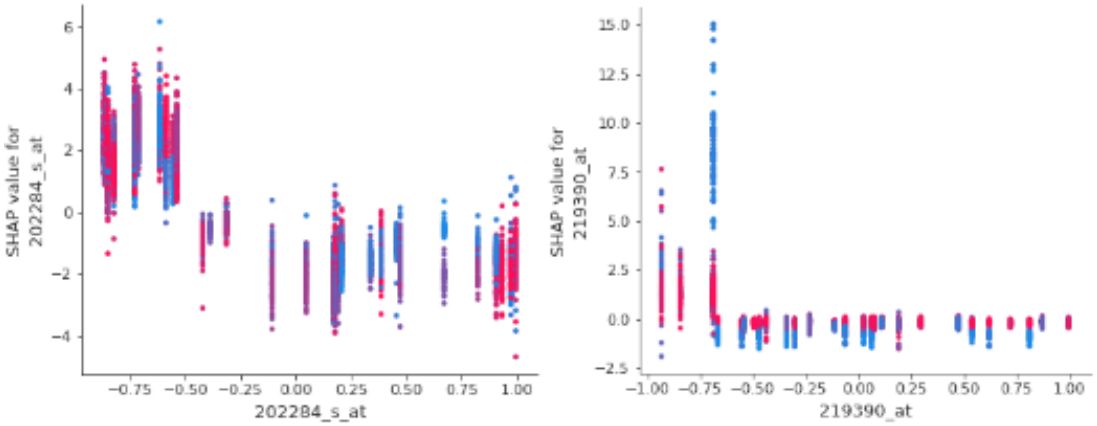
Aşağıda tam bağlı yapay sinir ağının en önemli olarak belirlediği üç genin ve gradyan arttırmanın en önemli olarak belirlediği iki genin analizleri gösterilmiştir. Dokuz yüz yetmiş sekiz gen arasından sadece bu beş genin incelenmesinin sebebi; tam bağlı

yapay sinir ağında analiz edilen bu üç genin SHAP değerleriyle diğer genlerin SHAP değerlerinin arasında büyük bir fark olmasıdır. Aynı zamanda gradyan arttırmanın belirlediği bu iki gen dışındaki genlerin SHAP değerleri 0 gelmiştir (gradyan arttırma modeline göre belirlenen iki gen dışında diğer genlerin bir önemi yoktur.).
Şekil 5.3'te tam bağlı yapay sinir ağının en önemli olarak belirlediği ilk üç genin, Şekil 5.4'te ise gradyan arttırma modelinin en önemli olarak belirlediği ilk iki genin analizi verilmiştir.



Şekil 5.3: CD ilaç gösterimi için TBYSA tarafından belirlenen genlerin analizi

Tam bağlı yapay sinir ağının en önemli olarak belirlediği ilk üç gen; 218695_at(EXOSC4), 207700_s_at(NCOA3), 205376_at(INPP4B)'dir. Şekil 5.3'de görülebileceği gibi INPP4B ve NCOA3 özniteliklerinin değerleri (normalizasyondan geçirilmiş değerleri) artarken SHAP değerleri de yükselmiştir. Dolayısıyla bu özniteliklerin değerleri yükseldikçe modelin daha yüksek sinerji skorları tahmin ettiğini göstermektedir. Literatürde yapılan araştırmalara göre NCOA3, bazı gen ve reseptörlerle etkileşime geçince göğüs kanseri için sinerjik etki yaratan bir gendir[56][57]. INPP4B ise [39]'daki veri kümesinde de örneği bulunan kolon kanserinde normalde sentezlendiği miktardan daha az sentezlenir. Aynı zamanda kanserin yayılmasında etkili olan bazı genlerin kontrolünde görev aldığı belirlenmiştir[58][59]. Bu genlerden farklı olarak EXOSC4 özniteliğinin değeri arttıkça SHAP değerleri düşmeye başlamıştır (Modelin tahmin ettiği sinerji skorlarının değerleri düşmektedir.). Literatür araştırılmasından anlaşıldığı kadarıyla bu genin madde alışverişi için önemli bir role sahip olup, kanser tanımlama için bakılan doku ve vücut sıvılarında yoğun miktarda bulunduğu belirtilmiştir[60]. Literatür araştırmalarımızdan çıkarılan sonuçlara göre, bu genler için TBYSA modeliyle alınan SHAP değerleri bize tutarlı bir analiz sunmaktadır. Bir ilaç, kanserli bir hücrede NCOA3 ve INPP4B genlerinin sentezlenme miktarını artırıyor veya EXOSC4 özniteliğinin sentezlenme miktarını azaltıyorsa; bu ilaç anti-kanser bir özellik gösterir ve kanserli hücre hattı için zehirli olabilir. Dolayısıyla böyle bir ilacın dahil olduğu kombinasyonun sinerjik olma olasılığı yüksektir.



Şekil 5.4: CD ilaç gösterimi için Gradyan Arttırma tarafından belirlenen genlerin analizi

Şekil 5.4, 219390_at(FKBP14) ve 202284_s_at(CDKN1A) özniteliklerinin aldığı SHAP değerlerini göstermektedir. Grafiklerden görülebileceği gibi FKBP14 özniteliğinin normalizasyondan geçirilmiş değeri -0.7 ve daha düşük olduğu zaman, CDKN1A özniteliğinin normalizasyondan geçirilmiş değeri -0.5'ten daha küçük ise tahmin edilen sinerji skorunun değeri artmıştır. Bu genler yumurtalık, göğüs ve mide kanserinde normalde sentezlendiği miktardan daha fazla sentezlenir[61][62][63]. Dolayısıyla eğer bir ilaç, kanserli bir hücrede bu genlerin anlatımı azalttıysa, bu ilacın anti-kanser bir özelliğe sahip olup, katıldığı kombinasyonlarda da sinerjik özellik sergileyebilir. TBYSA modelinde olduğu gibi, gradyan arttırmanın SHAP değerleri de literatür araştırmasıyla tutarlıdır.

