T.C.
TRAKYA ÜNİVERSİTESİ
SAĞLIK BİLİMLERİ ENSTİTÜSÜ
BİYOİSTATİSTİK VE TIBBİ BİLİŞİM
ANABİLİM DALI
YÜKSEK LİSANS PROGRAMI

Tez Yöneticisi
Dr. Öğr. Üyesi Selçuk KORKMAZ
İkinci Tez Yöneticisi
Prof. Dr. Necdet SÜT

DERİN ÖĞRENME İLE İLAÇ MOLEKÜLLERİNİN AKTİVİTELERİNİN SINIFLANDIRILMASI

(Yüksek Lisans Tezi)

Hatice KANBERİZ

EDİRNE - 2020

T.C.
TRAKYA ÜNİVERSİTESİ
SAĞLIK BİLİMLERİ ENSTİTÜSÜ
BİYOİSTATİSTİK VE TIBBİ BİLİŞİM
ANABİLİM DALI
YÜKSEK LİSANS PROGRAMI

Tez Yöneticisi
Dr. Öğr. Üyesi Selçuk KORKMAZ
İkinci Tez Yöneticisi
Prof. Dr. Necdet SÜT

DERİN ÖĞRENME İLE İLAÇ MOLEKÜLERİNİN AKTİVİTELERİNİN SINIFLANDIRILMASI

(Yüksek Lisans Tezi)

Hatice KANBERİZ

Destekleyen Kurum :

Tez No :

EDİRNE - 2020

T. C.

TRAKYA ÜNİVERSİTESİ

Sağlık Bilimleri Enstitüsü Müdürlüğü

ONAY

Trakya Üniversitesi Sağlık Bilimleri Enstitüsü Biyoistatistik ve Tıbbi Bilişim Anabilim Dalı yüksek lisans programı çerçevesinde ve Dr. Öğr. Üyesi Selçuk KORKMAZ danışmanlığında yüksek lisans öğrencisi Hatice KANBERİZ tarafından tez başlığı " Derin Öğrenme ile İlaç Moleküllerinin Aktivitelerinin Sınıflandırılması " olarak teslim edilen bu tezin tez savunma sınavı 23/01/2020 tarihinde yapılarak aşağıdaki jüri üyeleri tarafından "**Yüksek Lisans Tezi**" olarak kabul edilmiştir.

İmza

Prof. Dr. Necdet SÜT

JÜRİ BAŞKANI

İmza

Doç. Dr. Hayriye Ertem VEHİD

JÜRİ ÜYESİ

İmza

Dr. Öğr. Üyesi Selçuk KORKMAZ

JÜRİ ÜYESİ (DANIŞMAN)

Yukarıdaki imzaların adı geçen öğretim üyelerine ait olduğunu onaylarım.

Prof. Dr. Tammam SİPAHİ

Enstitü Müdürü

TEŞEKKÜR

Lisansüstü eğitim sürecinde tez konusunu belirlemede ve bu yolda ilerlememde bana destek olan Sayın Dr. Öğr. Üyesi Selçuk KORKMAZ'a, Sayın Prof. Dr. Necdet SÜT'e, Sayın Doç. Dr. Fatma Nesrin TURAN'a ve çok değerli rahmetli anneanneme, bugünlere gelmemde emeği olan kişilere, her zaman yanımda olan aileme en içten duygularımla teşekkür ederim.

İÇİNDEKİLER

SİMGE VE KISALTMALAR

CART : Classification and Regression Tree

DSA : Derin Sinir Ağları

DVM : Destek Vektör Makineleri

HTS : High Throughput Screening

KKT : Karush Kuhn Tucker

MCC : Matthew's Correlation Coefficient

OOB : Out Of Bag

QP : Quadratic Programming

QSAR : Quantitative Structure-Activity Relationship

RELU : Rectified Linear Units

RF : Random Forest

SGD : Stochastic Gradient Descent

ST : Sanal Tarama

VC **:** Vapnik Chervonenkis

YSA **:** Yapay Sinir Ağları

GİRİŞ VE AMAÇ

İlaç geliştirme, önceden tanımlanmış yapı-aktivite ilişkilerinden yararlanarak farmakolojik aktivitesi tahmin edilen potansiyel ilaç moleküllerini tasarlama işlemidir (1). Var olandan daha etkin, daha az toksik ve yan etkileri en aza indirilmiş yararlı bileşikler oluşturmak yeni ilaç geliştirme sürecinin amaçları arasındadır (2). Yeni bir ilaç geliştirme süreci hem çok maliyetli hem de çok zaman alıcıdır (3). Yeni ilaç geliştirme çalışmaları ortalama 15 yıl sürmekte ve süreç için bir milyar doların üzerinde para harcanmaktadır (4). Yeni ilaç geliştirme çalışmalarına harcanan süre ve para her ne kadar çok olsa da akılcı ilaç tasarımı ile birlikte ilaç geliştirme sürecine harcanan sürenin ve maliyetin azaldığı görülmektedir. Akılcı ilaç tasarımı, biyolojik hedefe ilişkin verilerden yola çıkarak, yeni ilaç moleküllerinin tasarlandığı ilaç geliştirme tekniği olarak adlandırılmaktadır. Akılcı ilaç tasarımı ile birlikte ilaç geliştirme çalışmalarının erken evresinde binlerce molekül hızlı bir şekilde taranmakta ve aktivite gösteren moleküller ile yola devam edilmektedir. Bu amaçla en sık kullanılan deneysel yöntem yüksek verimli tarama (high throughput screening-HTS) yöntemidir. Bu yöntemde binlerce molekülün belirli bir reseptöre veya enzime karşı aktivite gösterip göstermedikleri hızlı bir şekilde taranabilir.

Günümüzde, HTS yöntemi ile elde edilen veriler PubChem veri tabanına yüklenmektedir. HTS yöntemi ile elde edilen veriler genellikle dengesiz veri yapısında olmaktadır. Başka bir deyişle, inaktif molekül sayısı aktif molekül sayısından oldukça fazladır.

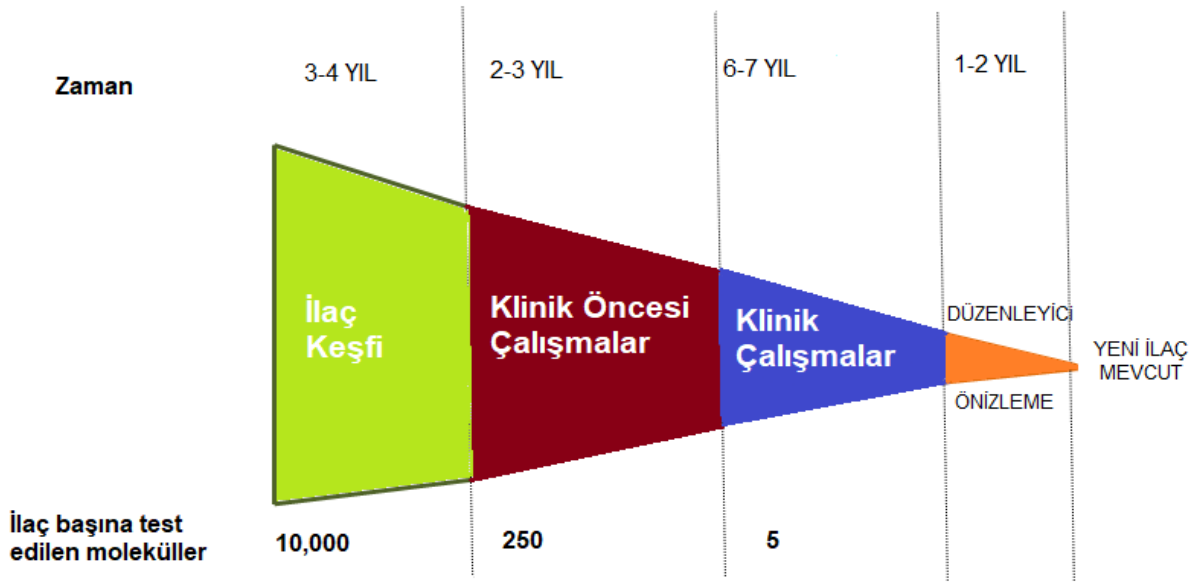
İlaç moleküllerinin hızlı bir şekilde taranmasında kullanılacak diğer bir yöntem ise sanal tarama (virtual screening) yöntemidir. Sanal taramada, makine öğrenimi yöntemleri kullanılarak ilaç molekülleri hızlı bir şekilde sınıflandırılabilmekte veya aktivite sıralamasına sokulabilmektedir. Bu amaçla, destek vektör makineleri (DVM) ve random forest (RF) gibi makine öğrenimi yöntemleri aktif moleküllerin tespiti için literatürde sıklıkla kullanılmaktadır. Ancak bu algoritmalar dengesiz veri yapılarında iyi performanslar gösterememektedir. Bu nedenle, literatürde bu algoritmaların kullanıldığı veri setleri çoğunlukla dengeli veri yapılarından oluşturulmuştur. PubChem veri tabanındaki veri boyutunun artmasıyla birlikte sanal tarama için yeni yöntemlerin kullanılması gerekliliği ortaya çıkmıştır. Son yıllarda, derin sinir ağları (DSA) birçok alanda oldukça iyi performanslar ortaya koymuş ve DVM ve RF gibi makine öğrenimi yöntemlerinin performanslarını geçmiştir. Özellikle yüksek boyutlu verilerde çok iyi performanslar gösteren DSA algoritması son yıllarda ilaç geliştirme çalışmalarında sanal tarama amacıyla da kullanılmaya başlanmıştır.

Bu çalışmada, PubChem veri tabanından elde edilen ve farklı derecede dengesizlik yapısına sahip olan 5 adet veri seti kullanılmıştır. Elde edilen veri setleri DSA algoritması ile eğitilmiştir. DSA algoritmasının performansı literatürde sanal tarama için sıklıkla kullanılan DVM ve RF algoritmalarının performansı ile karşılaştırılmıştır.

GENEL BİLGİLER

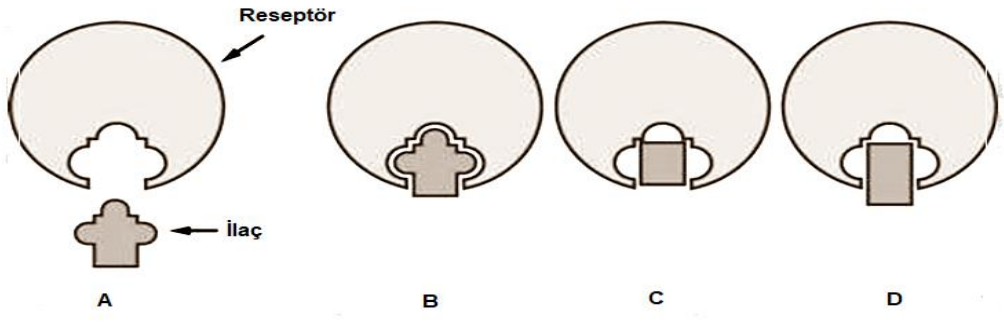
İLAÇ GELİŞTİRME ÇALIŞMALARI

İlaç geliştirme, yeni aday molekül bulma çalışmaları oldukça zorlu, zaman alıcı ve maliyetlidir. Bu süreç 12-15 yıl arası sürebilir ve maliyeti bir milyar doları aşabilir (3). On binlerce küçük molekülle başlayan süreç sadece bir molekülün ilaç olarak piyasaya sürülmesi ile son bulur (Şekil 1).



Şekil 1. Yeni bir ilaç geliştirme aşamaları

Akılcı ilaç tasarımının kullanılması ile birlikte hem geliştirilen ilaç sayısında anlamlı artış meydana gelmiştir hem de ilaç geliştirme için harcanan zaman kısalmıştır. Bu yaklaşımda ilk amaç hastalığa neden olduğu düşünülen reseptörün (proteinin) 3 boyutlu yapısını bulmak ve aktif yöresini ortaya çıkarmaktır. Böylece geliştirilecek/sentezlenecek olan molekül (ilaç) reseptörün bu aktif yöresine bağlanma ve aktivite gösterme şansı en yüksek molekül olacaktır. Bu durum literatürde anahtar-kilit modeli olarak adlandırılmaktadır (Şekil 2).



Şekil 2. Anahtar-kilit modeli

İlaç geliştirme çalışmalarındaki temel zorluk, hastalığa neden olan hedef reseptöre bağlanıp aktivite gösterebilecek yeni moleküllerin bulunmasıdır. İlaç geliştirme çalışmalarına başlamadan önce çok sayıda (binlerce) küçük kimyasal molekül arasından seçim yapmak gerekmektedir. Bu amaçla, deneysel bir yaklaşım olan yüksek verimli tarama (high throughput screening, HTS) tekniği kullanılarak çok sayıdaki küçük molekül arasından belirli bir reseptöre veya enzime karşı aktivite gösterenler tespit edilmektedir (5). Binlerce küçük molekül arasından HTS yönteminin kullanılmasıyla seçilen ve aktivite göstereceği düşünülen bu moleküllere öncü bileşikler (lead compounds) adı verilir. Daha sonra bu öncü bileşikler optimize edilir ve ön-klinik denemelere (hayvan deneyleri) geçilir. Ön-klinik denemeler tamamlandıktan sonra klinik denemeler aşamasına geçilir. Bu denemeler üç aşamadan oluşmaktadır: Faz I-II-III.

Faz I denemeleri insanlar üzerinde yapılan testlerin ilk aşamasıdır. Bu fazdaki amaç, ürün ile ilgili güvenirlik verilerinin elde edilmesi, doz aralığının belirlenmesi,

tolerans ve farmakokinetik özelliklerin değerlendirilmesidir. Faz I denemeleri için, 20-100 kişilik sağlıklı bir gönüllü grubu seçilir ve ortalama 1-1.5 yılda tamamlanır.

Faz II çalışmalarında ürünün klinik etkinliği ve yan etkileri daha büyük bir hasta kitlesinde değerlendirilir. Bu amaçla, ilgili hastalığa sahip 1000-5000 gönüllü hasta üzerinde bu çalışmalar yürütülür. Faz II çalışmaları genellikle çok merkezli, çok uluslu, rastgele kontrollü çift kör denemeler olarak yürütülür. Bu aşama ortalama 3-4 yıl sürer.

Klinik denemelerin Faz III aşaması da başarıyla yürütüldükten sonra ürünün ilaç olarak kullanılabilmesi için ilgili düzenleme kuruluşundan onay alınması gerekmektedir. Bu amaçla Amerika Birleşik Devletleri'nde FDA'ya (Food and Drug Administration) Yeni İlaç Başvurusu (New Drug Application, NDA) yapılması gerekir. Avrupa Birliği ülkeleri için bu başvuru EMA'ya (European Medicines Agency), Türkiye'de ise Sağlık Bakanlığı Türkiye İlaç ve Tıbbi Cihaz Kurumu'na yapılır. Ayrıca her ülkenin kendine özgü düzenleme kuruluşları mevcuttur ve bu kuruluşlara gerekli başvuruların yapılarak onay alınması gerekmektedir.

