Mehmet Umut ATASEVER Yüksek Lisans Tezi KÜ 2018

T.C.

KIRIKKALE ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

PROTEİN İKİNCİL YAPI TAHMİNİ İÇİN YAPAY SİNİR AĞI MODELLERİNİN OPTİMİZASYONU

Mehmet Umut ATASEVER

Temmuz 2018

Bilgisayar Mühendisliği Anabilim Dalında Mehmet Umut ATASEVER tarafından hazırlanan PROTEİN İKİNCİL YAPI TAHMİNİ İÇİN YAPAY SİNİR AĞI MODELLERİNİN OPTİMİZASYONU adlı Yüksek Lisans Tezinin Anabilim Dalı standartlarına uygun olduğunu onaylarım.

Prof. Dr. Hasan ERBAY
Anabilim Dalı Başkanı

Bu tezi okuduğumu ve tezin **Yüksek Lisans Tezi** olarak bütün gereklilikleri yerine getirdiğini onaylarım.

Dr. Öğr. Üyesi Zafer AYDIN
Ortak Danışman

Prof. Dr. Hasan ERBAY
Danışman

Jüri Üyeleri

Başkan (Danışman) : Prof. Dr. Hasan ERBAY _____
Üye : Dr. Öğr. Üyesi Bülent Gürsel EMİROĞLU _____
Üye : Dr. Öğr. Üyesi Hakan KÖR _____

31/07/2018

Bu tez ile Kırıkkale Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu Yüksek Lisans derecesini onaylamıştır.

Prof. Dr. Mustafa YİĞİTOĞLU
Fen Bilimleri Enstitüsü Müdürü

ÖZET

PROTEİN İKİNCİL YAPI TAHMİNİ İÇİN YAPAY SİNİR AĞI MODELLERİNİN OPTİMİZASYONU

ATASEVER, Mehmet Umut
Kırıkkale Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı, Yüksek Lisans Tezi
Danışman: Prof. Dr. Hasan ERBAY
Ortak Danışman: Dr. Öğr. Üyesi Zafer AYDIN
Temmuz 2018, 32 sayfa

Bu çalışmada protein işlevlerinin anlaşılmasında hayati önemi olan ikincil yapılarının tahmin edilebilmesi için çok katmanlı yapay sinir ağı ve çift yönlü tekrarlayan yapay sinir ağı optimizasyonu yapılmıştır. Bazı optimizasyon işlemleri için özgür yazılımlar kullanılarak bir hesaplama kümesi hazırlanmıştır. Optimizasyon sonucunda çift yönlü tekrarlayan yapay sinir ağlarının çok katmanlı yapay sinir ağından daha başarılı sonuç verdiği görülmüştür.

Anahtar Kelimeler: Biyoenformatik, yapay sinir ağı, protein ikincil yapı tahmini

ABSTRACT

ARTIFICIAL NEURAL NETWORK OPTIMIZATION FOR PROTEIN SECONDARY STRUCTURE PREDICTION

ATASEVER, Mehmet Umut
Kırıkkale University
Graduate School Of Natural and Applied Sciences
Department of Computer Engineering, M.Sc. Thesis
Supervisor: Prof. Dr. Hasan ERBAY
Co- Supervisor: Dr. Zafer AYDIN
July 2018, 32 pages

In this study, a multi-layer artificial neural network and bidirectional recurrent artificial neural network optimization were performed in order to estimate secondary structures that are vital for understanding protein functions. For some optimization operations, a computation grid is prepared using open source software. As a result of the optimization, it has been seen that bidirectional artificial neural networks are more successful than the multi-layer artificial neural network.

Keywords: Bioinformatics, artificial neural networks, protein secondary structure prediction

TEŞEKKÜR

Tezimin hazırlanması esnasında yardımlarını esirgemeyen tez yöneticisi hocalarım, Sayın Prof. Dr. Hasan ERBAY'a ve Dr. Öğr. Üyesi Zafer AYDIN'a, tez çalışmalarım esnasında gerekli hesaplamaların bazılarının yapıldığı TÜBİTAK ULAKBİM, Yüksek Başarım ve Grid Hesaplama Merkezi'ne ve tezimi hazırlamam esnasında yardımlarını esirgemeyen sevgili eşim Sema ATASEVER'e teşekkür ederim.