5.4 Sinerji skoru optimizasyonu sonuçları

Bu kısımda, tezin ikinci kısmındaki amacımız olan sinerji skoru optimizasyonu için; yapılan deneylerin ayrıntılarına ve sonuçlarına yer verilmiştir.
4.3 kısmında anlatıldığı gibi; [55]'daki ve [45]'de bulunan gen imzaları veri kümesindeki SMILES dizileriyle, JTVAE modeli belirli bir iterasyon boyunca eğitildi. [39]'daki ilaçlar için, eğitilen JTVAE modelinin, kodlayıcı çıktıları ilaç gösterimi olarak kullanılıp, hücre hattı özniteliklerileriyle birleştirilerek bir veri kümesi oluşturuldu. Bu veri kümesi standardizasyon yöntemiyle normalize edilerek, sinerji skoru tahmini için, gradyan artırma modelinin eğitimi ve testi için kullanıldı. Gradyan artırma modelinin çıktısını, bir ilaç-hücre hattı ikilisi için en iyileyecek ikinci ilaç (sinerji skorunu optimize edecek ikinci ilaç) gradyan çıkış yöntemi ile bulundu. Kullandığımız gradyan çıkış yönteminin formülü ve ilaç gösterimini güncellemek için nasıl kullanıldığı 4.3 kısmında anlatılmıştır. Gradyan çıkış işleminden sonra, güncellenen ilaç gösterimlerine standardizasyon işlemi tersten uygulandı. Bu şekilde en son halini alan ilaç vektörleri, JTVAE kod çözücülerine verilerek; gradyan artırma modelinin çıktısını en iyileyen SMILES dizinleri elde

edildi. Aşağıda gradyan artırma, JTVAE ve gradyan çıkışın deneylerimizde kullanılan parametre değerleri listelenmiştir. Gradyan arttırma paramatre optimizasyonu için 4.3'de anlatılan şekilde bölünen eğitim datası için en iyi ortalama hata karesi sonucunu veren parametre kombinasyonları bulundu. JTVAE modelinin parametrelerini belirlemek için bu modeli eğitirken kullanılan [55]'daki ve [45]'den alınan SMILES dizilerinin kod çözücüler tarafından yeniden oluşturulma oranını en iyileyen parametre kombinasyonları arandı. Gradyan çıkış yönteminin parametreleri belirlenirken test verisindeki herbir kombinasyonun en iyilenen sinerji skorunu olabilecek maksimum değere ulaştıran parametre kombinasyonları bulunmaya çalışıldı. Dolayısıyla, parametre optimizasyonu için izlenilen prosedürlerden anlaşılabileceği gibi, gradyan arttırma için 4.3'de belirlenen eğitim verisi, JTVAE için [55], [45]'den alınan SMILES dizileri ve gradyan çıkış için 4.3'de belirlenen test verisi kullanıldı.

JTVAE parametreleri:

* parça boyutu = 4
* gizli katmandaki nöron sayısı = 100
* kodlayıcı çıktı vektörü boyutu = 56
* öğrenme katsayısı = 0.0007

Gradyan arttırma parametreleri:

* ağaç sayısı = 500
* maksimum derinlik = 5
* iterasyon sayısı = 1000
* öğrenme katsayısı = 0.125

Gradyan çıkış parametreleri:

* iterasyon = 750
* $\Delta z = 0.75$
* öğrenme katsayısı = 0.15

Herhangi bir ilaç-hücre hattı ikilisi için elimizdeki gradyan artırma modelinin çıktısını en iyileyen ikinci ilaç yukarıda özetlenen şekilde bulunabiliyoruz. Fakat oluşan moleküllerin gerçekten bu sinerji skorunu sağlayan bir birleşim olup olmadığını doğrulamamız gerekmektedir. Tez çalışmasının ikinci aşamasındaki JTVAE, gözetimli yapay öğrenme

ve gradyan çıkış birleşenleriyle oluşturulan molekülleri doğrulamak için ikinci ilaç moleküllerinin öznitelikleri belirli değerlerle başlatılmıştır. Daha sonra, belirli bir gradyan çıkış iterasyonu sonucu, o iterasyondaki gradyan artırmanın çıktısına yakın sinerji skoru veren ve [39]'daki veri kümesinde bulunan birinci ilaç ve hücre hattının birarada kullanıldığı ikinci ilaçlara yaklaşılıp yaklaşılmadığı kontrol edildi.
Bu doğrulama aşamasında takip edilen herbir adımı aşağıda açıklanmıştır (Şekil 5.5'de bu aşamaların özeti görselleştirilmiştir.):

1. **Sinerji Optimizasyonu Yapılacak Kombinasyonların Belirlenmesi:**

   Bu aşamada gradyan artırma modeli için kullanılan test ve eğitim verisinin nasıl oluşturulduğu 4.3 bölümünde anlatılmıştır. Gradyan artırma modelinin, herbir test ögesi için yaptığı hataların kareleri hesaplandı ve hataların kareleri en az olan ilk beş yüz kombinasyon belirlendi. Sinerji optimizasyonunun diğer aşamalarına, belirlenen bu kombinasyonlarla devam edildi. Bu sayede gradyan çıkış sonucu oluşan sinerji skorlarında, gradyan artırma modelinin hatasından kaynaklı, standart sapma minimuma indirilmeye çalışıldı.

   Göz önünde bulundurulması gereken bir başka değişken de JTVAE modelinin performansıdır. JTVAE, optimize edilmeye çalışılan bir ilacın SMILES dizisi; kodlayıcı ile bir vektöre kodlayıp, kod çözücü ile %100 yeniden oluşturabiliyorsa, sinerji skoru optimizasyonunun diğer aşamalarına dahil edildi. Dolayısıyla, en az hatalı ilk beş yüz kombinasyon; kombinasyonlardaki ikinci ilaçların %100 yeniden oluşturulduğu ögelerden oluşacak şekilde yeniden güncellendi. Bu sayede kod çözücülerin doğru ilacı üretme olasılıkları maksimum yapılmaya çalışıldı.

2. **İkinci İlaç Özniteliklerinin Başlangıçları ve Gradyan Adımlar:**

   Sinerji optimizasyonu yapılacak kombinasyonlar belirlendikten sonra, belirlenen ögelerdeki ikinci ilaçlar 4.3 bölümünde detaylıca anlatılan gradyan çıkış işlemi ile güncellendi. Guncelleme işlemine başlamadan önce bu ilaçların gösterimlerinin başlangıç değerleri belirlenilmelidir. Kullandığımız yöntemin, molekül oluşturma için ne kadar uygun olduğunu anlamak amacıyla ilaçlara JTVAE gösteriminden farklı başlangıç değerleri verildi. Çünkü farklı bir gösterimden başlayıp, gradyan adımlar sonucu; [39]'daki veri kümesinde bulunan ve güncellediğimiz ilacın birarada kullanıldığı birinci ilaç-hücre hattı ikilisi ile birleştirilip, gradyan arttırma modelinin en iyilenen sinerji skoruna yakın bir skor veren herhangi bir SMILES dizisine yaklaşabiliyorsak, kullandığımız yöntem işe yarıyor demektir.