Geliştirilen ilaç piyasaya sürüldükten sonra yapılan klinik çalışmalar faz IV çalışmaları olarak adlandırılır. Faz IV denemeleri ayrıca satış sonrası gözetim denemeleri (post marketing surveillance trial) olarak da bilinir. Faz IV çalışmalar sırasında uzun süreli güvenirlik verileri toplanır. Böylece, klinik çalışmalar aşamasında ortaya çıkmayan yan etkiler bu faz IV sırasında rapor edilebilir. Ayrıca, ilaçla veya kullanıldığı hastalık ve hasta grubu ile ilgili ekonomik ve yaşam kalitesi çalışmaları bu fazda uygulanabilir. Bu fazda, ilaç daha büyük bir kitle üzerinde daha uzun bir zaman sürecinde gözlenerek herhangi nadir ya da uzun dönemli bir yan etki saptanabilir. Bu fazda ilacın herhangi bir zararlı etkisine rastlanması halinde, ilaç piyasadan geri çekilebilir.

SANAL TARAMA

HTS yöntemine alternatif olarak kullanılabilecek diğer bir yaklaşım ise kantitatif yapı-etki ilişkileri (quantitative structure–activity relationship, QSAR) yöntemi ile molekül aktivitelerinin teorik olarak kestirilmesi ve aktivite göstereceği düşünülen moleküller ile ilaç geliştirme çalışmalarına başlanmasıdır. Bu yaklaşıma sanal tarama (ST) adı verilmektedir. Özellikle son 20 yılda, makine öğrenimi yöntemleri kullanılarak moleküller sınıflandırılmakta (aktivite var-yok) ya da aktivite sıralamasına sokulmaktadırlar. Makine öğreniminin ilaç geliştirme çalışmalarında kullanıldığı ilk

çalışmalardan biri Sadowski ve Kubinyi (6) tarafından gerçekleştirilmiştir ve bu çalışmada yapay sinir ağları (YSA) kullanılarak ilaç özelliği gösteren ve göstermeyen moleküller sınıflandırılmıştır. Byvatov ve ark. (7) ve Zernov ve ark. (8) moleküllerin aktivitelerinin sınıflandırılmasında destek vektör makineleri (DVM) ve YSA algoritmalarının performanslarını karşılaştırmış ve DVM'nin performasının YSA'nın performansından daha iyi olduğunu ortaya koymuşlardır. Korkmaz ve ark. (3) ilaç moleküllerinin sınıflandırılma performansını arttırmak için farklı değişken seçim yöntemlerinin DVM'nin performansı üzerindeki etkilerini araştırmışlardır. Korkmaz ve ark. (4) 23 adet makine öğrenimi yönteminin performansını karşılaştırmış ve en iyi performans gösteren 10 algoritmayı kullanarak ilaç moleküllerini sınıflandırmak için web tabanlı bir uygulama geliştirmişlerdir. Naive Bayes, k-en yakın komşuluk, Bayes sinir ağları ve Random Forest (RF) algoritmaları da aktif ve inaktif molekülleri sınıflandırmada kullanılan diğer makine öğrenimi algoritmalarıdır (9,10). Aktivite tahmini için Gertrudes ve ark. (11) moleküllerin biyolojik aktivitesinin tahmininde çeşitli makine öğrenme yöntemlerinin performanslarını karşılaştırmıştır. Jorissen ve Gilson (12), Wassermann ve ark. (13), Agarwal ve ark. (14) ve Rathke ve ark. (15) aktivitelerine göre molekülleri sıralamak için DVM algoritmasını kullanmışlardır.

Diğer yandan, son yıllarda derin sinir ağları (DSA) birçok alanda sınıflandırma probleminin çözümü için oldukça iyi performanslar göstermiştir. Ma ve ark. (16) kantitatif yapı-aktivite ilişkilerinin tahmini için bir DSA modelini kullanmış ve DSA'nın RF modeline göre daha iyi performans gösterdiğini bulmuşlardır. Mayr ve ark. (17) bileşiklerin toksisitesini tahmin etmek için çok görevli bir DSA mimarisi kullanmış ve Tox21 yarışmasını kazanmışlardır.

Ramsundar ve ark. (18) çeşitli moleküler bileşik veri setlerine (PCBA, MUV, DUD-E, Tox21) çok görevli bir DSA algoritması uygulamışlardır. Koutsoukas ve ark. (19) bir DSA modelinin hiper-parametrelerinin optimizasyonunu araştırmış ve DSA modelinin performansını SVM, RF, NB ve kNN algoritmaları ile karşılaştırmışlardır. Lenselink ve ark. (20) bir DSA modelinin performansını, ChEMBL biyoaktivite veri seti kullanarak NB, RF, SVM ve lojistik regresyon ile karşılaştırmıştır.

ST deneysel olarak taranacak kimyasal kütüphanenin boyutunu azaltan bir hesaplama filtresi olduğundan, HTS yöntemine göre öncü bileşikleri bulma süresini ve maliyetini azaltabilmektedir. Günümüzde HTS yöntemi ile ilaç moleküllerine ilişkin deneysel olarak elde edilen veriler ücretsiz olarak erişilebilen veri tabanlarına

yüklenmektedir. İlaç moleküllerine ilişkin verileri içeren en büyük veri tabanlarından biri PubChem veri tabanıdır.

PUBCHEM VERİ TABANI

PubChem, 2004 yılında ABD Ulusal Sağlık Enstitüleri'nin (National Institutes of Health, NIH) Moleküler Kütüphaneler Yol Haritası Girişimleri'nin bir bileşeni olarak başlatılan kimyasal maddeler ve biyolojik etkinlikleri hakkında bilgi sağlayan bir veri tabanıdır. Araştırmacılar moleküllere ilişkin bilgileri indirmede ve bileşiklerin iki boyutlu yapılarını oluşturmada PubChem veritabanından yararlanabilmektedir. Son 15 yıldır PubChem, bilimsel araştırma topluluğu için kimyasal bir bilgi kaynağı olarak hizmet veren büyük bir sisteme dönüşmüştür. PubChem madde, bileşik ve bioassay olmak üzere birbirine bağlı üç veri tabanından oluşmaktadır. Bu üç veri tabanı tarafından sağlanan kimyasal örnek açıklamaları madde olarak adlandırılır ve açıklamalar madde veri tabanında tutulur. Madde veri tabanı, araştırmacılar tarafından PubChem'e veri sağlanması ile yerleştirilen kimyasal bilgileri içerir. Bu veri tabanında birbirinden bağımsız ayrı kayıtlar ile aynı molekül hakkında farklı açıklamalar tutulur.

Madde veri tabanı, madde kayıtlarının doğruluğunu korur ve araştırmacıların PubChem'e hangi bilgileri sağladığını görmelerine yardımcı olur. Bileşik veri tabanı, madde veri tabanından çıkarılan tekil kimyasal yapıları depolamaktadır. Tekil kimyasal yapılar madde veri tabanından çıkarılır ve bileşik veri tabanında saklanır.

Deneyler ile test edilen kimyasal maddelerin biyolojik aktivite verileri bioassay veri tabanında yer almaktadır. PubChem içerisinde bulunan veriler, üniversite laboratuvarları, devlet kurumları, ilaç şirketleri, kimyasal satıcılar, yayıncılar ve bir dizi kimyasal biyoloji kaynağı da dahil olmak üzere 350'den fazla paydaş tarafından sağlanmaktadır. Ayrıca PubChem ABD Gıda ve İlaç İdaresi, tekil madde tanımlayıcıları (unique ingredient identifiers, UNII) ve farmakolojik sınıflandırmalar da dahil olmak üzere önemli düzenleyici kurumlardan gelen verileri de barındırmaktadır. PubChem, yaklaşık 6 milyon patent belgesi ve 16 milyondan fazla tekil kimyasal yapı arasında, 329 milyondan fazla kimyasal madde-patent bağlantısı, 1800 yılından beri yayınlanan ABD, Avrupa ve Dünya Fikri Mülkiyet Örgütü patent belgelerini kapsayan bağlantılar sunmaktadır. PubChem, öncelikle HTS deneylerinden elde edilen büyük miktarda bioassay verisi içerir.

MOLEKÜLER DEĞİŞKENLER

HTS yöntemi ile elde edilen bioassay verilerinin makine öğrenimi yöntemleri ile analiz edilebilmeleri için aktif ya da inaktif olarak etiketlenmiş moleküllere ilişkin moleküler değişkenlerin hesaplanması gerekmektedir. PaDEL yazılımı moleküler değişkenleri hesaplamak için Yap (2011) (21) tarafından geliştirilen ücretsiz ve açık kaynak kodlu bir yazılımdır. Moleküler değişken, bir molekülün kimyasal bilgilerini sayıya ya da bazı standart deneylerin sonucuna dönüştüren mantıksal ve matematiksel işlemlerin sonucudur. Moleküler değişkenler kimyasal bileşikler için hesaplanır ve yeni bileşiklerin biyolojik aktivitelerinin öngörülmesi için QSAR modellerinde kullanılırlar. PaDEL yazılımı kimyasal bileşikler için 2757 adet moleküler değişken hesaplayabilmektedir. PaDEL yazılımı Java dili kullanılarak geliştirilmiştir ve hem kullanıcı arayüzü ile hem de Java kütüphanesi ile kullanılabilmektedir. Yazılım, moleküler değişkenlerin hesaplanmasını hızlandırmak için çoğu modern bilgisayarda bulunan çoklu işlemci çekirdeğinden yararlanmak için paralel programlama modeli kullanmaktadır.

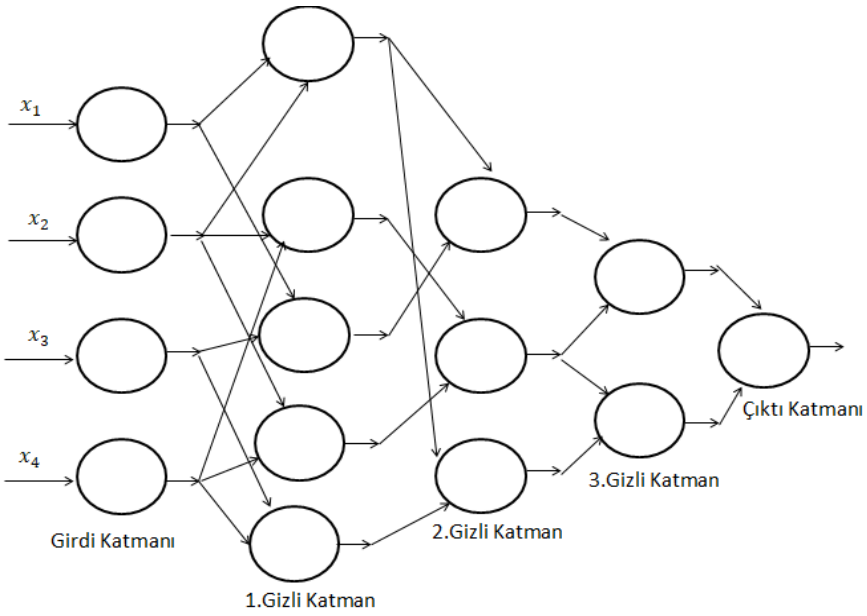
PaDEL yazılımının diğer mevcut yazılımlara (Dragon, Model, Molconn-Z ve PreADMET) göre en büyük avantajları; ücretsiz ve açık kaynak kodlu olması, hem kullanıcı arayüzü hem de komut satırı ile çalışabilmesi, tüm işletim sistemlerinde (Windows, Linux, MacOS) çalışabilmesi ve 90'dan fazla farklı moleküler dosya formatını destekleyebilmesidir. Bununla birlikte, diğer yazılımlar PaDEL yazılımından daha fazla moleküler değişken hesaplayabilmektedir.

DERİN SİNİR AĞLARI ALGORİTMASI

Perseptron algoritması sinir ağları konusundaki ilk algoritmalardan biridir (22). Bu ağda bir girdi katmanı bulunmaktadır. Doğrudan çıktıya bağlıdır. Karmaşık problemler için bu algoritmaya birden çok gizli katman eklenmiştir. Delta kuralı olarak adlandırılan öğrenme tekniğiyle her katmanın ağırlığı ayarlanabilmiştir (23). Bu tür sinir ağlarına daha fazla sayıda gizli katman (ikiden fazla) eklenmesiyle doğrusal olmayan karmaşık ilişkilerin de tespit edilebilmesi sağlanmıştır ve bu sinir ağlarına derin sinir ağları (DSA) adı verilmiştir (23).

Girdi, gizli ve çıktı katmanları derin öğrenme algoritmasının genel yapısını oluşturmaktadır (Şekil 3). Gizli katmanda bulunan nöronlar giriş ile çıkış nöronlarından, aktivasyon fonksiyonunun türünden, öğrenme algoritmasından ve ağ yapısından etkilenmektedir (24). Giriş katmanı, ağdaki girdi verilerinin nasıl

beslendiğini göstermektedir. Giriş katmanında bulunan nöronların sayısı genel olarak kullanılan verideki değişken sayısı ile aynı sayıdadır. Giriş katmanlarını bir veya daha fazla gizli katman takip eder. Klasik ileri beslemeli sinir ağlarındaki giriş katmanları bir sonraki gizli katmana tamamen bağlanır, fakat diğer ağ yapılarında giriş katmanı tam olarak bağlanamayabilir (25).



Şekil 3. Derin sinir ağları mimarisinin genel yapısı

Gizli katman, ileri beslemeli sinir ağında bir ya da daha fazla sayıda bulunabilir. Katmanlar arasındaki bağlantıların ağırlık değerleri, sinir ağlarının ham eğitim verilerinden çıkarılan öğrenilmiş bilgileri nasıl kodladığını gösterir. Gizli katmanlar, doğrusal olmayan sinir ağları fonksiyonlarının modellenmesine izin veren yapıdır (25). Çıktı katmanı, modelin tahminini ya da cevabını ortaya koymaktadır. Çıktı katmanı girdi katmanından gelen girdiyi temel alan bir çıktı verir (25). Nöronlar arasındaki bağlantılar ağırlıklar ile ilişkilidir. Bu ağırlıklar girdi değerinin önemini belirtmektedir. İlk ağırlıklar rastgele bir şekilde ayarlanmaktadır. Her nöron bir aktivasyon fonksiyonuna sahiptir ve aktivasyon fonksiyonunun amaçlarından biri nörondan elde edilen çıktıları standartlaştırmaktır.

Özellikle son yıllarda derin sinir ağları ses, video, metin gibi pek çok veri yapısının işlenebilmesinde oldukça başarılı sonuçlar ortaya koymuştur (26). Derin

öğrenme tekniklerinin daha ayrıntılı olarak verilebilecek uygulama alanları arasında aşağıdakiler bulunmaktadır (27):

1. Bilgi erişimi (information retrieval)
2. Çok modlu ve çok görevli öğrenme (multimodal and multitask learning)
3. Nesne tanıma ve bilgisayarlı görü (object recognition and computer vision)
4. Dil modelleme ve doğal dil işleme (language modeling and natural language processing)
5. Konuşma ve ses işleme (speech and audio processing)

Aktivasyon Fonksiyonları

Nöronların etkileşimini sağlayan sayısal bir fonksiyondur. Bir katmanın nöronlarının çıktısını bir sonraki katmana iletmesi için aktivasyon fonksiyonları kullanılmaktadır. Sinir ağındaki gizli katmanlar için ağın doğrusal olmayan modelleme yapabilmesinde aktivasyon fonksiyonları tercih edilir.