İÇİNDEKİLER DİZİNİ

Sayfa

ŞEKİLLER DİZİNİ

ŞEKİL Sayfa

ÇİZELGELER DİZİNİ

ÇİZELGE Sayfa

1. GİRİŞ

1.1. Biyoenformatik Hakkında

Bilgisayar yardımı ile biyoenformatik yöntemleri kullanılarak, otomatik kanser teşhisi, gen ifade profillerinin sınıflandırılması ve kuralların çıkarılması gibi çeşitli biyolojik veriler incelenip işlenebilir [1]. İncelenmesi gereken bu biyolojik verilerin miktarı ve çeşitliliği günümüzde hızla artmaktadır. Örneğin proteinlerin yapı bilgilerinin bulunduğu Protein Data Bank'ta 2000 yılında toplam 13.590 proteinin yapı bilgisi bulunurken, 2017 sonu itibari ile bu sayı 136.460'a çıkmıştır [2]. Nükleik asit dizilimlerinin bulunduğu GenBank veritabanında ise 2000 yılı sonunda 10 milyon dizilim bulunurken 2018 Nisan ayında 208 milyonu aşmıştır. Günümüzde biyolojik verilerin yüksek hızla üretiliyor olması bu büyük boyuttaki verilerin işlenmesinde bilgisayarların kullanılmasını zorunlu hale getirmiştir. Biyoenformatik terimi ilk defa 1970 yılında Paulien Hogeweg tarafından biyolojik sistemlerdeki bilgi süreçlerinin incelenmesi anlamında kullanılmış olup günümüzde hesaplamalı teknikler kullanılarak biyolojik verilerin içerdiği bilginin molekül seviyesinde organize edilmesi ve anlaşılması olarak tanımlanmaktadır [3].

1.2. Proteinler

Proteinler, yapı taşları olarak amino asit içeren, canlı hücrelerinde önemli rol oynayan büyük, karmaşık moleküllerdir. Hücrelerdeki işin çoğunu yapan proteinler, vücudun organlarının ve dokularının çalışması ve biçimsel yapılarının oluşması için gereklidirler.

Proteinler bir zincir halinde birbiri ardına eklenmiş yüzlerce veya binlerce amino asitten oluşurlar. Doğada bir protein oluşturmak üzere kullanılabilecek 20 amino asit vardır. Amino asitlerin hangi sırada birleşerek protein oluşturdukları, proteinlerin 3 boyutlu yapısının nasıl olacağını ve dolayısı ile fonksiyonunu belirler. Vücutta ribozomda sentezlenirler. Hayvanlar ihtiyaçları olan proteinlerinin hepsini kendileri

sentezleyemezler. Bunları dışardan almak zorundadırlar. Ayrıca proteinler yapay olarak sentezlenebilmektedirler.

Proteinler vücuttaki hemen her olaya katılırlar ve işlevlerine bağlı olarak gruplandırılırlar.

Taşıyıcı proteinler: Molekülleri vücutta bir yerden bir yere taşıyan taşıyıcı proteinlerdir. Hemoglobin kanda oksijeni taşır, sitokrom(cytochrome)'lar elektron taşır.

Depolama Proteinleri: Amino asitleri depolamak için kullanılırlar. Ovalbumin, yumurta beyazında bulunur. Kazein(casein) ise sütte bulunur.