   İlaçlara birden fazla şekilde başlangıç değeri verildi. Bunlardan ilki rastgele bir şeklide oluşturulan vektörlerdir. Bu şekilde oluşturulan başlangıç değerleri

herhangi bir molekül için geçerli gösterimler olmadıkları için üzerinde çalıştığımız herhangi bir kombinasyonda istenilen SMILES dizilerine yaklaşılamadı.

İkinci olarak, asıl moleküle benzer yapıda olan moleküllerin JTVAE gösterimleri ile başlanıldı. Bu başlangıç yönteminde, gene JTVAE tarafından %100 yeniden oluşturulabilen ilaçların gösterimleriyle başlanılmaya dikkat edilmiştir. Birinci deneyimizden farklı olarak, moleküller için geçerli olan başlangıç vektörleridir. Bu vektörlerle yapılan deneylerde, üzerinde çalıştığımız bazı kombinasyonlar için istenilen SMILES dizilerine yakınlaşıldı. (Benzer yapıda olan moleküller PubChem veri tabanından alınmıştır.)

Son olarak, asıl molekülün JTVAE gösterimine normal dağılıma sahip gürültü eklendi. Bu vektörlerle yapılan deneylerde de bazı kombinasyonlar için istenilen SMILES dizilerine yakınlaşıldı. Fakat, ikinci yöntemden farklı olarak eklenilen gürültü sebebiyle bir kombinasyon başına üretilen geçerli SMILES dizisi sayısı azaldı.

Gradyan çıkarma adımındaki maksimum iterasyon sayısı yedi yüz ellidir. Gradyan çıkarma işlemini, maksimum iterasyonu tamamlamak dışında bir sonlandırma koşulu yoktur. Yukarıda anlatılan şekilde başlangıç değerleri verilen molekül gösterimleri ve bu gösterimlere sahip SMILES dizileri, her bir iterasyonda (gradyan adımla güncellemede) kaydedildi. Bunu yapmamızın sebebi, istediğimiz moleküllere hangi iterasyonda yaklaşacağımızı bilmememizdir. Dolayısıyla herbir iterasyonun sonucunu oluşan SMILES dizilerini incelememiz gerekmektedir.

3. **SMILES Dizilerini Doğrulama:**

   Başlangıç değerleri belirlendikten sonra gradyan adımlarla güncellenen vektörler ve bu vektörlerden elde edilen SMILES dizileri, her kombinasyon için her iterasyonda kaydedildi.

   Bütün kombinasyonlar için kaydedilen her SMILES dizinine iki tane analiz uygulandı ve aşağıdaki analizlerin sonuçlarına göre oluşturduğumuz SMILES dizilerinin, istenilen ilaçlara yakınlaşıp yakınlaşmadığına karar verildi.

   (a) İlk analiz ile herbir SMILES dizini için, [39]'daki veri kümesinde, kombinas yondaki birinci ilaç-hücre hattı ikililerileyle kullanılmış hangi ilaca yapısal olarak daha çok benzediği bulunur. Bu benzer ilaçla oluşturulan kombinasyon ile elimizdeki SMILES dizinin oluşturuldu iterasyondaki sinerji skor farkı hesaplanır.

   (b) İkinci analiz ile herbir SMILES dizini için, [39]'daki veri kümesindeki, kombinasyondaki birinci ilaç-hücre hattı ikililerileyle kullanılmış ve sinerji

skoru elimizdeki SMILES dizinin oluşturuldu iterasyondaki sinerji skoruna en yakın olan ilaca, yapısal olarak, ne kadar benzediği hesaplanır.

Bu analizleri yapmak için herbir kombinasyon için oluşturulan SMILES dizilerinin ECFP, Rdkit, MACCS ve terim frekansı(TF) vektörleri çıkarıldı. Çıkarılan bu vektörlere göre, SMILES dizileri kosine, tanimato ve jaccard benzerlik fonksiyonla rıyla karşılaştırıldı.

SMILES dizilerini doğrulamak için, TF ve Jaccard benzerlik fonksiyonunun daha uygun olduğuna karar verilmiştir. Çünkü JTVAE kod çözücüleri SMILES dizilerini, özyineli olarak öbekleri birleştirerek oluşturuyorlar. Bu öbekleri birer kelime ve SMILES dizisini cümle olarak düşünebiliriz. Birleştirilen öbekler içerisinde farklı atom, kenar ve çember grupları bulundurabilir. Dolayısıyla bu öbekler, ECFP, MACCS, Rdkit gibi parmak izleriyle aynı topolojik bilgileri göstermemektedir. ECFP, MACCS, Rdkit gibi vektörler molekül atomlarına ve bu atomların komşuluklarına göre oluşturulup, güncellenen vektörlerdir. JTVAE çıktısı olan SMILES dizileri ise atomları da kapsayan alt-çizgilerin birleşimiyle oluşur.

Bir molekülün TF vektörleri, 1a bölümünde anlatılan sözlüğe göre oluşturuldu. TF vektörlerini karşılaştırırken; bir molekülün barındırdığı öbek sayısına göre veya birden fazla aynı alt-yapıya sahip olmasına göre sonuç olarak verilen değerin değiştiği bir benzerlik fonksiyonu istemediğimizden Jaccard benzerliği kullanıldı.

Gradyan çıkış aşamalarından sonra oluşturulan SMILES dizilerinin doğruluğundan emin olmak için SMILES Dizilerini Doğrulama bölümündeki yöntemler izlenerek [39]'daki veri kümesindeki ilaçlarla benzer olup olmadığı kontrol edildi. Deneylerimizde, SMILES Dizilerini Doğrulama bölümünde bahsedilen benzerlik fonksiyonlarından herhangi biri 0.5'in üstünde bir değer dönüyorsa, ilaçlar benzer olarak kabul edildi. Gürültü ekleyerek ve benzer ilaçların JTVAE gösterimleriyle oluşturduğumuz başlangıç vektörleriyle yaptığımız deneylerdeki bazı kombinasyonlar için; gradyan çıkış işleminde oluşan SMILES dizinlerinin ve bu dizinle gradyan artırmanın tahmin ettiği sinerji skorunun, [39]'daki veri kümesinde bulunan, kombinasyondaki birinci ilaç-hücre hattı ile biraraya getirilen ve gradyan çıkışta oluşturulan SMILES dizisi ile alınan skora yakın skorlar veren ilaçlarla bir benzerliğe sahip oldukları tespit edildi. Tespit edilen kombinasyonların analizini Çizelge 5.5 ve 5.4'te görebilirsiniz. Aynı zamanda gradyan çıkış sonucu oluşturulan ve yakınlaşılan ilaçların iki boyutlu gösterimleri Şekil 5.6'de varilmiştir.
Bu çizelgelerde ilk sütun kombinasyondaki birinci ilaç-hücre hattı ikilisini, ikinci sütun sütun gradyan çıkış sonucu oluşan SMILES dizisini, üçüncü sütun [39]'daki veri kümesinde, birinci ilaç-hücre hattı ikilisiyle biraraya getirilen, gradyan çıkış sonucu