Doğrusal aktivasyon fonksiyonu: Sinir ağlarının giriş katmanında bu aktivasyon fonksiyonu kullanılmaktadır. Doğrusal aktivasyon fonksiyonu temelde bir birim (identity) fonksiyonudur ve $f(x) = Wx$ fonksiyonu ile belirtilir. Burada, bağımlı değişken ile bağımsız değişken arasında oransal bir ilişki vardır.

Sigmoid aktivasyon fonksiyonu: Sonsuz aralıktaki bağımsız değişkenleri 0 ile 1 aralığındaki olasılıklara dönüştüren bir fonksiyondur. Sinir ağının çıktı katmanında kullanılır. İkili sınıflandırma yapmak amacı ile tercih edilir ve her sınıf için bağımsız bir olasılık üretir.

Softmax aktivasyon fonksiyonu: Softmax, ikili sınıflandırmanın yanı sıra sürekli verilere de uygulanabilen ve çoklu karar sınırları içerebildiğinden lojistik regresyonun genelleştirilmiş halidir. Multinomial etiketleme sistemlerini yönetmektedir. Softmax, genellikle bir sınıflandırıcının çıkış katmanında bulunan işlevdir. Softmax fonksiyonu

yapay sinir ağının ürettiği skor değerlerinden yararlanarak olasılık temelli loss (kayıp) fonksiyonu ortaya çıkarmaktadır. Bu fonksiyon, sınıfları bir ağaç yapısına dönüştürür ve softmax sınıflandırıcısı dallanmayı yönetmek için ağacın her bir düğümünde eğitilir.

Düzleştirilmiş doğrusal birim (rectified linear units (ReLU)) aktivasyon fonksiyonu: Girdiler belirli bir değerin üzerindeyken düğümlerin aktif olduğu dönüşümdür. Girdi sıfırın altındayken çıktı sıfırdır fakat girdi belirli bir eşiğin üstüne çıktığında bu aktivasyon fonksiyonu, $f(x) = \max(0, x)$, bağımlı değişken ile doğrusal bir ilişkiye sahiptir.

Gizli katmanlarda yapılan matematiksel işlemler sayesinde doğrusal yapıda olan ağı doğrusal olmayan yapıya dönüştürmek için ReLU aktivasyon fonksiyonu kullanılır (28). ReLU aktivasyon fonksiyonu sigmoid ve tanh aktivasyon fonksiyonları ile karşılaştırıldığında gradyanların yok olma problemi (vanishing gradient problem) gözlenmemektedir.

Gradyan İnişi Optimizasyon Algoritmaları (Gradient Descent Optimization Algorithms)

Gradyan inişi, sinir ağlarını optimize etmede ve performansını iyileştirmede kullanılan en yaygın algoritmalardan biridir. Gradyan inişi, $\nabla_\theta J(\theta)$ gibi parametrelerin amaç fonksiyonunu gradyanın karşı yönünde olacak şekilde parametreleri güncelleyerek $\theta \in R^d$ model parametreleri tarafından $J(\theta)$ amaç fonksiyonunu minimize etmenin bir yoludur.

Stokastik gradyan inişi (Stochastic gradient descent (SGD)): Stokastik, rastgele bir olasılık ile ilişkili olan süreci ifade etmektedir. Stokastik Gradyan İnişinde (SGD) her bir yineleme için ayarlanan verilerin tamamı yerine sadece tek bir örnek kullanılır. Örnek rastgele karıştırılır ve yinelemeyi gerçekleştirmek için seçilir. Bu durumda SGD tüm örneklerde amaç fonksiyonunun gradyanının toplamı yerine her bir yinelemede tek bir örneğin amaç fonksiyonunun gradyanının bulunmasını sağlar. SGD hızlı bir optimizasyon algoritmasıdır ve çevrimiçi öğrenme için de kullanılabilir. SGD amaç fonksiyonu ağır dalgalanmalara neden olan yüksek bir varyans ile sık sık güncellemeler yapar.

SGD'de her eğitim örneği x^i ve etiket y^i için bir parametre güncellemesi gerçekleştirir:

$$\theta_{t+1} = \theta_t - \eta . \nabla_\theta J(\theta_t; x^{(i)}; y^{(i)}) \quad (1)$$

Nesterov'un hızlandırılmış gradyanı (Nesterov's accelerated gradient): Dışbükey fonksiyonları optimize etmek için Nesterov (1983) (29) tarafından geliştirilmiştir. Standart momentum metodu önce mevcut konumdaki gradyanı hesaplar ve ardından güncellenmiş biriken gradyan yönünde büyük bir sıçrama yapar. Daha sonra sonuna kadar gradyan ölçülür ve düzeltmeler yapılır. Güncelleme kuralı aşağıda belirtilmektedir:

$$\vartheta_{t+1} = \gamma \vartheta_t + \eta \nabla_\theta J(\theta - \gamma \vartheta_{t-1}) \quad (2)$$

$$\theta_{t+1} = \theta_t - \vartheta_t$$

Adagrad: Öğrenme hızını parametrelere uyarlayan, seyrek parametreler için daha büyük güncellemeler ve sık parametreler için daha küçük güncellemeler gerçekleştiren gradyan tabanlı bir optimizasyon algoritmasıdır. Bu yüzden seyrek verilerle uğraşmak için çok uygundur. Güncelleme kuralı aşağıda belirtilmiştir:

$$G_{t,i} = G_{t-1,i} + (\nabla_{\theta_t} j(\theta_{t,i}))^2 \quad (3)$$

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,i} + \epsilon}} \nabla_{\theta_t} J(\theta_{t,i})$$

Burada $G_{t,i}$: t anında θ_i parametresine göre hesaplanmış, amaç fonksiyonunun eğim değerini ifade etmektedir.

Adagrad algoritmasının avantajlarından biri de öğrenme hızını manuel olarak ayarlama ihtiyacını ortadan kaldırmasıdır. Diğer yandan, bu algoritmanın zayıf noktalarından birisi paydada kare gradyanların birikmesidir. Eklenen her terim pozitif

olduğundan biriken toplam gradyan, eğitim sırasında artmaya devam eder. Bu da öğrenme hızının küçülmesine ve nihayetinde sonsuz derecede küçük olmasına neden olur ve bu noktada algoritma artık ek bilgi edinemez.

Adadelta: Tüm geçmiş kare gradyanları biriktirmek yerine, hareketli gradyan güncellemelerinin bir penceresine dayanan öğrenme hızlarını uyarlayan daha güçlü bir Adagrad uzantısıdır. Bu sayede Adadelta birçok güncelleme yapıldığında bile öğrenmeye devam eder. Güncelleme kuralı aşağıda belirtilmiştir:

$G_{t,i} = \gamma G_{(t-1,i)} + (1-\gamma)\ (\nabla_{\theta_t} j(\theta_{t,i}))^2$ ← Kare Gradyanın Hareketli Ortalaması (4)

$S_t = \gamma S_{t-1} + (1-\gamma)\ \nabla_{\theta_t}{}^2$ ← Kare Deltanın Hareketli Ortalaması

$\nabla_{\theta_t} = -\frac{\sqrt{S_{t-1} + \epsilon}}{\sqrt{G_{t,ii} + \epsilon}} \nabla_{\theta_t} J(\theta_{t,i})$ ← Delta, parametrenin ne kadar güncelleneceğine karar verir.

$\theta_{t+1} = \theta_t + \nabla_{\theta_t}$

Adam: Düşük dereceli momentlerin uyarlamalı tahminlerine dayanan stokastik amaç fonksiyonlarının birinci dereceden gradyan tabanlı optimizasyonu için tasarlanan bir algoritmadır.

Bu optimizasyon algoritmasının uygulanması basit olup hesaplama açısından etkindir. Adam optimizasyon algoritması az miktarda bellek kapasitesine ihtiyaç duyar. Veri ya da parametrelerin büyük olduğu problemler için uygundur.

Ayrıca sabit olmayan amaçlar ve çok gürültülü veya seyrek gradyanlar ile ilgili problemler için de kullanılabilmektedir. Güncelleme kuralı aşağıda ifade edilmektedir:

$$g_t = \nabla_\theta J(\theta) \quad (5)$$

$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$ ← Momentum terimi

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)\, g_t^2 \longleftarrow \text{RMSprop terimi}$$

$$\widehat{m}_t = \frac{m_t}{1-\beta_1^t} \longleftarrow \text{Yan düzeltme birinci moment terimi}$$

$$\widehat{v}_t = \frac{v_t}{1-\beta_2^t} \longleftarrow \text{Yan düzeltme ikinci moment terimi}$$

$$\theta_{t+1} = \theta_t - \eta \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t} + \epsilon} \longleftarrow \text{Güncelleme parametresi}$$

Burada g_t : t anında θ parametresine göre hesaplanmış amaç fonksiyonunun eğitimi olup $\beta_1 = 0{,}9$, $\beta_2 = 0{,}999$ (β_1 ve β_2 moment tahminleri için üstel bozulma oranları), $\epsilon = 10^{-8}$, $J(\theta)$: θ parametreli stokastik amaç fonksiyonunu, η: öğrenme hızını belirtmektedir.

Sinir Ağlarında Model Performansının İyileştirilmesi

Seyreltme: Derin sinir ağlarında oluşturulan modelin aşırı öğrenme yaptığı durumları engellemek için seyreltme (dropout) yöntemi kullanılır. Seyreltme, birçok model türünde kullanılabilen güçlü bir düzenleme yöntemidir. Seyreltme yöntemi, SGD ve her türlü sinir ağı mimarisi ile çalışmaktadır. Seyreltme yöntemi, bir birimin aktivasyonlarını geçici olarak inaktif hale getirmektedir.

Seyreltme işlemi sinir ağı katmanındaki nöronlar için 0 ile 1 arasındaki olasılık değerleri (bir aktivasyonu kaldırma ya da tutma olasılığı) ile yapılır. Giriş katmanında ve özellikle gürültülü ya da seyrek veri kümelerinde seyreltme yöntemi kullanılmamaktadır. Nöronlar rastgele atlanarak algılayıcılar arasındaki senkronize uyum önlenmekte ve bu sayede modellerde tutulan verilerde daha iyi genelleme yapılmasına olanak sağlanmaktadır.

Öğrenme hızı (learning rate): Bir sinir ağının kayıp fonksiyon alanını geçerken x parametre vektörüne attığı adımların boyutunu ölçekleyen bir katsayıdır (η). Öğrenme hızı, sinir ağının tahmin hatalarını en aza indirgemek için optimizasyon sırasında ayarlanan parametre miktarını etkilemektedir. Algoritmanın bir sonraki adımı için gradyanın ne kadarının kullanılması gerektiğini belirler. Minimal hataya yaklaştıkça ve gradyan düzleştikçe adım boyutu kısalma eğilimindedir. Öğrenme hızı katsayısı büyük (örneğin 1) olduğunda parametreler hızlı ve büyük adımlar ile

güncellenirken, öğrenme hızı katsayısı küçük (örneğin 0.00001) olduğunda parametreler yavaş ve küçük adımlar ile güncellenir.

Çok büyük bir öğrenme hızı algoritmanın hiç durmadan minimum kaybın her iki tarafında ileri ve geri sıçrama yapmasını sağlayarak global minimum noktasını kaçırmasına neden olabilir. Diğer yandan, küçük öğrenme hızları ile global minimum daha etkin bir şekilde bulunabilir ancak bu hataları bulmak çok uzun zaman alabilir ve işlem yükünü arttırabilir. Bu nedenle optimal bir öğrenme hızının bulunması hem global minimum noktasının doğru bir şekilde bulunmasına hem de işlem yükünün artmamasına olanak sağlayacaktır.

Mini-batch boyutu: Derin öğrenme çalışmalarında veri setindeki bütün verileri aynı anda işleyerek öğrenme işlemini gerçekleştirmek hem zaman hem de kapasite açısından maliyetli bir süreçtir. Bu nedenle veri seti küçük gruplar halinde ayrılarak öğrenme işlemi seçilen bu küçük gruplar üzerinde yapılır. Bu durumda mini-batch birden çok girdinin parçalar şeklinde işlenmesi olarak tanımlanmaktadır.

Modeldeki mini-batch parametresi modelin aynı anda kaç veriyi işleyeceğini belirtir. Verilerin gruplar halinde işlenmesinde (mini-batch) kaybın arttığı fakat zamandan kazanıldığı gözlenmektedir. Mini-batch değeri 1 olarak belirlendiğinde SGD ile aynı işlevi görür. Diğer bir deyişle her iterasyonda yalnızca tek bir veri üzerinde işlem yapar hale gelmektedir. Mini-batch değerinin eğitim kümesi bütün elemanların sayısına eşit ise eğitim kümesindeki tüm veriler eğitime gireceğinden yapılan işlem Toplu Gradyan İnişi (Batch Gradient Descent) ile aynı işlevi görür. Mini-batch değeri seçiminde optimal değer, 1 ile eğitim kümesindeki bütün verilerin sayısı arasında olacak şekilde belirlenmelidir. Bu durumda öğrenme hızlı bir şekilde gerçekleşecektir.

Kayıp Fonksiyonları (Loss Functions)

Derin sinir ağlarında tasarlanan modelin hata oranını aynı zamanda başarısını ölçen bir fonksiyondur. Kayıp fonksiyonunun tanımlandığı katman derin ağların son katmanıdır. Kayıp fonksiyonu, modelin yaptığı tahminin gerçek değerden ne kadar farklı olduğunu ölçmektedir. Literatürde minimizasyon durumunda kayıp fonksiyonu, maliyet fonksiyonu ya da hata fonksiyonu olarak da tanımlanabilmektedir (31). Eğitim sırasında en aza indirgenmek istenilen miktardır. Kayıp fonksiyonu eğitim sonucunda elde edilen w ağırlık ve b bias parametrelerinin sorunun çözümü için ne kadar uygun

olduğunu ölçmektedir. İkili sınıflandırma problemlerinde en sık kullanılan kayıp fonksiyonları aşağıda açıklanmıştır.

Kategorik çapraz entropi kayıp fonksiyonu (categorical cross entropy loss function): Log Softmax ve Negative Log Likelihood fonksiyonlarından türetilmiştir. Hem iki hem de çok kategorili (ikiden fazla) sınıflandırma problemlerinde kullanılır. Tek etiket sınıflandırması için kullanılan kayıp fonksiyonudur. Yani bir örnek yalnızca bir sınıfa ait olabilir. Bu kayıp fonksiyonu için olabilirlik fonksiyonu aşağıdaki gibi yazılabilir:

$$L(y, \hat{y}) = -\sum_{j=0}^{M} \sum_{i=0}^{N} (y_{ij} * \log(\widehat{y_{ij}})) \quad (6)$$

Burada $\hat{y}$ kestirilen değerdir.