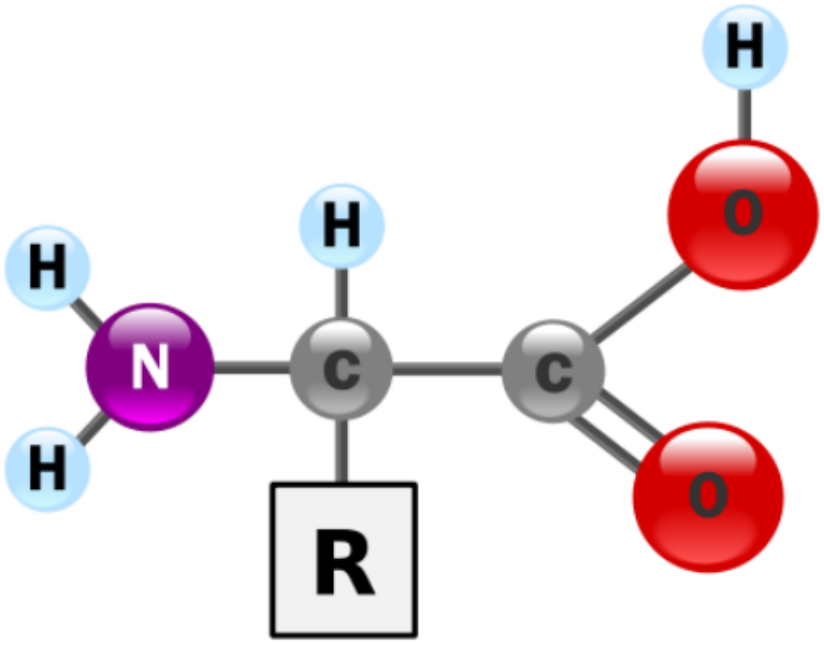
Yapı elemanı olan proteinler: Lifli, tel tel olan ve destek sağlayan proteinlerdir. Örnek olarak Keratin verilebilir.

Hormonsal Proteinler: Belirli vücut işlemlerinin koordine edilmesinde görev alan mesajcı proteinlerdir. Örneğin insülin, şeker miktarının kandaki oranının ayarlanmasını sağlar.

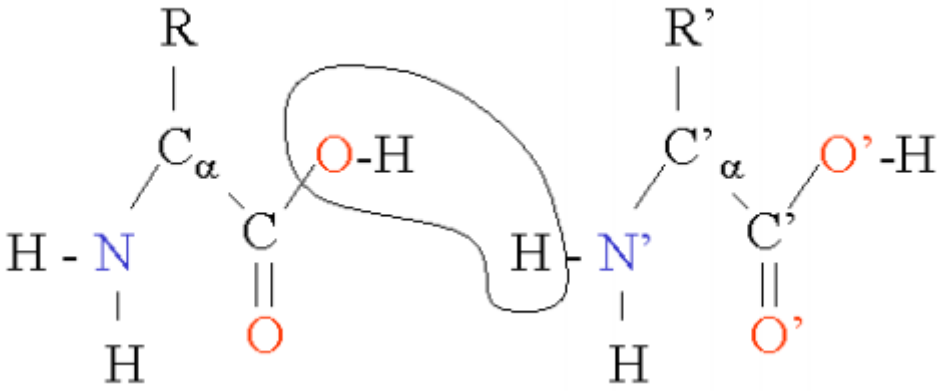
Enzimler: Biyokimyasal reaksiyonları kolaylaştıran proteinlerdir. Genellikle katalizör olarak adlandırılırlar. Laktaz ve pepsin örnek olarak verilebilir. Laktaz, laktoz şekerini yıkıma uğratır. Pepsin ise proteinlerin sindirimini sağlayan enzimdir.

Kısalabilen proteinler: Hareketten sorumludurlar. Aktin ve miyozin örnek olarak verilebilirler. Kasların oluşumunda görev yaparlar.

Antikorlar: Dışarıdan gelen antijenlere karşı vücudu savunurlar. Antijenleri etkisiz hale getirerek beyaz kan hücrelerinin onları yok etmesini sağlarlar.