Çizelge 5.4: Gürültü eklenerek istenilen SMILES dizilerine yaklaşılan kombinasyonlar

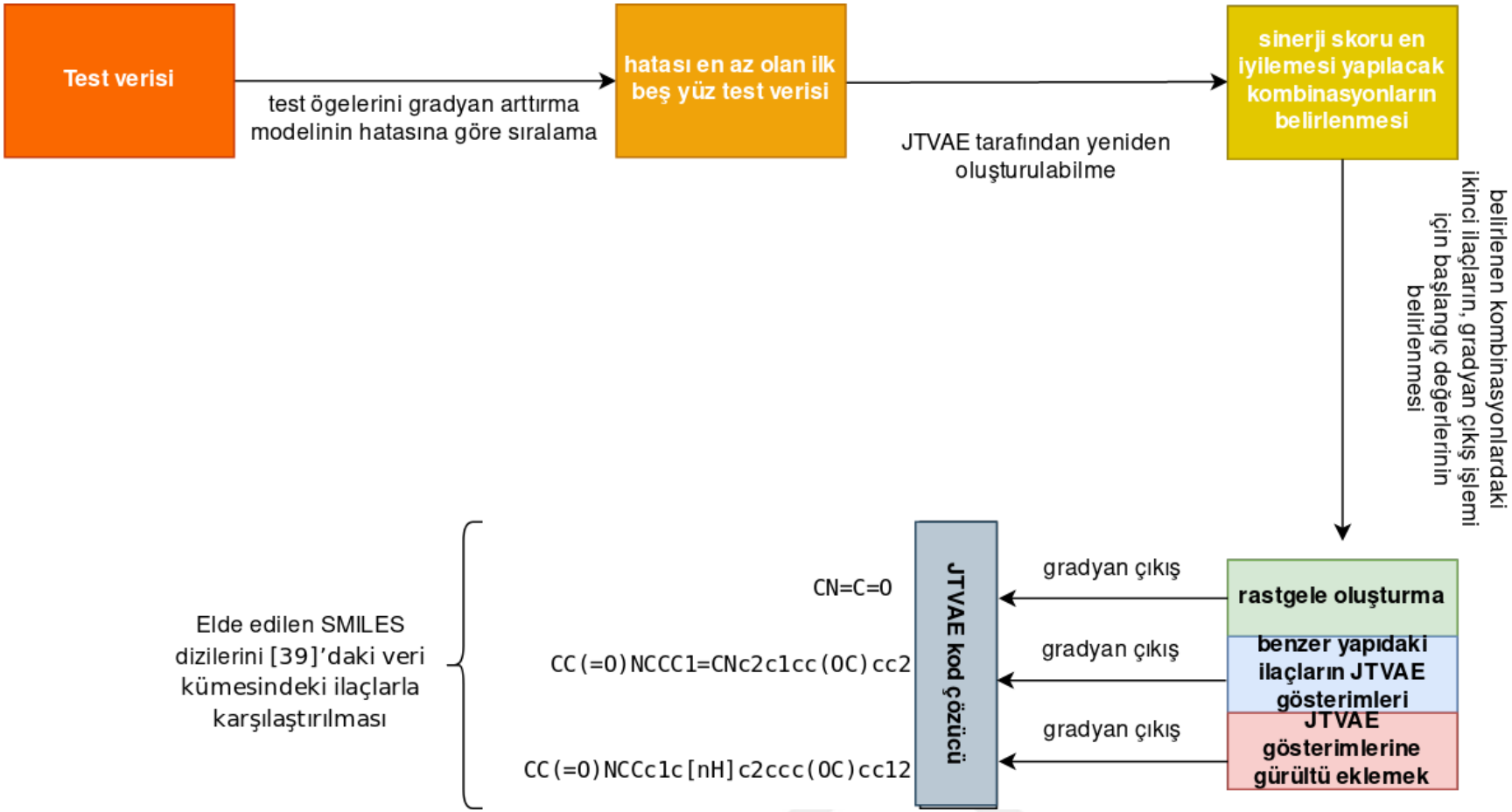
*VINORELBINE_SW*837	*CC*(=*O*)*N*1*CCN*(*CCCOC*(=*O*)*CCCc*2*cccc*(*O*)*c*2)*C*(=*O*)*C*1	*DASATINIB*	0.7	1.846
MK−4541_*KPL*1	*CCCc*1*ccc*(*NC*(=*O*)*OCCCCCCC*(*C*)=*O*)*cc*1	*ABT*−888	0.625	0.168
*L*778123_*CAOV*3	*CCN*=*Cc*1*ccc*(*C*(=*O*)*CCCCCC*(*C*)*C*=*O*)*cc*1	*ABT*−888	0.625	2.789

Çizelge 5.5: Benzer ilaçlarla istenilen SMILES dizilerine yaklaşılan kombinasyonlar