İkili çapraz entropi kayıp fonksiyonu (binary cross entropy loss function): Sigmoid çapraz entropi kaybı olarak da bilinmektedir. İki sınıflı problemlerde kullanılan kayıp fonksiyonudur. Bu kayıp fonksiyonu için olabilirlik fonksiyonu aşağıdaki gibi yazılabilir:

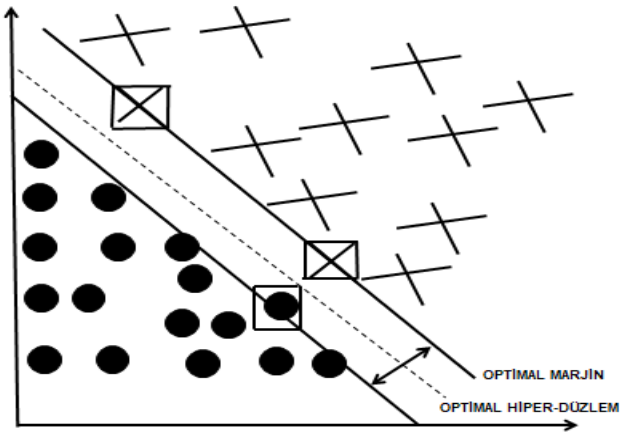
$$L(y, \hat{y}) = -\frac{1}{N} \sum_{i=0}^{N} (y * \log(\widehat{y_i}) + (1 - y) * \log(1 - \widehat{y_i})) \quad (7)$$

Burada $\hat{y}$ kestirilen değerdir.

DESTEK VEKTÖR MAKİNELERİ ALGORİTMASI

Makine öğrenimi algoritmalarının seçiminde en önemli kriterlerden biri algoritmanın genelleme performansıdır. Model ve parametre seçimi, bağımsız değişkenlerin sayısı ve yapısı, eğitim verisi gibi faktörler algoritmaların genelleme performansı üzerinde etkilidir. Destek vektör makineleri (DVM), makine öğrenimi problemlerinden biri olan sınıflandırma sorununun çözümüne özgü tasarlanmış, genelleme performansı oldukça yüksek olan ve yüksek boyutlu verilerle çalışma imkânı sunan bir sınıflandırma algoritmasıdır (32). Cortes ve Vapnik (33) tarafından istatistiksel öğrenme teorisi ile yapısal riski en aza indirme prensibine dayanarak

geliştirilmiş DVM algoritması dağılımdan bağımsız olarak çalışabilen, ikili sınıflandırma ve regresyon işlemlerini gerçekleştirebilen bir algoritmadır. DVM yapısal risk minimizasyonu ve Vapnik-Chervonenkis (VC) teorisinden yararlanarak, çok sayıda aday model arasından beklenen riski ya da genelleme hatasını minimum yapacak modeli bulabilmektedir (34). Ayrıca, DVM hem ikili hem de çoklu sınıflandırma problemlerinde kullanılabilen bir makine öğrenimi algoritmasıdır. DVM algoritmasının amacı, farklı sınıflara ait destek vektörler arasındaki uzaklığı optimal şekilde ayırabilen bir hiper-düzlem bulmaktır (Şekil 4).



Şekil 4. İki boyutlu uzayda sınıflandırılabilen problem (33).

Sınıflandırma probleminin doğrusal olarak çözülemediği durumlarda, DVM doğrusal olmayan örnek uzayını, örneklerin doğrusal olarak ayrılabileceği yüksek bir boyuta aktararak bu yüksek boyutlu uzayda sınıflar arasındaki optimal marjini bulmaya çalışır (35).

Sınıflandırma için oluşturulan hiper-düzlemler arasında "ayırıcı hiper-düzlem" denilen ve optimal sınıra sahip sadece bir adet hiper-düzlem bulunmaktadır. Bu ayırıcı hiper-düzlem üzerindeki vektörlere de "destek vektörleri" denilmektedir.

DVM ile sınıflandırılacak eğitim veri kümesinin N sayıda örnekten oluştuğu ve i = 1,...,N olmak üzere $\{x_i , y_i\}$ ile gösterildiği varsayılsın. Burada $x_i \epsilon R^d$ olmak üzere d-boyutlu bir uzayda özellikler vektörünü (giriş vektörü), $y_i \epsilon$ {-1,+1} olmak üzere sınıf etiketlerini (çıkış vektörü) tanımlamaktadır. w hiper-düzlemin normal vektörü ve b eğim değeri olmak üzere eğitim kümesinin aşağıdaki şartı sağlaması gerekmektedir (34).

$$y_i(w.x_i + b) \geq +1 \text{ , i= 1,…,N} \quad (8)$$

DVM doğrusal ve doğrusal olmayan destek vektör makineleri olmak üzere iki gruba ayrılmaktadır. Gerçek hayattaki problemlerin büyük çoğunluğu doğrusal olarak ayrılamayan problemlerden oluşmaktadır. Doğrusal olarak ayrılabilen sınıflar arasındaki maksimum sınırın bulunması oldukça kolaydır.

Fakat doğrusal olarak ayrılamayan sınıflar önce doğrusal olarak ayrılabileceği yüksek boyutlu bir uzaya aktarılmalıdır (35). Daha sonra, sınıflandırma problemi bu yeni yüksek boyutlu uzayda çözülür.

Çekirdek Fonksiyonları

Doğrusal olmayan problemlere çözüm bulmada alternatif olarak çekirdek fonksiyonlar ile örnekler daha yüksek boyutlu ve doğrusal olarak ayrılabilecekleri bir uzaya taşınır ve çözüm bu yeni uzayda aranır. Giriş uzayındaki eğitim verilerini H Öklid uzayına taşıyabilen Φ fonksiyonunu inceleyelim (36, 37, 38).

$\phi : R^d \rightarrow H$ olur. (46)

DVM doğrusal olarak ayrılamayan veriyi doğrusal olarak ayrılabileceği yüksek boyutlu değişken uzayına taşımaktadır. Bu sayede en uygun ayırıcı hiper-düzlem bu değişken uzayında bulunabilir. Giriş uzayındaki eğitim verisi çekirdek fonksiyonlarından yararlanarak değişken uzayına aktarılır (36). Bu durumda DVM'nin eğitim aşaması sadece H uzayındaki verilerin $\phi(x_i).\phi(x_j)$ iç çarpımlarına bağlıdır.

İç çarpımı K ile gösterirsek:

$K(x_i, x_j) = \phi(x_i).\phi(x_j)$ (47)

olarak ifade edilir. Buradaki K fonksiyonu çekirdek fonksiyonu (kernel function) olarak tanımlanmaktadır. Test aşamasındaki sistemin test örneğinin alacağı değer:
Sonuç olarak karar fonksiyonu:

$$f(x) = \sum_{i=1}^{ls} \alpha_i y_i \, \phi(x_i).\phi(x) + b = \sum_{i=1}^{ls} \alpha_i y_i \, K(x_i, x) + b \quad (48)$$

fonksiyonunun (eşitlik (36)'nın) işareti ile belirlenir. Karar fonksiyonu yeniden yazılırsa;

Karar fonksiyonu = $sign\big(f(x)\big) = sign\big(\sum_{i=1}^{ls} \alpha_i y_i \, \mathrm{K}\,(x_i\,, x) + b\big)$ (49)

olur (36). Bu fonksiyonda ls, destek vektörlerin sayısı, x_i ise destek vektörleridir. Çekirdek fonksiyonu olarak çeşitli fonksiyonlar kullanılabilir. Sık kullanılan çekirdek fonksiyonları Tablo 1'de verilmiştir.

Tablo 1. DVM'de kullanımı uygun olan çekirdek fonksiyonları (38).

Çekirdek Fonksiyonu	Matematiksel Gösterim
Doğrusal	k (x,$x^{\prime}$) = $x^{T}.x^{\prime}$ + c
Polinomial	k (x,$x^{\prime}$) = $(ax^{T}x^{\prime} + c)^{d}$
Gaussian	k (x,$x^{\prime}$) = exp (- $\|x - x^{\prime}\|^{2}/2\sigma^{2}$)
Eksponansiyel	k (x,$x^{\prime}$) = exp (- $\|x - x^{\prime}\|/2\sigma^{2}$)

RANDOM FOREST ALGORİTMASI

Random Forest (RF), orijinal veri setinden rastgele ve yerine koyarak seçilen bootstrap örnekleme yöntemi ile Breiman (2001) tarafından geliştirilmiş tahmin başarısı oldukça yüksek olan karar ağacı tabanlı bir makine öğrenimi yöntemidir (39). RF hem sınıflandırma hem de regresyon için geliştirilen topluluk (ensemble) öğrenme tekniğidir. İlgili verideki sınıf değişkeni kategorik ise sınıflandırma, sürekli ise regresyon ağaçları oluşturulmaktadır. RF yüksek boyutlu karmaşık veri yapılarında ve kayıp veri olduğu durumda kullanılabilmektedir. RF, çok sayıdaki karar ağacının birleşmesiyle ortaya çıkmaktadır ve bireysel ağaçlar tarafından oylanarak kazanan sınıf belirlenmektedir (40).

RF sınıflama yönteminde her bir özelliğin "düğüm" tarafından temsil edildiği en son yapının "yaprak" en üst yapının "kök" yaprak ve kök arasında kalan yapıların da "dal" olarak tanımlandığı çalışma sistemi bulunmaktadır (41). RF sınıflandırma amacı ile kullanıldığında ağaçlar ayrı ayrı incelenir ve her biri hedef sınıfı tahmin etmek için işlenir. Nihai sınıflandırma sonucu ağaçların ayrı ayrı elde ettiği tahminlerin

çoğunluk oyu (majority vote) ile hesaplanır (42). RF yönteminde CART (Classification and Regression Tree) algoritmasıyla ağaçlar oluşturulur ve ağaçlar budanmaz. Her ağacın oluşumu sırasında ve yeni bir düğüm eklenirken rastgele seçilen özellik alt kümesi, giriş özellikleri kümesinden seçilmektedir. Bu alt kümedeki özellikler daha sonra araştırılmakta ve en iyi bölme sonucuna sahip olan seçilmektedir (42). CART algoritması bilgi kazancını (information gain) ya da Gini indeksini kullanarak veri setinin hangi değişkenden başlayarak dallara ayrılacağına karar verir (41,43).

RF algoritması aşağıdaki adımlardan oluşmaktadır (41):

- Orijinal veri setinden n adet bootstrap örneklem oluşturulur. Oluşturulan her örneklemin 3'te 2'si ağacı oluşturmak için eğitim verisi olarak kullanılır (inBag).
- Her bootstrap örneklem içerisinde sınıflandırma aşağıdaki adımlar izlenerek oluşturulur:
    - inBag veri setinden her düğümde bütün tahmin değişkenleri içerisinden en iyi değişkeni seçmek yerine rastgele m tane tahmin değişkeni seçilir ve bunların içerisinden dallara en iyi ayıracak (en çok bilgi kazancı sağlayan) olanı belirlenir.
    - Belirlenen tahmin değişkeni için en iyi dallanma kriteri Gini indeksi ile hesaplanır ve hesaplanan değere göre veri setini her düğümde iki alt dala ayırır.
    - Yukarıda verilen adımlar aşağıya doğru yaprak düğüm elde edilene kadar her düğümde tekrarlanır.
- n tane ağacın ayrı ayrı yaptığı tahminler bir araya getirilir ve en çok oyu alan sınıf son tahmin olarak belirlenir.

Breiman tarafından varsayılan m değeri sınıflandırma ağaçları oluşturulurken $p^{1/2}$ olarak önerilmiştir. Burada, p değeri toplam tahmin edici değişken sayısını ifade etmektedir (41).

Veri setindeki hata oranını hesaplamak için aşağıdaki adımlar izlenir (41):

- Bootstrap aşamasında karar ağacı oluşturulurken bootstrap örnekleme, ağaç oluşturulacak veri (in bag) ve ağaç oluşturmak için kullanılmayan veri (out of bag, OOB) olarak iki parçaya ayrılır. OOB ile oluşturulan RF modeli test edilir ve hata oranı tahmini yapılır.
- Yapılan OOB tahminleri bir araya getirilir ve ormanın hata oranı kestirimi yapılır.

Bootstrap Örnekleme

Veri setindeki verilerden her defasında yerine koyarak farklı örnekler seçip yeni bir veri seti oluşturma işlemi bootstrap yöntemi olarak tanımlanmaktadır (39). Bootstrap yöntemi ile elde edilen örnek veri setinden çıkarılmaksızın seçim işlemlerine devam edildiği için eğitim veri setindeki bir örneğin birden fazla tekrar edebilme durumu bulunmaktadır. Eğitim veri setinin oluşturulmasının ardından eğitim veri setine alınmayan tüm örnekler test veri setine aktarılmaktadır. Test veri setinde bulunan her örnek sadece bir kez tekrar edebilmektedir (32).

Bootstrap metodu ile örnek seçimi aşağıdaki gibi ifade edilmektedir:

N adet gözlemden oluşan veri seti $X = (X_1 , X_2 , X_3 , X_4 , \dots , X_N)$ olsun. Bu veri setinden 1/N olasılıkla şansa bağlı bootstrap örnek veri seti $X_i^* = (X_1^*, X_2^* , X_3^* , X_4^* , \dots , X_N^*)$ elde edilmektedir. Bu işlem ne kadar örneklem oluşturulmak isteniyorsa o kadar tekrarlanarak istenilen kadar bootstrap veri seti oluşturulabilmektedir (39).

Bagging Yöntemi

Bagging (Bootstrap Aggregating), bootstrap tekniği ile seçilen örneklemlerle oluşturulan çok sayıda karar ağacının yapmış olduğu tahminleri toplayarak nihai sınıf tahmini yapan bir yöntemdir. Oluşturulan ağaç yapısında orijinal veri setindeki tüm değişkenleri kullanmaktadır (41).

Sınıflama ve regresyon modelleri için uygulanabilen aşırı öğrenmeye karşı güçlü olan doğru sınıflandırma oranını arttıran ve varyans düşürücü etkisi olan veri setinde kayıp verilerin yer aldığı durumlarda başarılı sınıflandırma ortaya çıkaran topluluk öğrenme şeklidir (39).

Bagging tekniğinde veri eğitim ile test veri seti olarak iki gruba ayrılır ve ayrılan eğitim setinden bootstrap örnekleme yöntemi ile m sayıda ağaç oluşturulmaktadır. Oluşturulan ağaçlarda dallara ayırıcı nitelikteki değişken tüm değişkenler içinden rastgele seçilir oylama işlemi yapıldığında en yüksek oyu alan sınıf nihai sınıf olarak ortaya çıkmaktadır (46).

Boosting Yöntemi

Boosting, verilen eğitim algoritmalarının doğruluğunu artırmak için kullanılan genel bir yöntemdir (47). Boosting tekniğinde amaç, veri setine farklı ağırlıklar verildiğinde oluşan ağaçlar topluluğundan tahminlerde bulunmaktır. Başlangıçta bütün gözlemler eşit ağırlığa sahiptir. Ağaç topluluğu büyüdükçe, problem bilgisine dayalı olarak ağırlıklandırmalar düzenlenir. Yanlış sınıflandırılan gözlemlerin ağırlığı

arttırılırken nadiren yanlış sınıflandırılan gözlemlerin ağırlığı azaltılır. Böylece ağaçlar zor sınıflandırılan gözlemler karşısında kendini düzenleyebilme imkânı kazanmaktadır (48). Bu yönteminin temelinde sınıflandırıcı serisinin oluşturulması vardır (49).