Şekil 1.1. Serbest Haldeki Bir Proteinin Yapısı [4]



Şekil 1.2. Peptid Bağı Oluşumu

1.3. Makine Öğrenmesi

Makine öğrenmesi, Arthur Samuel tarafından 1959'da bilgisayarların ayrıca programlanmaya gerek kalmadan öğrenmelerini sağlayan çalışma alanı olarak tanımlanmıştır. Tom Mitchell ise 1998 yılında bir bilgisayar programının

öğrenmesinden bahsedebilmek için bir T işiyle ilgili E deneyiminin bir P performans ölçümü temel alınarak, T işiyle ilgili P cinsinden ölçülen performansın E deneyimi ile artması gerektiğini belirtmiştir. Öte yandan, makine öğrenmesi algoritmalarını genel olarak gözetimli öğrenme, gözetimsiz öğrenme ve pekiştirmeli(destekli) öğrenme başlıkları altında toplayabiliriz.

1.3.1. Gözetimli Öğrenme

Gözetimli öğrenmede elimizde giriş değişkenleri (x) ve çıkış değeri (y) varken giriş değerini çıkış değerine işleyen fonksiyon bulunmaya çalışılır. Amaç yaklaşık eşleme fonksiyonunun bulunarak bilinmeyen yeni bir giriş değeri geldiğinde çıkış değerini tahmin edebilmektir. Gözetimli denmesinin sebebi, eğitim veri setinden öğrenme sürecinin bir öğretmenin eğitim sürecini gözetim altında tutması gibi düşünülmesi ve doğru cevabın bilinmesinden dolayı algoritma her hata yaptığında öğretmenin algoritmayı düzeltmesine benzetilmesindendir. Öğrenme kabul edilebilir bir performansa ulaşıldığında durdurulur. Gözetimli öğrenme sınıflandırma ve regresyon problemlerinde kullanılır. Sınıflandırma problemlerinde çıkış değeri “hasta-hasta değil” veya “kırmızı-mavi” gibi bir kategoridir. Regresyon problemlerinde ise çıkış değeri belirli bir miktar para, ağırlık, süre gibi gerçek değerlerdir.

1.3.2. Gözetimsiz Öğrenme

Gözetimsiz öğrenme ise elimizde sadece giriş değerlerinin vardır ama karşılık gelen çıkış değerleri yoktur. Amaç elimizdeki verinin altında yatan yapının modellenerek veri hakkında daha çok bilgi sahibi olmaktır. Elimizde doğru çıkış değerleri olmadığından ve cevapları düzeltecek bir öğretmenden söz edilemediğinden gözetimsiz öğrenme olarak adlandırılır. Kümeleme ve ilişki bulma problemlerinde kullanılır.

Kümeleme problemlerinde amaç verinin içindeki gruplaşmaları bulmaktır. Örneğin satın alma davranışlarına göre müşterilerin gruplara ayrılması gibi.

İlişki bulma problemlerinde ise amaç verinin çoğunluğunu tarif eden kuralları bulmaktır. Örneğin X malını alan müşterilerin genellikle Y malını da alması gibi.

1.3.3. Pekiştirmeli (Destekli) Öğrenme

Pekiştirmeli öğrenmede gözetimsiz öğrenmede olduğu gibi doğru cevapların elimizde olmadığı bir durum söz konusudur. Doğru cevap elinde olmamasına rağmen algoritmamız yine de kendisine verilen işi tamamlamak için bir karar vermek zorunda kalır. Eğitim verisi olmadığından algoritmamız öğrenmek için deneyimlerini kullanır. Deneme yanılma ile eğitim verisini oluşturan algoritmanın amacı uzun dönemde elde edeceği ödülü maksimize etmeye çalışır.