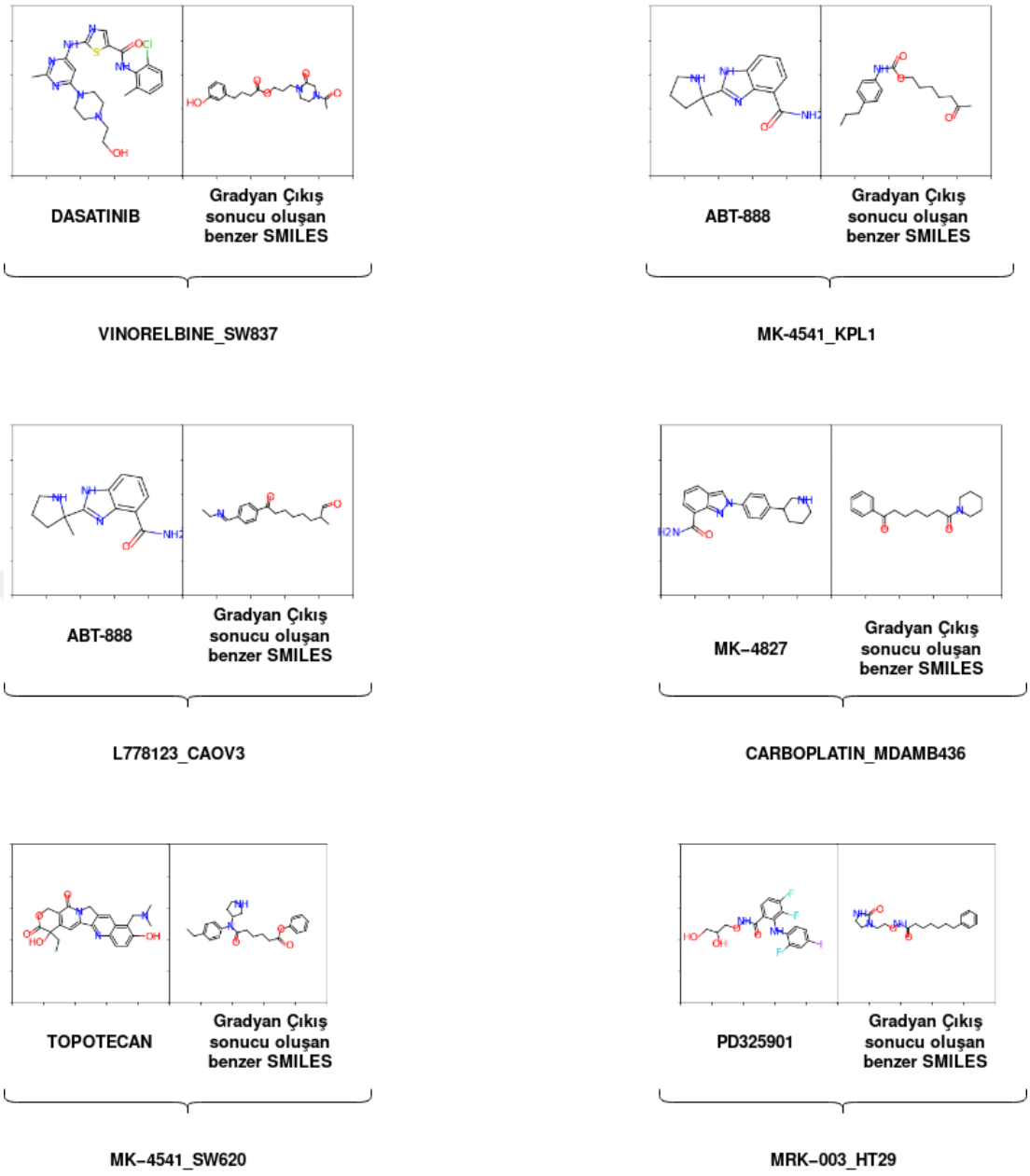
*CARBOPLATIN_MDAMB*436	*O*=*C*(*CCCCCC*(=*O*)*N*1*CCCCC*1)*c*1*ccccc*1	*MK*−4827	0.75	1.22
MK−4541_*SW*620	*CCc*1*ccc*(*N*(*C*(=*O*)*CCCCC*(=*O*)*Oc*2*ccccc*2)*C*2*CCNC*2)*cc*1	*TOPOTECAN*	0.727	0.285
MRK−003_*HT*29	*O*=*C*(*CCCCCCc*1*ccccc*1)*NOCCN*1*CCNC*1=*O*	*PD*325901	0.7	0.738

oluşan SMILES dizisine hem benzer olup hem de yakın skoru veren ilacı, dördüncü sütun [39]'daki veri kümesindeki ilaç ile gradyan çıkış sonucu oluşan SMILES dizisinin Jaccard benzerliğini,beşinci sütun ise [39]'daki veri kümesindeki ilaç ve gradyan çıkış sonucu oluşan SMILES dizisi ile oluşturulan iki kombinasyon arasındaki sinerji skor farkını gösterir.

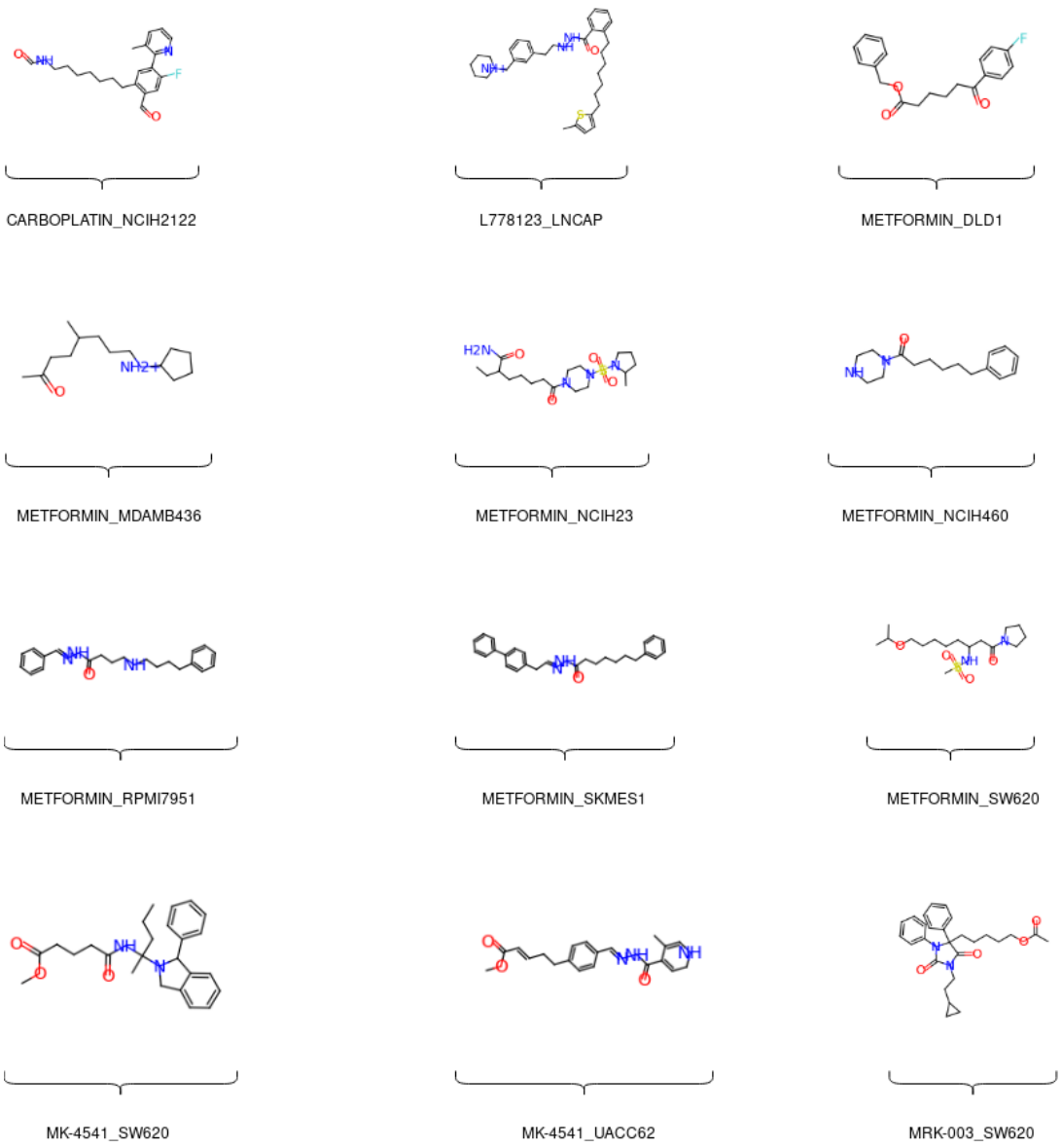
Analiz sonuçlarına göre, sinerji skoru optimizasyonu için kullandığımız yöntem, bir ilaç-hücre hattı ikilisi için istenilen sinerji skorunu vermesini sağlayacak ve kombinasyona ikinci ilaç olarak eklenecek SMILES dizilerini bulabiliyor. Dolayısıyla bu yöntem, gene bir ilaç-hücre hattı ikilisi için, literatürde bilinen en yüksek sinerji skorundan daha yüksek bir sinerji skoru elde etmesini sağlayacak ikinci ilacı üretmek için kullanılabilir. Deneylerimizde, [39]'daki veri kümesindeki bazı ilaç-hücre hattı ikilileri için, gradyan çıkış işlemiyle, [39]'daki veri kümesindeki en yüksek sinerji skorundan daha yüksek sinerji skorları veren SMILES dizileri elde edilmiştir. Şekil 5.7'de bu ilaç-hücre hattı ikilileri ile bu ikililer için [39]'daki veri kümesinden daha yüksek sinerji skorları üreteceğini tahmin ettiğimiz SMILES dizileri verilmiştir.