Zayıf ya da temel sınıflandırıcı olarak adlandırılan bu bireysel sınıflandırıcılar, Karar Ağaçları, Perseptron Öğrenme Kuralı, Maksimum Olabilirlik Kuralı gibi kurallar olabilmektedir. Her bir iterasyon sırasında, bir zayıf sınıflandırıcı seçilir ve sınıflandırılmamış vektörlere dayanan farklı bir örnek dağılımı kullanılarak eğitilir.

Boosting yönteminin diğer yöntemlere göre önemli bir avantajı hassas ayarlamanın, karmaşık ve lineer olmayan optimizasyon yapmanın gerekli olmamasıdır.

Bu yöntemde seriye ait bir önceki sınıflandırıcıların hatalı olarak tahmin ettiği örnekler bir sonraki sınıflandırıcının kullanacağı eğitim setindeki doğru tahmin edilen verilere göre daha fazla tekrar edilerek örnekleri daha doğru tahmin edebilen sınıflandırıcı oluşturmak istenir. Boosting yönteminde her bir gözleme ait hata durumuna göre bir ağırlık değerinin verilmesi mantığı bulunmaktadır. Boosting yönteminde en yaygın kullanılan algoritma Adaboost algoritmasıdır (50).

GEREÇ VE YÖNTEMLER

VERİ SETLERİ

Bu çalışmada PubChem veri tabanından elde edilen 5 adet bioassay verisi kullanılmıştır ve kullanılan verilere ilişkin bilgiler Tablo 2'de özetlenmiştir.

1) **AID652178:** Bu bioassay Alzheimer hastalığı ve şizofreni ile ilişkili bilişsel dejenerasyonun tedavisinde önemli bir etkiye sahip olan bir transmembran alan reseptörü (GQ-bağlı GPCR M1 Muskarinik reseptör) için oluşturulmuştur. Bu bioassay veri seti içerisinde 178 aktif ve 897 inaktif bileşik olmak üzere toplam 1075 bileşik bulunmaktadır.
2) **AID1053187**: Bu bioassay verisi Muskarinik M1 reseptörü için oluşturulmuş olup farklı HTS deneylerinden (AID628, AID677, AID859, AID860) elde edilen sonuçları içermektedir. Bu bioassay veri seti içerisinde 420 aktif ve 1172 inaktif olmak üzere toplam 1592 adet bileşik bulunmaktadır.
3) **AID1053196**: Bu bioassay Choline Transporter (CHT) inhibitörleri için geliştirilmiş olup farklı HTS deneylerinden (AID488975, AID493221, AID504840, AID588401, AID493222, AID602208, AID49322) elde edilen sonuçları içermektedir. Bu bioassay veri seti içerisinde 231 aktif ve 2058 inaktif olmak üzere toplam 2289 adet bileşik bulunmaktadır.
4) **AID1159608:** Bu bioassay verisi Nöropeptid Y reseptörü Y2'nin (NPY-Y2) antagonistleri için oluşturulmuş olup farklı HTS deneylerinden (AID793, AID1257, AID1256, AID1279, AID1272, AID2210, AID2212, AID2224) elde

edilen sonuçları içermektedir. Bu bioassay veri seti içerisinde 624 aktif ve 637 inaktif olmak üzere toplam 1261 adet bileşik bulunmaktadır.

5) **AID115909:** Bu bioassay verisi Fenolik amino karboksilik asitlerin esterleri ve laktonları, demir şelasyonu için ön ilaçlardan oluşmuştur. Bu bioassay veri seti içerisinde 717 aktif ve 1070 inaktif olmak üzere toplam 1787 adet bileşik bulunmaktadır.

Tablo 2. Beş veri setinin sınıf değişkenlerine ilişkin bilgiler

Veri seti	Aktif	İnaktif	Toplam	Aktif / İnaktif
AID652178	178	897	1075	1:5
AID1053187	420	1172	1592	1:3
AID1053196	231	2058	2289	1:9
AID1159608	624	637	1261	1:1
AID115909	717	1070	1787	1:1,5

PERFORMANS ÖLÇÜLERİ

Eğitilen makine öğrenimi algoritmalarının test seti performanslarını değerlendirmek için aşağıdaki performans ölçüleri Tablo 3'te verilen çapraz tablo yardımıyla hesaplanmıştır.

Tablo 3. Performans ölçülerini hesaplamak için kullanılan çapraz tablo

		GERÇEK	
		Aktif	**İnaktif**
TAHMİN	**Aktif**	Doğru Pozitif (DP)	Yanlış Pozitif (YP)
	İnaktif	Yanlış Negatif (YN)	Doğru Negatif (DN)

Doğru Pozitif (DP) : Gerçekte aktif olan moleküllerden modelin aktif olarak tahmin ettiği moleküllerin sayısıdır.

Yanlış Pozitif (YP) : Gerçekte inaktif olan moleküllerden modelin aktif olarak tahmin ettiği moleküllerin sayısıdır.

Yanlış Negatif (YN) : Gerçekte aktif olan moleküllerden modelin inaktif olarak tahmin ettiği moleküllerin sayısıdır.

Doğru Negatif (DN): Gerçekte inaktif olan moleküllerden modelin inaktif olarak tahmin ettiği moleküllerin sayısıdır.

Dengeli Doğruluk Oranı (DDO): Her bir sınıfın ayrı ayrı doğrularının oranının ortalaması olarak hesaplanmaktadır. Dengesiz veri setlerindeki abartılı performans tahminlerini önlemek amacıyla dengeli doğruluk oranı kullanılır.

$$\text{DDO} = \frac{1}{2}\left(\frac{DP}{DP+YN} + \frac{DN}{DN+YP}\right) \tag{50}$$

Duyarlılık (Duy): Testin belirli bir hastalığı olan hastaları tespit etme yeteneğini ifade etmektedir. Modelin yanlış negatifleri ne kadar iyi önlediğini belirtir.

$$\text{Duyarlılık} = \frac{DP}{DP + YN} \tag{51}$$

Pozitif Kestirim Değeri (PKD) : Doğru sınıflandırılan pozitif örneklerin toplam pozitif tahmin edilen örneklere oranıdır.

$$\text{PKD} = \frac{DP}{DP+YP} \tag{52}$$

F1 Skor (F1 Score) : İkili sınıflandırmada F1 skoru modelin doğruluğunun ölçüsü olarak düşünülmektedir. 0 (kötü) ile 1 (iyi) arasında değer alır. F1 skoru duyarlılık ve pozitif kestirim değerinin harmonik ortalamasıdır. F1 skor değeri ne kadar yüksek ise, sınıflandırma performansı o kadar iyidir.

$$\text{F1 Skor} = \frac{2(\text{PKD}*\text{Duy})}{\text{PKD}+\text{Duy}} \tag{53}$$

Matthews Korelasyon Katsayısı (MCC) : Makine öğreniminde ikili sınıflandırmaların kalite ölçüsü olarak kullanılmaktadır. MCC, gözlenen ve kestirilen ikili sınıflandırmalar arasında bir korelasyon katsayısıdır. (-1) ile (+1) arasında bir değer döndürür. (-1) mükemmel negatif korelasyonu diğer bir deyiş ile tahmin ve gerçek değerler arasındaki toplam uyuşmazlığı temsil etmektedir. 0 rasgele dağılımı ifade eder. (+1) mükemmel bir korelasyonu yani tamamen doğru ikili sınıflandırıcıyı ifade eder.

$$\text{MCC} = \frac{\text{DP}*\text{DN}-\text{YP}*\text{YN}}{\sqrt{(\text{DP}+\text{YP})(\text{DP}+\text{YN})(\text{DN}+\text{YP})(\text{DN}+\text{YN})}} \tag{54}$$

VERİ ÖN İŞLEME VE MODEL KURMA

PubChem veri tabanından indirilen bioassay verileri için PaDEL yazılımı kullanılarak 2757 adet moleküler değişken hesaplanmıştır. Daha sonra, verilerdeki sıfır veya sıfıra yakın varyansa sahip değişkenler çıkarılmış ve değişken sayısı 1348'e indirgenmiştir. Oluşturulan verilerin her biri %80 eğitim ve % 20 test seti olarak iki kısma ayrılmıştır. Verilerin standartlaştırılması için eğitim setlerine z-skor dönüşümü uygulanmıştır. Test setleri ise eğitim setlerinin parametrelerine (yani ortalama ve standart sapmasına) göre standartlaştırılmıştır. DVM ve RF'de

parametre optimizasyonu için 10 kat çapraz geçerlilik kullanılmıştır. DSA algoritmasında 4 gizli katmana (birinci katman 1024 düğümden, ikinci katman 2048 düğümden, üçüncü katman 1500 düğümden ve dördüncü katman 128 düğümden oluşmaktadır) sahip model kurulmuştur. DSA'da oluşturulan modelin aşırı öğrenmesini engellemek için %20'lik seyreltme (dropout) oranı kullanılmıştır. DSA algoritması için model oluşturma adımları Python 3.7.3, DVM ve RF için ise model oluşturma adımları R 3.6.1 programları kullanılarak gerçekleştirilmiştir.

BULGULAR

Çalışmamızda 5 adet HTS verisi DSA, DVM ve RF algoritmaları kullanılarak eğitilmiş, her bir algoritmanın performansı aynı test seti üzerinde test edilmiştir. Algoritmaların performansları dengeli doğruluk oranı, duyarlılık, pozitif kestirim değeri, F1 skoru ve MCC ölçüleri kullanılarak karşılaştırılmıştır.

AID652178 verisi için DSA, DVM ve RF algoritmalarının performans ölçüleri hesaplanmış ve sonuçlar Tablo 4'te özetlenmiştir. Bu veri seti dengesiz bir yapıda olup inaktif molekül sayısı aktif molekül sayısının yaklaşık 5 katıdır. Dengeli doğruluk oranının en yüksek olduğu algoritma DSA (0,767) iken DVM'de ve RF'de dengeli doğruluk oranı DSA'ya göre düşük bulunmuştur (sırasıyla, 0,526 ve 0,540). Duyarlılık ölçüsü açısından incelendiğinde, DSA algoritması en yüksek duyarlılığa sahip algoritma iken (0,686) DVM ve RF algoritmalarında duyarlılık oldukça düşük çıkmıştır (sırasıyla 0,057 ve 0,086). Pozitif kestirim değeri açısından RF (0,750) ve DVM (0,667) algoritmalarının DSA'ya göre (0,471) daha yüksek performans gösterdiği görülmüştür. F1 skoru açısından en yüksek performansı DSA (0,558) gösterirken DVM ve RF algoritmalarının F1 skor değerleri DSA'ya göre oldukça düşük çıkmıştır (sırasıyla 0,105 ve 0,154). Benzer şekilde MCC açısından incelendiğinde DSA'nın DVM ve RF'ye göre daha iyi performans gösterdiği görülmektedir (sırasıyla 0,464, 0,162 ve 0,219). Elde edilen sonuçlara göre; dengeli doğruluk oranı, duyarlılık, F1 skoru ve MCC açısından DSA algoritması DVM ve RF'ye göre daha başarılı bir performans göstermiştir.

Tablo 4. AID652178 verisi için DSA, DVM ve RF algoritmalarının performans ölçüleri

AID652178	DSA	DVM	RF
Dengeli Doğruluk Oranı	0,767	0,526	0,540
Duyarlılık	0,686	0,057	0,086
Pozitif Kestirim Değeri	0,471	0,667	0,750
F1 Skor	0,558	0,105	0,154
MCC	0,464	0,162	0,219

AID1053187 verisi için DSA, DVM ve RF algoritmalarının performans ölçüleri hesaplanmış ve sonuçlar Tablo 5'de özetlenmiştir. Bu veri seti de dengesiz bir yapıda olup inaktif molekül sayısı aktif molekül sayısının yaklaşık 3 katıdır. Elde edilen sonuçlara göre tüm performans ölçüleri açısından DSA algoritması DVM ve RF'ye göre daha iyi performans göstermiştir. Dengeli doğruluk oranının en yüksek olduğu algoritma DSA (0,865) iken DVM'de ve RF'de dengeli doğruluk oranı DSA'ya göre düşük bulunmuştur (sırasıyla 0,556 ve 0,765). Duyarlılık ölçüsü açısından incelendiğinde DSA algoritması en yüksek duyarlılığa sahip iken (0,809) RF algoritması (0,619) olup DVM algoritmasının duyarlılığı oldukça düşük çıkmıştır (0,155). Pozitif kestirim değeri açısından DSA (0,782) algoritmasının DVM (0,565) ve RF (0,712) algoritmalarına göre daha yüksek performans gösterdiği görülmüştür. F1 skoru açısından en yüksek performansı DSA (0,795) gösterirken DVM ve RF algoritmalarının F1 skor değerleri DSA'ya göre düşük çıkmıştır (sırasıyla 0,243 ve 0,663). Benzer şekilde MCC açısından incelendiğinde DSA'nın DVM ve RF'ye göre daha iyi performans gösterdiği görülmektedir (sırasıyla 0,721, 0,191 ve 0,555).

Tablo 5. AID1053187 verisi için DSA, DVM ve RF algoritmalarının performans ölçüleri

AID1053187	DSA	DVM	RF
Dengeli Doğruluk Oranı	0,865	0,556	0,765
Duyarlılık	0,809	0,155	0,619
Pozitif Kestirim Değeri	0,782	0,565	0,712
F1 Skor	0,795	0,243	0,663
MCC	0,721	0,191	0,555

AID1053196 verisi için DSA, DVM ve RF algoritmalarının performans ölçüleri hesaplanmış ve sonuçlar Tablo 6'da özetlenmiştir. Bu veri seti çalışmada kullanılan en dengesiz veri setidir. Bu veri setinde inaktif molekül sayısı aktif molekül sayısının yaklaşık 9 katıdır. Elde edilen sonuçlara göre tüm performans ölçüleri açısından DSA algoritması DVM ve RF'ye göre daha iyi performans göstermiştir. Dengeli doğruluk oranının en yüksek olduğu algoritma DSA (0,764) iken DVM ve RF dengeli doğruluk oranı birbirine eşit olup DSA'ya göre düşük bulunmuştur (0,544). Duyarlılık ölçüsü açısından incelendiğinde DSA algoritması en yüksek duyarlılığa sahip iken (0,630) DVM ve RF algoritmalarında duyarlılık birbirine eşit olup oldukça düşük çıkmıştır (0,087). Pozitif kestirim değeri açısından RF(1) ve DVM (1) algoritmalarının DSA'ya göre (0,409) daha yüksek performans gösterdiği görülmüştür. F1 skoru açısından en yüksek performansı DSA (0,496) gösterirken DVM ve RF algoritmalarının F1 skor değerleri birbirine eşit olup DSA'ya göre oldukça düşük çıkmıştır (0,16). Benzer şekilde MCC açısından incelendiğinde DSA'nın DVM ve RF'ye göre daha iyi performans gösterdiği görülmektedir (sırasıyla 0,439, 0,281 ve 0,281).