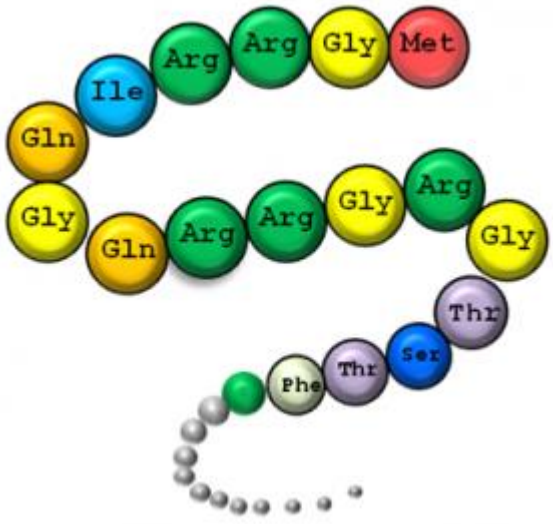
1.4. Protein Yapı Tahmini

Günümüzde biyoenformatiğin önemli araştırma konularından biri olan ve uygulanabilirlik aralığı giderek artan protein yapı tahmin yöntemleri, deneysel olarak yapısı bilinmeyen proteinlerin yapıları hakkında öngörüde bulunmayı amaçlamaktadır ve geçmiş yıllarda yapılan dizi ve yapı bilgisi çalışmalarından elde edilen tecrübeler ve analiz araçlarındaki gelişmelerle ilerleme sağlanan bir çalışma konusu konumundadır [5].

Protein zincirlerinin ikincil yapılarının ve üç boyutlu yapılarının birincil amino asit diziliminden doğru bir şekilde tahmin edilmesi, protein katlanmasını anlama adına atılan önemli bir adımdır ve hesaplamalı moleküler biyolojide temel bir problemdir [6]. Bunun için amino asitlerin fizikokimyasal özellikleri, sekans homolojisi, desen eşleştirmesi ve bilinen yapıdaki proteinlerin istatistiksel analizlerinin kullanılması gibi tahmin başarısını artırmaya yarayan çeşitli yöntemler önerilmiştir [7]. Bu özelliklerin çoğu için, genel olarak makine öğrenme yöntemleri ve daha spesifik olarak nöral ağ yaklaşımlarının kullanılmasının etkili olduğu kanıtlanmıştır [8].

1.4.1. Birincil Yapı

Proteinler çok sayıda amino asidin art arda eklenmesi sonucunda oluşurlar. Art arda eklenen amino asitlerin sırasını belirten diziye proteinin birincil yapısı denmektedir. Birincil yapıda amino asitler genellikle bir harfli ve üç harfli kısaltmalarıyla belirtilirler. Birincil yapı her ne kadar proteinin hangi amino asitlerden oluştuğunu gösterse de, proteinin işlevi hakkında bize bir bilgi vermez. Çünkü proteinin işlevini belirleyen üç boyutlu yapısıdır.



Şekil 1.3. Proteinin Birincil Yapısı

1.4.2. İkincil Yapı

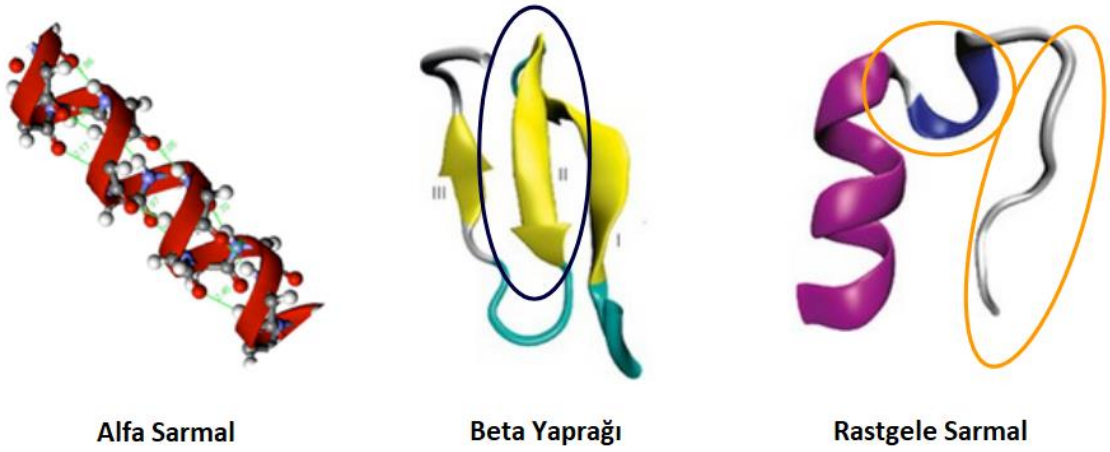
Bir proteini oluşturan aminoasitlerin birbirleri ile atom düzeyinde etkileşime girmesi proteinlerde yerel üç boyutlu yapıların oluşmasını sağlar. Bu yapıların en çok görülenleri alfa sarmal ve beta yapraktır. Bu yapılar atomlar arası oluşan hidrojen bağları ile şekillenirler. Yerel yapılar oldukları için R gruplarının etkisi ikincil yapıda anlaşılamaz.