Şekil 5.5: Sinerji skoru en iyileme izlenilen yöntem özeti



Şekil 5.6: Gradyan çıkış sonucunda oluşan ve yakınlaşılan SMILES dizileri



Şekil 5.7: İlaç-hücre hattı ikilileri için kullanılan veri kümesinden daha sinerjik skorlar veren ve gradyan çıkarma işlemiyle oluşturulan SMILES dizileri

6. DEĞERLENDİRME VE GELECEK ÇALIŞMALAR

Kısacası bu tez çalışmasında, ilk olarak CD, Chem, GNN ilaç gösterimleri ile farklı yapay öğrenme modellerinden alınan tahminler birleştirilerek sinerji skoru tahmini için literatürdeki tüm çalışmalardan daha başarılı sonuçlar elde edildi. Sinerji skoru tahmini için aldığımız sonuçları daha da geliştirmek için gelecek çalışmalarda;

* ilaç-ilaç etkileşim, ilaç-protein etkileşim, protein-protein etkileşim ağlarının farklı çizge yapay sinir ağı mimarileri kullanarak sinerji skoru tahminine etkilerini incelemek istiyoruz.

* Gerçek hayatta kullanılan ilaç kombinasyonlarının sinerji skorunu etkileyen birçok farklı değişken bulunmaktadır (birarada kullanılan ilaçların dozu, ilaç kombinasyonunun uygulandığı kişi vs.). Gelecek çalışmalarda kullanacağımız yapay öğrenme çalışmalarını, bu parametreleri göz önünde bulunduracak şekilde düzenlemeyi düşünüyoruz. Bu sayede yapay öğrenme çalışmalarını, bir kişi için en optimum politerapi(kombinasyonel terapi) bulma gibi problemleri çözmek için kullanabiliriz.

İkinci olarak, ilaç-kanserli hücre hattı ikilileri için gradyan arttırma modelinin tahminini en iyileyecek ikinci ilaç SMILES dizileri üretildi.Bu SMILES dizilerini üretmek için JTVAE modeli ve gradyan çıkış yöntemleri kullanıldı. Sinerji skoru optimizasyonu için üretilen moleküllerin gerçekten ulaştığımız sinerji skorlarını sağlayıp sağlayamayaca ğı laboratuvar deneyi yapılmadan kesin olarak bilinemez. Oto-kodlayıcı tarafından üretilen SMILES dizinleri tamamen yeni olup, herhangi bir veri tabanında bulunmadıkları için literatür araştırması da yapılamaz. Bu sebeplerden dolayı oluşturulan SMILES dizileri, kullandığımız veri kümesindeki ilaçlarla Jaccard benzerliğine göre karşılaştırıldı. Bu karşılaştırma sonucu, sinerji skorunu en iyilemeye çalıştığımız altı kombinasyon için elde edilen sinerji skoruna yakın skorlar veren SMILES dizilerine yakınlaşılabildiği tespit edildi.
Belirlenen kombinasyonlar için oluşturulan ve yakınlaşılan SMILES dizileri Şekil 5.6'te verilmiştir. Gözlemlenebildiği gibi, gradyan çıkış sonucu oluşturulan SMILES dizileri, yakınlaşılan SMILES dizileri ile ortak alt yapıları içermektedir. Buna ek olarak,

gradyan çıkış tarafından oluşturulan SMILES dizileri, orijinal SMILES dizilerinden farklı bir alt yapı içermedikleri fark edildi. Analizlerimizde, JTVAE modelinde, nodsal ağaç oluştururken, molekül çizgesi, çember(ring), atom, kenar gruplarından oluşan öbeklere ayrıldığı için Jaccard benzerliği kullanıldı. Bu sebepten, gradyan çıkış sonucu oluşturulan SMILES dizilerinde bulunan ortak alt yapıların sayısı, orijinal SMILES dizinen farklı olabiliyor.

Sinerji skorunu en iyileyen molekülü oluşturmak için yaptığımız çalışmaları ileride; bir öznitelik analizi yaparak sinerji skoruna en çok etki eden topolojik özelliği bulup, oto-kodlayıcının oluşturduğu gizli vektörü, bu topolojik özelliğe göre en iyileyecek molekülleri oluşturmayı planlıyoruz. Fakat elde edilen moleküller, literatürde bulunamaz sa, istenilen sinerji skorunu sağlayıp sağlamayacağını sadece laboratuvar deneyi yaparak kesinleştirebiliriz.

KAYNAKLAR

[1] **Wang**, **H.** et al. (2016). Global, regional, and national life expectancy, all-cause mortality, and cause-specific mortality for 249 causes of death, 1980–2015: a systematic analysis for the Global Burden of Disease Study 2015. In: *The lancet* 388.10053, pp. 1459–1544.