Tablo 6. AID1053196 verisi için DSA, DVM ve RF algoritmalarının performans ölçüleri

AID1053196	DSA	DVM	RF
Dengeli Doğruluk Oranı	0,764	0,544	0,544
Duyarlılık	0,630	0,087	0,087
Pozitif Kestirim Değeri	0,409	1,000	1,000
F1 Skor	0,496	0,160	0,160
MCC	0,439	0,281	0,281

AID1159608 verisi için DSA, DVM ve RF algoritmalarının performans ölçüleri hesaplanmış ve sonuçlar Tablo 7'de özetlenmiştir. Bu veri seti çalışmamızda kullanılan tek dengeli yapıdaki veri setidir ve aktif ve inaktif molekül sayıları yaklaşık olarak birbirine eşittir. Elde edilen sonuçlara göre tüm performans ölçüleri açısından DSA algoritması DVM ve RF'ye göre daha iyi performans göstermiştir. Dengeli doğruluk oranının en yüksek olduğu algoritma DSA (0,849) iken DVM'de ve RF'de dengeli doğruluk oranı DSA'ya göre düşük bulunmuştur (sırasıyla 0,625 ve 0,645). Duyarlılık ölçüsü açısından incelendiğinde DSA algoritması en yüksek duyarlılığa sahip iken (0,823) DVM ve RF algoritmalarında duyarlılık DSA'ya göre düşük çıkmıştır (sırasıyla 0,565 ve 0,621). Pozitif kestirim değeri açısından DSA (0,864) algoritmasının DVM (0,637) ve RF (0,647) algoritmalarına göre daha yüksek performans gösterdiği görülmüştür. F1 skoru açısından en yüksek performansı DSA (0,843) gösterirken DVM ve RF algoritmalarının F1 skor değerleri DSA'ya göre düşük çıkmıştır (sırasıyla 0,598 ve 0,634). Benzer şekilde MCC açısından incelendiğinde DSA'nın DVM ve RF'ye göre daha iyi performans gösterdiği görülmektedir (sırasıyla 0,698, 0,252 ve 0,291).

Tablo 7. AID1159608 verisi için DSA, DVM ve RF algoritmalarının performans ölçüleri

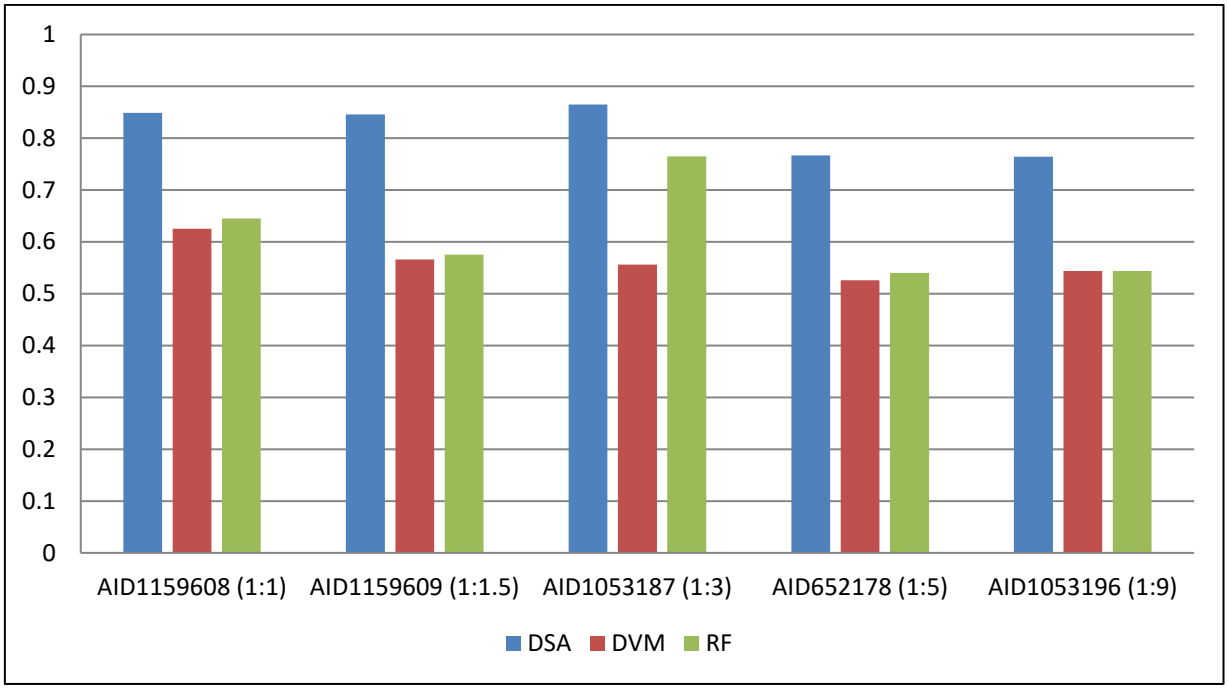
AID1159608	DSA	DVM	RF
Dengeli Doğruluk Oranı	0,849	0,625	0,645
Duyarlılık	0,823	0,565	0,621
Pozitif Kestirim Değeri	0,864	0,637	0,647
F1 Skor	0,843	0,598	0,634
MCC	0,698	0,252	0,291

AID1159609 verisi için DSA, DVM ve RF algoritmalarının performans ölçüleri hesaplanmış ve sonuçlar Tablo 8'de özetlenmiştir. Bu veri seti çalışmamızda kullanılan dengesiz veri setleri arasında dengesizlik oranı en düşük veri setidir. Bu veri setinde inaktif molekül sayısı aktif molekül sayısının yaklaşık 1,5 katıdır. Elde edilen sonuçlara göre tüm performans ölçüleri açısından DSA algoritması DVM ve RF'ye göre daha iyi performans göstermiştir. Dengeli doğruluk oranının en yüksek olduğu algoritma DSA (0,846) iken DVM'de ve RF'de dengeli doğruluk oranı DSA'ya göre düşük bulunmuştur (sırasıyla 0,566 ve 0,575). Duyarlılık ölçüsü açısından incelendiğinde DSA algoritması en yüksek duyarlılığa sahip iken (0,805) DVM ve RF algoritmalarında duyarlılık DSA'ya göre düşük çıkmıştır (sırasıyla 0,259 ve 0,266). Pozitif kestirim değeri açısından DSA (0,827) algoritmasının DVM (0,578) ve RF'ye göre (0,603) daha yüksek performans gösterdiği görülmüştür. F1 skoru açısından en yüksek performansı DSA (0,816) gösterirken DVM ve RF algoritmalarının F1 skor değerleri DSA'ya göre düşük çıkmıştır(sırasıyla 0,358 ve 0,369). Benzer şekilde MCC açısından incelendiğinde DSA'nın DVM ve RF'ye göre daha iyi performans gösterdiği görülmektedir (sırasıyla 0,696, 0,169 ve 0,192).

Tablo 8. AID1159609 verisi için DSA, DVM ve RF algoritmalarının performans ölçüleri

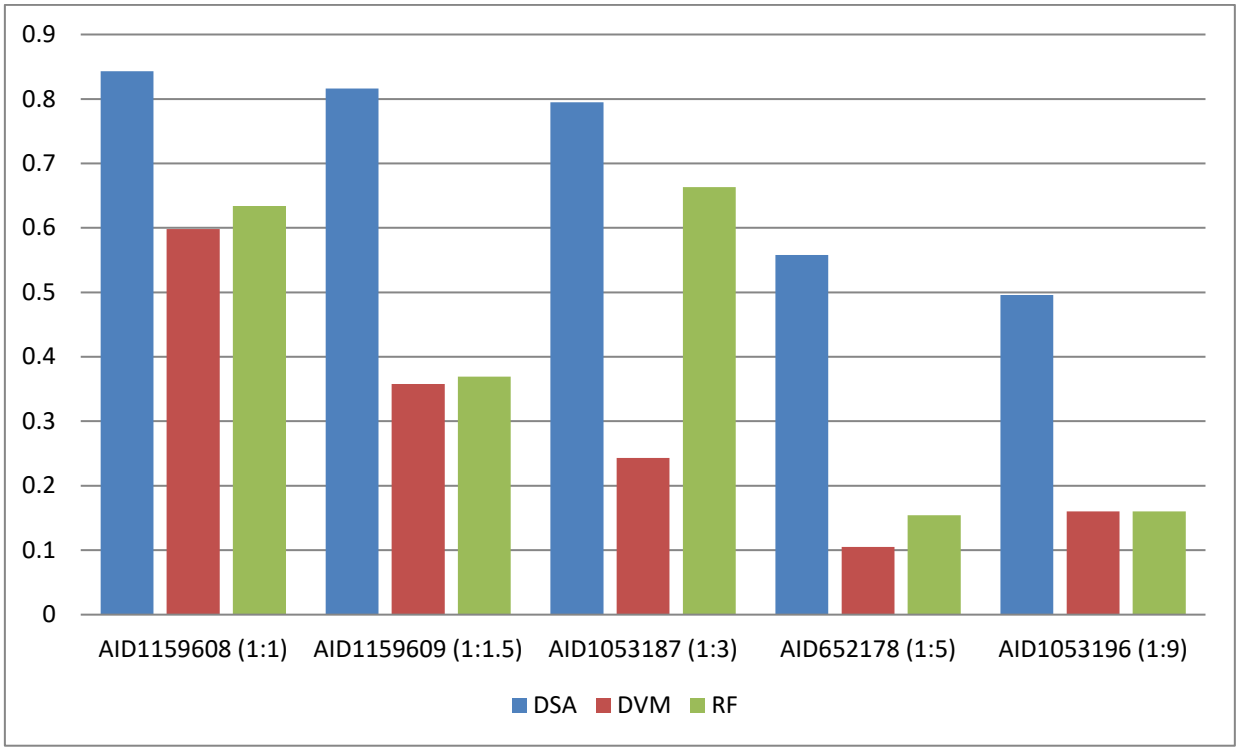
AID1159609	DSA	DVM	RF
Dengeli Doğruluk Oranı	0,846	0,566	0,575
Duyarlılık	0,805	0,259	0,266
Pozitif Kestirim Değeri	0,827	0,578	0,603
F1 Skor	0,816	0,358	0,369
MCC	0,696	0,169	0,192

Algoritmaların performansları veri setlerinin dengesizlik yapıları göz önüne alınarak karşılaştırılmıştır. Dengeli doğruluk oranı açısından incelendiğinde; DSA algoritmasının tüm dengesizlik yapılarında en iyi performansı gösteren algoritma olduğu görülmektedir. DVM ve RF algoritmalarının performansları AID1053187 (1:3) dışında benzer bulunmuştur. DSA algoritması tüm dengesiz veri yapılarında en iyi performansı göstermekle birlikte, dengesizlik oranı arttıkça performansında düşüş olduğu gözlenmektedir. Dengeli doğruluk oranına ilişkin elde edilen sonuçlar Şekil 5'te verilmiştir.



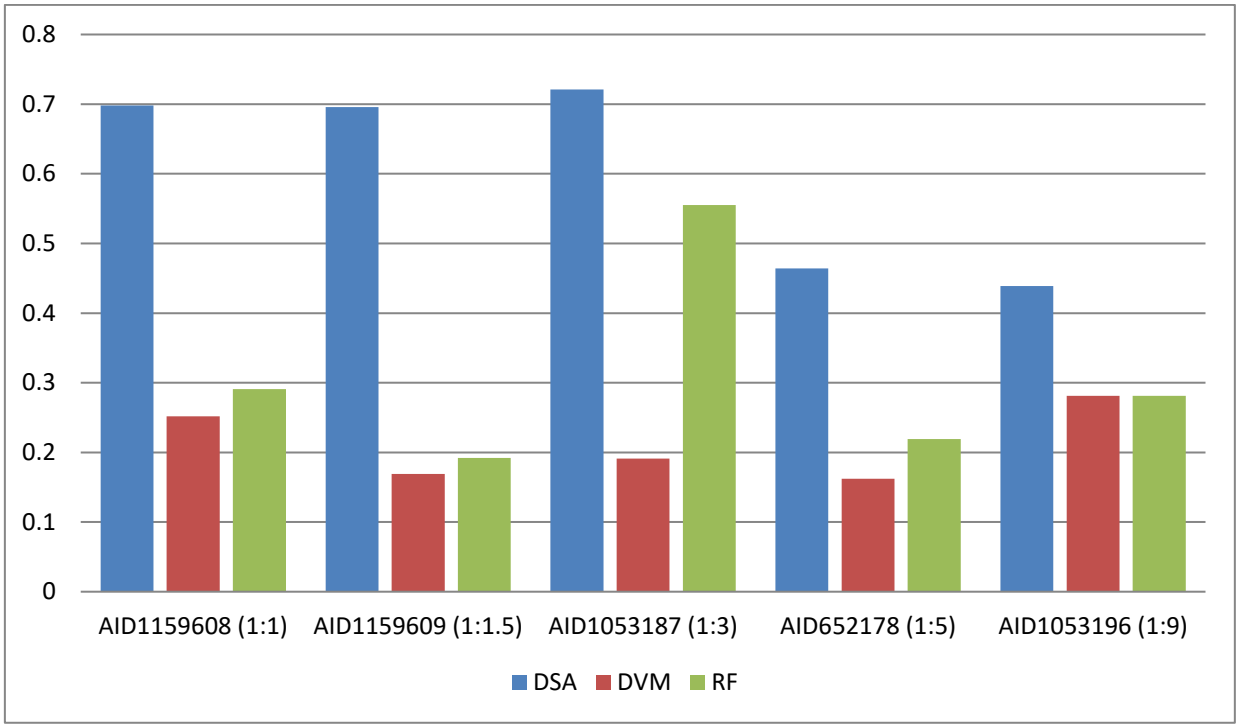
Şekil 5. Dengeli doğruluk oranı açısından DSA, DVM ve RF performanslarının dengesizlik oranlarına göre karşılaştırılması

F1 skoru açısından incelendiğinde; DSA algoritmasının tüm dengesizlik yapılarında en iyi performansı gösteren algoritma olduğu görülmektedir. DVM ve RF algoritmalarının performansları AID1053187 (1:3) dışında benzer bulunmuştur. DSA algoritması tüm dengesiz veri yapılarında en iyi performansı göstermekle birlikte, dengesizlik oranı arttıkça performansında düşüş olduğu gözlenmektedir. F1 skoruna ilişkin elde edilen sonuçlar Şekil 6'da verilmiştir.



Şekil 6. F1 skoru açısından DSA, DVM ve RF performanslarının dengesizlik oranlarına göre karşılaştırılması

MCC açısından incelendiğinde; DSA algoritmasının tüm dengesizlik yapılarında en iyi performansı gösteren algoritma olduğu görülmektedir. DVM ve RF algoritmalarının performansları AID1053187 (1:3) dışında benzer bulunmuştur. DSA algoritması tüm dengesiz veri yapılarında en iyi performansı göstermekle birlikte, dengesizlik oranı arttıkça performansında düşüş olduğu gözlenmektedir. MCC'ye ilişkin elde edilen sonuçlar Şekil 7'de verilmiştir.