Alfa sarmal yapısı, amino asidin birinin karbonil grubunun dizinin dört altındaki amino asidin amino grubu ile hidrojen bağı oluşturması ile polipeptid dizisinin

kıvrılmış bir kuşak şekline gelmesi ile oluşur. Bu yapıda amino asitlerdeki R grupları yapıdan dışarı doğru sarkar ve böylece etkileşime açık olurlar.

Beta yaprağı yapısı bir polipeptid'in iki ya da daha fazla kısmının hidrojen bağları sayesinde birbirlerinin karşısına gelecek şekilde dizilmeleri sayesinde oluşurlar. R grupları dizilerin hidrojen bağlarının oluşmadığı dış kısımlarında yer alırlar.

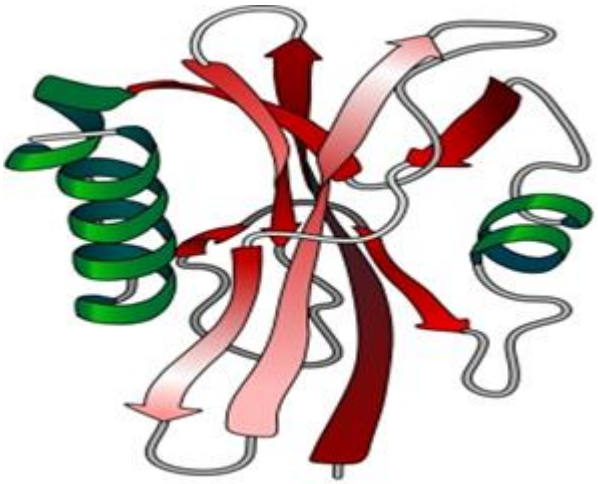
Alfa sarmal ve beta yaprağı yapısına uymayan düzensiz üç boyutlu yapılar ve dönüşler de proteinlerin yerel yapıları arasında görülürler. Bu yapılar genellikle {Alfa sarmalı, beta yaprağı} yapısını bir diğer {Alfa sarmalı, beta yaprağı} yapısına bağlarken oluşurlar.



Şekil 1.4. İkincil Yapıda Sık Görülen Alt-yapı Çeşitleri

1.4.3. Üçüncül Yapı

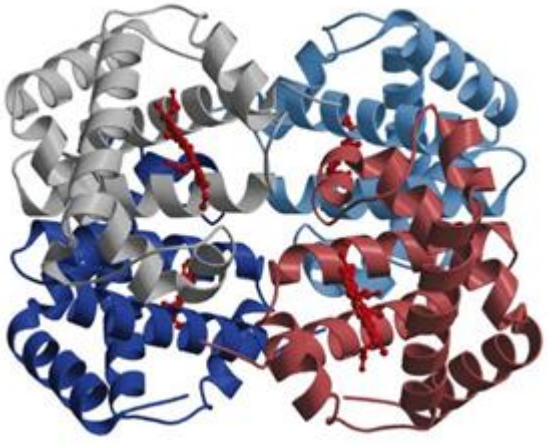
Tek bir polipeptid dizisinden oluşan proteinin tüm üç boyutlu yapısına üçüncül yapısı denir. Üçüncül yapı bir ya da daha fazla ikincil yapı türü barındırır ve üç boyutlu yapısının nasıl şekilleneceği öncellikle proteinin yapısında bulunan R gruplarının etkileşimiyle belirlenir.



Şekil 1.5. Üçüncül Yapı

1.4.4. Dördüncül Yapı

Birçok protein tek bir polipeptid zincirinden oluşmakta olup üç seviyeli yapıya sahiptirler. Bazı proteinler ise birden fazla polipeptid zincirinden oluşmakta olup, bu farklı polipeptid zincirleri bir araya gelerek dördüncül yapıyı oluşturur.