[2] **Csermely**, **P.** et al. (2013). Structure and dynamics of molecular networks: a novel paradigm of drug discovery: a comprehensive review. In: *Pharmacology & therapeutics* 138.3, pp. 333–408.

[3] **Griner**, **L. A. M.** et al. (2014). High-throughput combinatorial screening identifies drugs that cooperate with ibrutinib to kill activated B-cell–like diffuse large B-cell lymphoma cells. In: *Proceedings of the National Academy of Sciences* 111.6, pp. 2349–2354.

[4] **Goldoni**, **M.** and **Johansson**, **C.** (2007). A mathematical approach to study combined effects of toxicants in vitro: evaluation of the Bliss independence criterion and the Loewe additivity model. In: *Toxicology in vitro* 21.5, pp. 759–769.

[5] **Bliss**, **C.** (1939). The toxicity of poisons applied jointly 1. In: *Annals of applied biology* 26.3, pp. 585–615.

[6] **Yadav**, **B.** et al. (2015). Searching for drug synergy in complex dose–response landscapes using an interaction potency model. In: *Computational and structural biotechnology journal* 13, pp. 504–513.

[7] **Preuer**, **K.** et al. (2018). DeepSynergy: predicting anti-cancer drug synergy with Deep Learning. In: *Bioinformatics* 34.9, pp. 1538–1546.

[8] **Janizek**, **J. D.**, **Celik**, **S.**, and **Lee**, **S.-I.** (2018). Explainable machine learning prediction of synergistic drug combinations for precision cancer medicine. In: *bioRxiv*, p. 331769.

[9] **Lundberg**, **S. M.** and **Lee**, **S.-I.** (2017a). A Unified Approach to Interpreting Model Predictions. In: *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., pp. 4765–4774.

[10] **Liljefors**, **T.**, **Krogsgaard-Larsen**, **P.**, and **Madsen**, **U.** (2002). Textbook of drug design and discovery. CRC Press.

[11] **Güner**, **O. F.** (2000). Pharmacophore perception, development, and use in drug design. Vol. 2. Internat'l University Line.

[12] **Mauser**, **H.** and **Guba**, **W.** (2008). Recent developments in de novo design and scaffold hopping. In: *Current opinion in drug discovery & development* 11.3, pp. 365–374.

[13] **Elton**, **D. C.** et al. (2019). Deep learning for molecular design—a review of the state of the art. In: *Molecular Systems Design & Engineering* 4.4, pp. 828–849.

[14] **SALLOUM**, **Z.** (2019). Back Propagation, the Easy Way (Part 1).

[15] **Goodfellow**, **I.**, **Bengio**, **Y.**, and **Courville**, **A.** (2016). Deep learning. MIT press.

[16] **Breiman**, **L.** (1997). *Arcing the edge*. Tech. rep. Technical Report 486, Statistics Department, University of California at ...

[17] **Grover**, **P.** (2017). Gradient Boosting from scratch.

[18] **Zou**, **H.** and **Hastie**, **T.** (2005). Regularization and variable selection via the elastic net. In: *Journal of the royal statistical society: series B (statistical methodology)* 67.2, pp. 301–320.

[19] **Ho**, **T. K.** (1995). Random decision forests. In: *Proceedings of 3rd international conference on document analysis and recognition*. Vol. 1. IEEE, pp. 278–282.

[20] **Hatipoğlu**, **E.** (2018). Machine Learning — Prediction Algorithms — Decision Tree —Random Forest — Part 5.

[21] **Géron**, **A.** (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media.

[22] **Lundberg**, **S. M.**, **Erion**, **G. G.**, and **Lee**, **S.-I.** (2018). Consistent individualized feature attribution for tree ensembles. In: *arXiv preprint arXiv:1802.03888*.

[23] **Lundberg**, **S. M.** and **Lee**, **S.-I.** (2017b). A unified approach to interpreting model predictions. In: *Advances in neural information processing systems*, pp. 4765–4774.

[24] **Tsigelny**, **I. F.** (2019). Artificial intelligence in drug combination therapy. In: *Briefings in bioinformatics* 20.4, pp. 1434–1448.

[25] **Chen**, **X.** et al. (2016). NLLSS: predicting synergistic drug combinations based on semi-supervised learning. In: *PLoS computational biology* 12.7.

[26] **Li**, **X.** et al. (2017). Prediction of synergistic anti-cancer drug combinations based on drug target network and drug induced gene expression profiles. In: *Artificial intelligence in medicine* 83, pp. 35–43.

[27] **Bansal**, **M.** et al. (2014). A community computational challenge to predict the activity of pairs of compounds. In: *Nature biotechnology* 32.12, p. 1213.

[28] **Lamb**, **J.** et al. (2006). The Connectivity Map: using gene-expression signatures to connect small molecules, genes, and disease. In: *science* 313.5795, pp. 1929–1935.

[29] **Zitnik**, **M.**, **Agrawal**, **M.**, and **Leskovec**, **J.** (2018). Modeling polypharmacy side effects with graph convolutional networks. In: *Bioinformatics* 34.13, pp. i457–i466.

[30] **Singh**, **H.**, **Rana**, **P. S.**, and **Singh**, **U.** (2018). Prediction of drug synergy in cancer using ensemble-based machine learning techniques. In: *Modern Physics Letters B* 32.11, p. 1850132.

[31] **Sun**, **Y.** et al. (2015). Combining genomic and network characteristics for extended capability in predicting synergistic drugs for cancer. In: *Nature communications* 6.1, pp. 1–10.

[32] **KalantarMotamedi**, **Y.** et al. (2018). A systematic and prospectively validated approach for identifying synergistic drug combinations against malaria. In: *Malaria journal* 17.1, p. 160.

[33] **Duan**, **Q.** et al. (2014). LINCS Canvas Browser: interactive web app to query, browse and interrogate LINCS L1000 gene expression signatures. In: *Nucleic acids research* 42.W1, W449–W460.

[34] **Mott**, **B. T.** et al. (2015). High-throughput matrix screening identifies synergistic and antagonistic antimalarial drug combinations. In: *Scientific reports* 5, p. 13891.

[35] **Koutsoukas**, **A.** et al. (2011). From in silico target prediction to multi-target drug design: current databases, methods and applications. In: *Journal of proteomics* 74.12, pp. 2554–2574.

[36] **Gaulton**, **A.** et al. (2012). ChEMBL: a large-scale bioactivity database for drug discovery. In: *Nucleic acids research* 40.D1, pp. D1100–D1107.