Şekil 7. MCC açısından DSA, DVM ve RF performanslarının dengesizlik oranlarına göre karşılaştırılması

TARTIŞMA

İlaç geliştirme çalışmalarının erken evresinde taranması gereken binlerce molekül bulunmaktadır. Bu moleküller arasından aktif olanlarının hızlı ve doğru bir şekilde tespit edilmesi ilaç geliştirme çalışmalarının maliyetini ve bu işe harcanan zamanı anlamlı derecede düşürecektir. Bu amaçla makine öğrenimi yöntemleri ilaç geliştirme çalışmalarının erken evresinde aktif ve inaktif molekülleri hızlı ve doğru bir şekilde ayırmak için kullanılabilir. Günümüzde HTS deneyleriyle elde edilen yüksek boyutlu veriler PubChem veri tabanına yüklenmekte ve araştırmacıların hizmetine sunulmaktadır. Bu veri tabanında bulunan yüksek boyutlu veriler kullanılarak makine öğrenimi algoritmaları eğitilebilir ve molekülleri yüksek bir doğruluk oranı ile aktif ve inaktif olarak ayırabilecek modeller oluşturulabilir. PubChem veri tabanında yer alan HTS verilerinin en önemli dezavantajı veri setlerinin dengesiz bir yapıda olmalarıdır. Bu dengesiz veri yapısı literatürde sıklıkla kullanılan makine öğrenimi algoritmalarının performansını olumsuz yönde etkileyen en önemli faktörlerden biridir. Son yıllarda özellikle veri boyutunun artmasıyla birlikte DSA algoritmasının kullanım alanı genişlemiş ve birçok alanda oldukça iyi performanslar elde edilmiştir. Çalışmamızda farklı derecelerde dengesiz veri yapısına sahip olan 5 adet HTS verisi kullanılarak DSA algoritması ile eğitilmiş ve performansı test edilmiştir. Daha sonra DSA algoritmasının performansı literatürde sıklıkla kullanılan DVM ve RF algoritmaları ile karşılaştırılmıştır. Bu algoritmaların karşılaştırılmasında dengeli doğruluk oranı, duyarlılık, pozitif kestirim değeri, F1 skor, MCC kriterleri göz önüne alınmıştır.

Bu performans ölçüleri açısından değerlendirildiğinde, DSA algoritmasının DVM ve RF'ye göre tüm ölçüler açısından daha iyi performans gösterdiği gözlenmiştir.

İlaç geliştirme çalışmalarının erken evresinde aktif molekülerin tespit edilmesine yönelik literatürde çok sayıda çalışma bulunmaktadır. Bu çalışmalarda farklı makine öğrenimi yöntemleri kullanılmıştır.

Sadowski ve Kubinyi (1998) ilaç ve ilaç olmayan moleküllerin bulunduğu veri tabanlarından (Mevcut Kimyasallar Dizininden (Available Chemicals Directory, ACD) 169.331 ilaç olmayan molekül, Dünya İlaç İndeksi'nden (World Drug Index, WDI) 38.416 ilaç molekülü) yararlanarak hızlı bir şekilde sınıflandırma yapabilmek için DSA algoritmasını kullanarak puanlama şeması geliştirmişlerdir. Bu çalışma sonucunda ACD dizininin doğruluk oranı %83 (ilaç olmayan molekül), WDI dizininin doğruluk oranı ise %77 (ilaç molekülü) olarak belirlenmiştir (6).

Byvatov ve ark., (2003) erken evre sanal bileşik filtreleme ve taramada ikili karar problemlerine örnek olarak ilaç/ilaç olmayan molekül sınıflandırmasında 4998 ilaç ve 4210 ilaç olmayan molekül kullanarak DVM ve YSA algoritmalarının performanslarını karşılaştırmışlardır. Çalışma sonucunda, DVM'nin %82 doğruluk oranı ve 0.63 MCC değerine sahip olduğu, YSA'nın ise %80 doğruluk oranı ve 0.58 MCC değerine sahip olduğu bulunmuştur (7). Böylelikle, DVM'nin YSA'ya göre ilaç sınıflandırma performansını arttırdığı ortaya konmuştur.

Diğer bir çalışmada, Zernov ve ark., (2003) 15000 ilaç ve 15000 ilaç olmayan molekülden yararlanarak moleküllerin aktivitelerini DVM ve DSA algoritmalarını kullanarak sınıflandırmışlardır. Sonuç olarak, DVM'nin %75.15'lik doğruluk oranı ile farklı DSA modellerinden (çok katmanlı algılayıcı: %72.52, modüler ileri beslemeli ağ %70.92, genelleştirilmiş ileri beslemeli ağ %69.85) daha iyi performans gösterdiği görülmüştür (8).

Fang ve arkadaşları Alzheimer hastalığı tedavisi için önemli bir farmakolojik hedef olan BuChE inhibitörlerini inhibitör olmayanlardan ayırt etmek için DVM ve Naive Bayes modelleri ile 1870 yapısal tanımlayıcıdan oluşan veri seti kullanmıştır.
En iyi iki modelin test seti için MCC değerleri 0.9551 ve 0.9550 olarak bulunmuştur. Çalışma ligandların biyoaktivitelerini tahmin etmek ve öncü bileşikleri keşfetmek için DVM algoritmasının uygulanabilirliğini kanıtlamıştır (9).

Korkmaz ve ark., (2014) çeşitli değişken seçim yöntemleri ile DVM algoritmasını kullanarak ilaç ve ilaç olmayan bileşikler arasında ayrım yapmayı amaçlamışlardır. Çalışmalarında eğitim seti için 311 ilaç ve 320 ilaç olmayan molekül, test seti için ise 98 ilaç ve 118 ilaç olmayan molekül kullanmışlardır. Sonuç olarak, test seti için doğruluk oranı %76-%81, duyarlılık %87-%89, pozitif kestirim değeri %67-%74, F1 skor %77-%80, MCC %55-%64 olarak elde edilmiştir (3).

Korkmaz ve ark., (2015) eğitim seti 631 bileşik (311 ilaç ve 320 ilaç dışı molekül), test seti için 216 bileşikten (98 ilaç ve 118 ilaç dışı molekül) oluşan veri seti ile 23 adet makine öğrenimi yönteminin (diskriminant sınıflandırıcıları, karar ağacı sınıflandırıcıları, çekirdek tabanlı sınıflandırıcılar, topluluk sınıflandırıcıları ve diğer sınıflandırıcılar) performansını karşılaştırmıştır. Sonuç olarak, doğruluk oranı %68-79, duyarlılık %81- 92, pozitif tahmin değeri % 60-72, F1 skoru %72-79, MCC %42-59, dengeli doğruluk oranı % 70-79 bulunmuş olup ilaç moleküllerini sınıflandırmak için web tabanlı bir uygulama geliştirmişlerdir (4).

İlaç geliştirme çalışmalarının erken evresinde kimyasal moleküllerin aktivitelerinin sınıflandırılmasının (aktif-inaktif) yanı sıra bazı çalışmalarda ilaç molekülleri için aktivite kestiriminde bulunulmuş ve ilaç molekülleri aktivitelerine göre sıralanmışlardır.

Jorissen ve Gilson (2005), gerçekleştirdikleri çalışmada, istenen aktiviteye sahip molekülleri bulmak için yapılan sanal tarama işleminde DVM algoritması ile her biri farklı bir proteini hedef alan 250 aktif ve 250 inaktif molekül ve 50 değişkenden oluşan dengeli bir veri seti kullanmışlardır. Bu çalışmada DVM algoritması kullanılarak moleküller aktivitelerine göre başarıyla sınıflandırılmıştır (12).

İlaç moleküllerinin aktivite sıralaması için bir diğer çalışma Rathke ve ark., (2010) tarafından gerçekleştirilmiştir. Bu çalışmada, BZR (benzodiazepine receptor), COX-2 (cyclooxygenase-2) ve DHFR (dihydrofolate reductase) reseptörlerine karşı afinite gösterebilecek moleküller taranmış ve DVM algoritması kullanılarak aktivitelerine göre sıralanmışlardır (15). Ma ve ark.,(2015) QSAR tahmin edilmesi çalışmasında 15 QSAR veri setindeki çeşitli boyutlarda (2000-50000) molekül ile DSA modelini kullanmışlardır. DSA'nın uygulamalı bir QSAR yöntemi olarak kullanılabileceğini ve birçok durumda RF'den daha iyi performans gösterdiğini bulmuşlardır (16).

Mayr ve ark., (2016) derin öğrenmeyi kullanarak gerçekleştirdikleri toksisite tahmini çalışmasında 12.707 kimyasal bileşik (11.764 bileşik eğitim veri seti, 296 bileşik liderlik seti, 647 bileşik test seti) içeren veri setini kullanmışlar ve DSA'nın tüm rakip yöntemlere kıyasla sürekli olarak çok yüksek performans gösterdiğini belirtmişlerdir (17).

Çok görevli öğrenme (multi-task learning) yönteminin sanal taramaya uygulandığı bir çalışma yürüten Ramsundar ve ark.,(2015), topladıkları 259 adet veri setini (bu veri setleri dört gruba ayrılmıştır: PCBA, MUV, DUD-E ve Tox21) DSA algoritması ile eğitmişlerdir ve çok görevli öğrenmenin ilaç molekülü sınıflandırmada performansı arttırdığını göstermişlerdir (18).

DSA modelinin hiper-parametrelerinin optimizasyonunu araştıran Koutsoukas ve ark.,(2017) DSA modelinin performansını SVM, RF, NB ve kNN algoritmalarıyla karşılaştırmada yedi farklı biyoaktivite sınıfı (ChEMBL205, CHEMBL301, CHEMBL240, CHEMBL219, CHEMBL244, CHEMBL218, CHEMBL1978) kullanmışlardır. Çalışmadan elde edilen MCC değerleri açısından incelendiğinde; DSA'nın NB'den 0,149, kNN'den 0,092, doğrusal çekirdeğe sahip SVM'den 0,052, RF'den 0,021 ve radyal tabanlı çekirdek fonksiyona sahip SVM'den 0,009 daha yüksek olduğunu belirtmişlerdir (19).

Lenselink ve ark.,(2017) 13,488,513 veri içeren CHEMBL biyoaktivite veri setinden yararlanarak DSA algoritmasının performansını NB, RF, SVM ve lojistik regresyon ile karşılaştırmışlardır. Bu çalışmanın sonucunda DSA algoritmasının NB, RF, SVM ve lojistik regresyondan daha yüksek performans sergilediği görülmüştür (20).

Bu çalışmada, PubChem veri tabanı aracılığı ile elde edilen farklı derecelerdeki dengesiz veri yapısına sahip olan 5 adet HTS verisi DSA algoritması ile eğitilmiş ve performansı test edilmiştir. Daha sonra, DSA algoritmasının performansı literatürde sıklıkla kullanılan DVM ve RF algoritmaları ile karşılaştırılmıştır. Algoritmaların performansları dengeli doğruluk oranı, duyarlılık, pozitif kestirim değeri, F1 skor, MCC ölçüleri kullanılarak karşılaştırılmıştır. Çalışmamızda DSA için dengeli doğruluk oranı 0,764 ile 0,865 arasında bulunurken DVM'de ve RF'de ise dengeli doğruluk oranı sırasıyla 0,526 – 0,625 ve 0,540 – 0,765 arasında bulunmuştur. Duyarlılık ölçüsü DSA için 0,630 – 0,823 arasında bulunurken DVM'de ve RF'de ise duyarlılık sırasıyla 0,057 – 0,565 ve 0,086 – 0,619 arasında bulunmuştur. Pozitif kestirim değeri DSA için 0,409 ile 0,864 arasında bulunurken

DVM'de ve RF'de ise pozitif kestirim değeri sırasıyla 0,565 – 1,000, 0,603 – 1,000 arasında bulunmuştur. F1 skoru DSA için 0,496 ile 0,843 arasında bulunurken DVM'de ve RF'de ise F1 skor sırasıyla 0,160 – 0,598, 0,160 – 0,663 arasında bulunmuştur. MCC DSA için 0,439 ile 0,721 arasında bulunurken DVM'de ve RF'de ise MCC sırasıyla 0,162 – 0,281, 0,192 – 0,555 arasında bulunmuştur. Çalışmadan elde edilen bulgular incelendiğinde; pozitif kestirim değeri dışındaki tüm performans ölçüleri açısından DSA algoritmasının DVM ve RF'ye göre daha iyi performans gösterdiği görülmektedir. Özellikle dengesiz veri setlerinde performans değerlendirmesinde kullanılan en önemli performans ölçülerinden olan F1 skor ve MCC açısından DSA algoritmasının DVM ve RF algoritmalarına göre daha iyi performanslar gösterdiği gözlenmiştir. Algoritmaların performansları verilerin sınıf dengesizlik yapıları göz önüne alınarak değerlendirildiğinde, DSA algoritmasının dengeli doğruluk oranı, F1 skor ve MCC açısından DVM ve RF'ye göre daha iyi performans gösterdiği görülmektedir. Bununla birlikte, sınıflar arasındaki dengesizlik durumu arttıkça DSA performansının azaldığı gözlenmiştir.

Literatürde gerçekleştirilen çalışmaların çoğunda kullanılan veri setleri dengeli yapıdadır. Ancak, gerçekte ilaç geliştirme çalışmalarında kullanılan verilerin büyük bir bölümü dengesiz yapıdadır. Literatürde standart olarak kullanılan makine öğrenimi algoritmalarının birçoğu dengesiz veri yapısında kötü sonuçlar vermektedir. Bu nedenle, bu algoritmalar sınıflandırma amacıyla kullanılmadan önce dengeli yapıdaki veri setleri oluşturulmaktadır. Bununla birlikte, çalışmamızdan elde edilen sonuçlar göstermektedir ki; dengesiz veri yapılarında DSA algoritması DVM ve RF'den daha iyi performans göstermektedir.

SONUÇLAR

İlaç geliştirme çalışmaları zorlu, maliyetli, zaman alıcı çalışmalardır. Akılcı ilaç tasarımı ile birlikte ilaç geliştirme çalışmalarındaki zamanı ve maliyeti azaltmak için HTS yöntemi kullanılmaya başlanmıştır. HTS yöntemi ile elde edilen bioassay verileri PubChem veri tabanında depolanmaktadır. Böylece PubChem veri tabanındaki veriler kullanılarak makine öğrenimi yöntemleri eğitilebilir ve aktif moleküllerin tespiti için performansı yüksek modeller oluşturulabilir. Makine öğrenimi yöntemleri ilaç geliştirme çalışmalarında uzun süredir kullanılmalarına rağmen bu yöntemleri eğitmek için kullanılan veri setleri genellikle dengeli yapıdadır. Ancak PubChem veri tabanında bulunan gerçek veri setleri dengesiz yapıdadır. Bu durum klasik makine öğrenimi yöntemlerinin performansını olumsuz yönde etkilemektedir. Son yıllarda DSA algoritması birçok alanda oldukça iyi performanslar göstermiş ve özellikle büyük boyutlu verilerin sınıflandırılmasında sıklıkla kullanılmıştır.

Çalışmamızda dengesiz veri yapılarında DSA algoritmasının performansını literatürde sıklıkla kullanılan DVM ve RF algoritmaları ile karşılaştırarak DSA algoritmasının bu tür dengesiz veri yapılarında daha iyi performans gösterip göstermediği ortaya konmaya çalışılmıştır.

Bu amaçla, PubChem veri tabanından alınan 5 adet HTS verisi DSA, DVM ve RF algoritmaları kullanılarak eğitilmiş ve dengeli doğruluk oranı, duyarlılık, pozitif kestirim değeri, F1 skor ve MCC ölçüleri göz önüne alınarak bu üç algoritmanın performansları karşılaştırılmıştır. Farklı derecelerdeki beş dengesiz veri yapısında DSA algoritmasının DVM ve RF algoritmalarından daha iyi performans gösterdiği görülmüştür.

Bu çalışmanın sonuçları, PubChem veri tabanının makine öğrenimi modellerinin eğitilmesi için iyi bir kaynak olduğunu ve DSA'nın, ilaç keşif çalışmalarındaki maliyet ve zamanı azaltmak için kullanılabilecek iyi bir makine öğrenme yöntemi olduğunu göstermektedir.

ÖZET

İlaç geliştirme çalışmalarının erken evresinde binlerce molekül arasından aktivite gösteren moleküller tespit edilerek ilaç geliştirme çalışmalarına harcanan süre ve maliyet azaltılmaya çalışılmaktadır. Bu amaçla yüksek verimli tarama deneyleri yapılarak moleküller aktif ve inaktif olarak sınıflandırılmaktadır. Bu deneylerden elde edilen veriler PubChem veri tabanına yüklenmektedir. Bu veri tabanındaki veriler kullanılarak makine öğrenimi algoritmaları yardımıyla sınıflandırma modelleri geliştirilebilir, böylece aktivite gösteren moleküller daha hızlı ve daha ucuz bir şekilde tespit edilebilir.

Bu çalışmada PubChem veri tabanından elde edilen farklı derecelerde dengesizlik yapısına sahip 5 adet veri seti derin sinir ağları (DSA) algoritmasıyla eğitilmiştir. Eğitilen DSA algoritmasının performansı literatürde sıklıkla kullanılan destek vektör makineleri (DVM) ve random forest (RF) algoritmalarıyla karşılaştırılmıştır. Algoritmaların performans karşılaştırmasında dengeli doğruluk oranı, duyarlılık, pozitif kestirim değeri, F1 skor, MCC ölçütleri göz önüne alınmıştır. Bu ölçütler değerlendirildiğinde, pozitif kestirim değeri dışındaki diğer ölçütler açısından, özellikle dengesiz veri setlerinde performans değerlendirmesinde en önemli ölçütlerden olan F1 skor ve MCC açısından, DSA algoritmasının DVM ve RF algoritmalarına göre daha yüksek performans gösterdiği görülmüştür.

Sonuç olarak, DSA algoritması dengesiz veri yapılarında diğer makine öğrenimi algoritmalarına göre daha iyi bir performans gösterdiği için ilaç geliştirme çalışmalarına harcanan süreyi ve maliyeti azaltmada tercih edilebilecek iyi bir makine öğrenimi algoritmasıdır.

Anahtar Kelimeler: PubChem, derin öğrenme, dengesiz veri, destek vektör makineleri, random forest, sanal tarama.

ACTIVITY CLASSIFICATION OF DRUG MOLECULES USING DEEP LEARNING

SUMMARY

In the early stages of drug development studies, molecules that are active among thousands of molecules are identified and the time and cost spent on drug development studies are tried to be reduced. For this purpose, molecules are classified as active and inactive by performing high-throughput screening experiments. The data obtained from these experiments are uploaded to PubChem database. By using the data in this database, classification models can be developed with the help of machine learning algorithms, so that the molecules showing activity can be detected faster and cheaper.

In this study, 5 data sets with different degree of imbalance structure obtained from PubChem database were trained with deep neural network (DSA) algorithm. The performance of the trained DSA algorithm was compared with the support vector machines (DVM) and random forest (RF) algorithms that are frequently used in the literature. Balanced accuracy, sensitivity, positive predictive value, F1 score and MCC criteria were taken into consideration in the performance comparison of the algorithms.

When these criteria were evaluated, it was observed that DSA algorithm performed better than DVM and RF algorithms in terms of F1 score and MCC which is one of the most important criteria in performance evaluation especially in unbalanced data sets in terms of other criteria except positive predictive value.

As a result, DSA algorithm is a good machine learning algorithm that can be preferred in reducing time and cost spent on drug development studies because it performs better in unbalanced data structures than other machine learning algorithms.

Key Words: PubChem, deep learning, unbalanced data, support vector machines, random forest, virtual screning.

KAYNAKLAR

1. Aydın O. Hidrazon Türevi Mao İnhibitörleri İle Akılcı İlaç Tasarımına Yönelik Qsar Analizi (tez). İstanbul: Marmara Üniversitesi Fen Bilimleri Ensitüsü; 2011.

2. Ratti E, Tristi D. The continuing evolution of the drug discovery process in the pharmaceutical industry. Farmaco 2001;56:13-9.

3. Korkmaz S, Zararsız G, Goksuluk D. Drug/nondrug classification using support vector machines with various feature selection strategies. Comput Methods Programs Biomed 2014;117:51-60.

4. Korkmaz S, Zararsız G, Goksuluk D. MLViS: a web tool for machine learning-based virtual screening in early-phase of drug discovery and development. Plos One 2015;10(4): e0124600.

5. Broach JR, Thorner J. High-throughput screening for drug discovery. Nature 1996;384(7):14-6.

6. Sadowski J, Kubinyi H. A scoring scheme for discriminating between drugs and nondrugs. J Med Chem Sci 1998;41:3325-29.

7. Byvatov E, Fechner U, Sadowski J, Schneider G. Comparison of support vector machine and artificial neural network systems for drug/nondrug classification. J Chem Inf Comput Sci 2003;43:1882-89.

8. Zernov VV, Balakin KV, Ivaschenko AA, Savchuk NP, Pletnev IV. Drug discovery using support vector machines. The case studies of druglikeness, agrochemical-likeness, and enzyme inhibition predictions. J Chem Inf Comput Sci 2003;43:2048-56.

9. Fang J, Yang R, Gao L, Zhou D,Yang S, Liu A,Du G. Predictions of buche inhibitors using support vector machine and naive bayesian classification techniques in drug discovery. J Chem Inf Model 2013;53:3009-20.

10. Ehrman TM, Barlow DJ, Hylands PJ. Virtual screening of Chinese herbs with random forest. J Chem Inf Model 2007;47:264-78.

11. Gertrudes JC, Maltarollo VG, Silva RA, Oliveira PR, Honorio KM, Silva ABF. Machine learning techniques and drug design. Curr Med Chem 2012;19:4289-97.

12. Jorissen RN, Gilson MK. Virtual screening of molecular databases using a support vector machine. J Chem Inf Model 2005;45:549-61.

13. Wassermann AM, Geppert H, Bajorath J. Searching for target-selective compounds using different combinations of multiclass support vector machine ranking methods, kernel functions, and fingerprint descriptors. J Chem Inf Model 2009;49:582-92.

14. Agarwal S, Dugar D, Sengupta S. Ranking chemical structures for drug discovery: A new machine learning approach. J Chem Inf Model 2010;50:716-31.

15. Rathke F, Hansen K, Brefeld U, Müller KR. StructRank: A new approach for ligand-based virtual screening. J Chem Inf Model 2010;51:83-92.

16. Ma J, Sheridan RP, Liaw A, Dahl GE, Svetnik V. Deep neural nets as a method for quantitative structure−activity relationships. J Chem Inf Model 2015;55:263-74.

17. Mayr A, Klambauer G, Unterthiner T, Hochreiter S. DeepTox: Toxicity prediction using deep learning. Front Environ Sci 2016;3:80.

18. Ramsundar B, Kearnes S, Riley P, Webster D, Konerding D, Pande V. Massively multitask networks for drug discovery. Arxiv 2015;1502.02072.

19. Koutsoukas A, Monaghan KJ, Li X, Huan J. Deep-learning: investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data. J Chem Inform Comput Sci 2017;9(1):42.

20. Lenselink EB, Dijke N, Bongers B, Papadatos G, Vlijmen HWT, Kowalczyk W ve ark. Beyond the hype: deep neural networks outperform established methods using a ChEMBL bioactivity benchmark set. J Chem Inform Comput Sci 2017;9(1):45.

21. Yap CW. Software news and update PaDEL-descriptor : an open source software to calculate molecular descriptors and fingerprints. J Comput Chem 2011;32:1466-74.

22. Rosenblatt F. The perceptron : a probabilistic model for information storage and organization in the brain. Psychol Rev 1958;65(6):386-408.

23. Küçük D, Arıcı N. Doğal dil işlemede derin öğrenme uygulamaları üzerine bir literatür çalışması. Uluslararası Yönetim Bilişim Sistemleri Ve Bilgisayar Bilimleri Dergisi 2018;2(2):76-86.

24. Alsugair AM, Al-qudrah AA. Artificial neural network approach for pavement maintenance. J Comput Civil Eng 1998;12(4):249-55.

25. Patterson J, Gibson A. Deep learning a practitioner's approach. 1st ed. California: O'Reilly, 2017.

26. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521:436-44.

27. Deng L, Yu D. Deep learning: methods and applications.1st ed. Hanover: Now Publishers Inc, 2014:3-148.

28. İnik Ö, Ülker E. Derin öğrenme ve görüntü analizinde kullanılan derin öğrenme modelleri. Gaziosmanpaşa Bilimsel Araştırma Dergisi 2017;6(3):85-104.

29. Nesterov YE. A method of solving a convex programming problem with convergence rate O ($1/k^2$). Soviet Math Dokl 1983;27(2):372-6.

30. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 2014;15:1929-58.

31. Bircanoğlu C. Derin Gömme Yitim Fonksiyonlarının Karşılaştırılması (tez). İstanbul: Bahçeşehir Üniversitesi Fen Bilimleri Enstitüsü; 2017.

32. Erdoğmuş Ş, Ayhan S. Destek vektör makineleriyle sınıflandırma problemlerinin çözümü için çekirdek fonksiyonu seçimi. Eskişehir Osmangazi Üniversitesi İİBF Dergisi 2014;9(1):175-98.

33. Cortes C, Vapnik V. Support-vector networks. Machine Learning 1995;20(3):273-97.

34. Erpolat S, Öz E. Kanser verilerinin sınıflandırılmasında yapay sinir ağları ile destek vektör makinelerinin karşılaştırılması. İstanbul Aydın Üniversitesi Dergisi 2010;2:71-83.

35. Demirci D.A. Destek Vektör Makineleri İle Karakter Tanıma (tez). İstanbul: Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü; 2007.

36. Eray O. Destek Vektör Makineleri İle Ses Tanıma Uygulaması (tez). Denizli: Pamukkale Üniversitesi Fen Bilimleri Enstitüsü; 2008.

37. Alpaydın E. Introduction to machine learning. 1st ed. The MIT Press; 2014.

38. Ayhan S. Kaba Küme Ve Destek Vektör Makineleri Kullanılarak Nitelik İndirgeme Ve Sınıflandırma Problemlerinin Çözümü İçin Bütünleşik Bir Yaklaşım (tez). Eskişehir: Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü; 2013.

39. Akman M. Veri Madenciliğine Genel Bakış Ve Random Forests Yönteminin İncelenmesi: Sağlık Alanında Bir Uygulama (tez). Ankara: Ankara Üniversitesi Sağlık Bilimleri Enstitüsü; 2010.

40. Liu Y. Random forest algorithm in big data environment. Computer Modelling And New Technologies 2014;18(12):147-51.

41. Akman M, Genç Y, Ankaralı H. Random forests yöntemi ve sağlık alanında bir uygulama. Türkiye Klinikleri J Biostat 2011;3(1):36-48.

42. Abou-zleikha M, Shaker N. Evolving random forest for preference learning. In: Mora AM, Squillero G (Eds.) Applications of Evolutionary Computation:18th European Conference: 2015 April 8-10; Copenhagen, Denmark. Springer; 2015, p.318-20.

43. Çınaroğlu S. Farklı sayılarda ağaç türetildiğinde ve çapraz geçerlilikte “k” parametresi değiştirildiğinde random forest performans sonuçlarının incelenmesi. Türkiye Klinikleri J Biostat 2015;7(2):108-18.

44. Ayas S. Mikroskobik İmgelerde Tüberküloz Bakterisinin Rastgele Ormanlar Yöntemiyle Sınıflandırılması (tez). Trabzon: Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü; 2014.

45. Sacchi MD. A bootstrap procedure for high-resolution velocity analysis. Geophysics 1998;63(5):1716-25.

46. Korkem E. Mikroarray Gen Ekspresyon Veri Setlerinde Random Forest Ve Naive Bayes Sınıflama Yöntemleri Yaklaşımı (tez). Ankara: Hacettepe Üniversitesi Sağlık Bilimleri Ensitüsü; 2013.

47. Freund Y, Schapire RE. A short introduction to boosting. J Japanese Society for Artificial Intelligence 1999;14(5):771-80.

48. Yılmaz H. Random Forests Yönteminde Kayıp Veri Probleminin İncelenmesi Ve Sağlık Alanında Bir Uygulama (tez). Eskişehir: Eskişehir Osmangazi Üniversitesi Sağlık Bilimleri Enstitüsü; 2014.

49. Schapire RE. Theoretical views of boosting and applications. In: Watanabe O, Yokomori T (Eds.) 10th International Conference on Algoritmic Learning Theory: 1999 December 6-8; Tokyo, Japan. Springer; 1999, p.13-25.

50. Atasever ÜH. Uydu Görüntülerinin Sınıflandırılmasında Hızlandırma (Boosting), Destek Vektör Makineleri, Rastgele Orman (Random Forest) Ve Regresyon Ağaçları Yöntemlerinin Kullanılması (tez). Kayseri: Erciyes Üniversitesi Fen Bilimleri Enstitüsü; 2011.

ŞEKİLLER LİSTESİ

ŞEKİLLER

TABLOLAR

ÖZGEÇMİŞ

Adı Soyadı **:** Hatice Kanberiz

Doğum Yeri **:** Merkez / Edirne

Doğum Tarihi **:** 21/01/1993

Medeni Hali **:** Bekar

Yabancı Dili **:** İngilizce

Eğitim Durumu

Ön Lisans : Trakya Üniversitesi Edirne Teknik Bilimler Meslek Yüksekokulu Bilgisayar Programcılığı 3,35 / 4

Lisans : Ankara Üniversitesi Fen Fakültesi İstatistik 3, 30 / 4

Yüksek Lisans : Trakya Üniversitesi Sağlık Bilimleri Enstitüsü Biyoistatistik ve Tıbbi Bilişim Anabilim Dalı 88,89 / 100