[37] **Geer**, **L. Y.** et al. (2010). The NCBI biosystems database. In: *Nucleic acids research* 38.suppl_1, pp. D492–D496.

[38] **Jeon**, **M.** et al. (2018). In silico drug combination discovery for personalized cancer therapy. In: *BMC systems biology* 12.2, p. 16.

[39] **O'Neil**, **J.** et al. (2016). An unbiased oncology compound screen to identify novel combination strategies. In: *Molecular cancer therapeutics* 15.6, pp. 1155–1162.

[40] **Clark**, **N. R.** et al. (2014). The characteristic direction: a geometrical approach to identify differentially expressed genes. In: *BMC bioinformatics* 15.1, p. 79.

[41] **Cao**, **D.-S.** et al. (2013). ChemoPy: freely available python package for computational biology and chemoinformatics. In: *Bioinformatics* 29.8, p. 1092.

[42] **Hinselmann**, **G.** et al. (2011). jCompoundMapper: An open source Java library and command-line tool for chemical fingerprints. In: *Journal of cheminformatics* 3.1, pp. 1–14.

[43] **Tsubaki**, **M.**, **Tomii**, **K.**, and **Sese**, **J.** (2019). Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. In: *Bioinformatics* 35.2, pp. 309–318.

[44] **Maust**, **J.**, **Leopold**, **J.**, and **Bugrim**, **A.** (2019). Network Entropy Reveals that Cancer Resistance to MEK Inhibitors Is Driven by the Resilience of Proliferative Signaling. In: *International Conference on Complex Networks and Their Applications*. Springer, pp. 751–761.

[45] **Wang**, **Z.**, **Clark**, **N. R.**, and **Ma'ayan**, **A.** (2016). Drug-induced adverse events prediction with the LINCS L1000 data. In: *Bioinformatics* 32.15, pp. 2338–2345.

[46] **Rogers**, **D.** and **Hahn**, **M.** (2010). Extended-connectivity fingerprints. In: *Journal of chemical information and modeling* 50.5, pp. 742–754.

[47] **Sushko**, **I.** et al. (2011). Online chemical modeling environment (OCHEM): web platform for data storage, model development and publishing of chemical information. In: *Journal of computer-aided molecular design* 25.6, pp. 533–554.

[48] **Mueller**, **J.**, **Gifford**, **D.**, and **Jaakkola**, **T.** (2017). Sequence to better sequence: continuous revision of combinatorial structures. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, pp. 2536–2544.

[49] **Gómez-Bombarelli**, **R.** et al. (2018). Automatic chemical design using a data-driven continuous representation of molecules. In: *ACS central science* 4.2, pp. 268–276.

[50] **Sterling**, **T.** and **Irwin**, **J. J.** (2015). ZINC 15–ligand discovery for everyone. In: *Journal of chemical information and modeling* 55.11, pp. 2324–2337.

[51] **Rasmussen**, **C. E.** and **Williams**, **C. K.** (2006). Gaussian Processes for Machine Learning the MIT Press. In: *Cambridge, MA*.

[52] **Kang**, **S.** and **Cho**, **K.** (2018). Conditional molecular design with deep generative models. In: *Journal of chemical information and modeling* 59.1, pp. 43–52.

[53] **Lim**, **J.** et al. (2018). Molecular generative model based on conditional variational autoencoder for de novo molecular design. In: *Journal of cheminfor matics* 10.1, pp. 1–9.

[54] **Jin**, **W.**, **Barzilay**, **R.**, and **Jaakkola**, **T.** (2018). Junction tree variational autoencoder for molecular graph generation. In: *arXiv preprint arXiv:1802.04364*.

[55] **Polykovskiy**, **D.** et al. (2018). Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. In: *arXiv preprint arXiv:1811.12 823*.

[56] **Wang**, **S.** et al. (2018). LRH1 enhances cell resistance to chemotherapy by transcriptionally activating MDC1 expression and attenuating DNA damage in human breast cancer. In: *Oncogene* 37.24, 3243 –3259.

[57] **Li**, **Z.** et al. (2018). The PI3K and AIB1 interaction is involved in estrogen treated breast cancer cells. In: *Cellular and molecular biology (Noisy-le-Grand, France)* 64.6, pp. 65–70.

[58] **Sung**, **J.-Y.**, **Na**, **K.**, and **Kim**, **H.-S.** (2017). Down-regulation of inositol polyphosp hate 4-phosphatase type II expression in colorectal carcinoma. In: *Anticancer research* 37.10, pp. 5525–5531.

[59] **Agoulnik**, **I. U.** et al. (2011). INPP4B: the new kid on the PI3K block. In: *Oncotarget* 2.4, p. 321.

[60] **Ni**, **J.** et al. (2019). Exosomes in Cancer Radioresistance. In: *Frontiers in oncology* 9, p. 869.

[61] **Sun**, **L.-Y.** et al. (2017). Inhibitory effects of FKBP14 on human cervical cancer cells. In: *Molecular medicine reports* 16.4, pp. 4265–4272.

[62] **Wei**, **C.-Y.** et al. (2015). Expression of CDKN1A/p21 and TGFBR2 in breast cancer and their prognostic significance. In: *International journal of clinical and experimental pathology* 8.11, p. 14619.

[63] **Abbas**, **T.** and **Dutta**, **A.** (2009). p21 in cancer: intricate networks and multiple activities. In: *Nature Reviews Cancer* 9.6, pp. 400–414.

ÖZGEÇMİŞ

Ad-Soyad : Işıksu Ekşioğlu

Uyruğu : T.C.

Doğum Tarihi ve Yeri : 07.07.1994 Ankara

E-posta : sksueksioglu@gmail.com

ÖĞRENİM DURUMU:

- **Yüksek Lisans** : 2020, TOBB ETÜ, Bilgisayar Müh.
- **Lisans** : 2017, TOBB ETÜ, Bilgisayar Müh.

MESLEKİ DENEYİM VE ÖDÜLLER:

Yıl	**Yer**	**Görev**
2017 - Halen	TOBB ETÜ	Yüksek Lisans Öğrencisi

YABANCI DİL: İngilizce

TEZDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER:

- **Eksioglu, I.,**, Tan,M., Prediction of Drug Synergy by Ensemble Learning, *2019 The International Conference on Computational Intelligence methods for Bioinformatics and Biostatistics (CIBB).*

DİĞER YAYINLAR, SUNUMLAR VE PATENTLER:

- Tan, M., Ozgul, O. F., Bardak, B., **Eksioglu, I.,** Sabuncuoglu, S., Drug response prediction by ensemble learning and drug-induced gene expression signatures, in *Genomics*, 2019, 111.5: 1078-1088.