Şekil 1.6. Dördüncül Yapı

1.5. Yapay Sinir Ağları

Çoklu dizi hizalamalarında yer alan bilgiler, yapay sinir ağlarının eğitiminde kullanılarak protein ikincil yapı tahmininde önemli ölçüde başarılı sonuçların elde edildiği gözlenmiştir. Rost ve Sander yapay sinir ağı kullanarak yaptıkları çalışmada 126 eşsiz protein zincirine çoklu çapraz doğrulama uygulayarak %71,6 doğruluğa sahip bir model elde etmişlerdir. Öğrenim setiyle anlamlı bir benzerlik göstermeyen 124 yeni çözülmüş protein yapısına dair test kümesi ile, yüksek doğruluk düzeyini doğrulamışlardır. 250 eşsiz protein zincirinin ortalama çapraz doğrulama oranı ise %72'nin üzerinde çıkmıştır [9]. Holley ve Karplus, bir sinir ağına dayalı protein ikincil yapı tahmini için bir yöntem sunarak ağın bilinen yapıdaki 48 proteinlik örnek bir kümesi üzerindeki ikincil yapı ve amino asit sekansları arasındaki ilişkiyi tanıması için bir eğitim fazı kullanmışlardır. Bilinen yapıya sahip 14 proteinden oluşan ayrı bir test kümesinde yöntemi test ederek üç durum için (sarmal, yaprak, ve coil.) %63'lük maksimum öngörme doğruluğunu elde etmişlerdir [7].

1.5.1. Çok Katmanlı Yapay Sinir Ağı

Çok katmanlı sinir ağlarının, lineer olmayan problemlerin çözümünde en sık kullanılan yapay sinir ağı modellerinden biri olmasından dolayı birçok araştırmanın konusu olmuştur [10]. Çok katmanlı sinir ağı bir giriş katmanı, en az bir gizli katman ve bir çıktı katmanı olmak üzere üç katmandan meydan gelen ve ileri besleme ile geri yayılım algoritmalarına sahip olan bir ağdır [11].

1.5.2. Tekrarlayan Yapay Sinir Ağları

İleriye dönük sinir ağları, protein yapısının tahmin problemlerinde kullanılan, özellikle, ikincil yapının kestiriminde kullanılan en önemli makine öğrenme araçlarından biri olmuştur [8].

Pollastri ve arkadaşları proteinlerde kalıntı sayısının tahmininin geliştirilesi için iki yönlü tekrarlayan yapay sinir ağı mimarileri kullandıkları çalışmalarında, kabul edilen yarıçapa bağlı olarak %70,1 ile %73,1 arasında değişen performanslara imza atmışlardır [8].

2. MATERYAL VE YÖNTEM

2.1. Veri Kümesi

Yapay sinir ağlarının eğitimi ve testi için CB512 protein veri kümesi kullanılmıştır. CB512 veri kümesinde 512 adet proteinin birincil ve ikincil yapı bilgisi bulunmaktadır. Özellik çıkarımı için cb512 veri kümesinde bulunan proteinlerle benzeşen Protein Database Bank'te ve NR protein database'de bulunan proteinlerin yapıları da kullanılmıştır. Bu iki veri kümesindeki proteinlerle hizalama yapmak için ise PSI-BLAST algoritması kullanılmıştır.

Kullandığımız DSPRED ismi verilen metot iki aşamalı bir sınıflandırma metodudur. Bu aşamalar dinamik bayes ağı (DBA) ve yapay sinir ağından (YSA) oluşmaktadır. PSI-BLAST ve HHBlits'in ilk aşaması tarafından üretilen her pozisyona özgü skor matrislerinin her biri için ayrı bir DBA eğitilmektedir. Daha sonra tahminler HHBlitz metodunun ikinci aşamasından elde edilen yapısal profil matrisleri ile birleştirilmekte ve YSA sınıflandırıcısına giriş değeri olarak verilmektedir.

2.2. Yapay Sinir Ağları

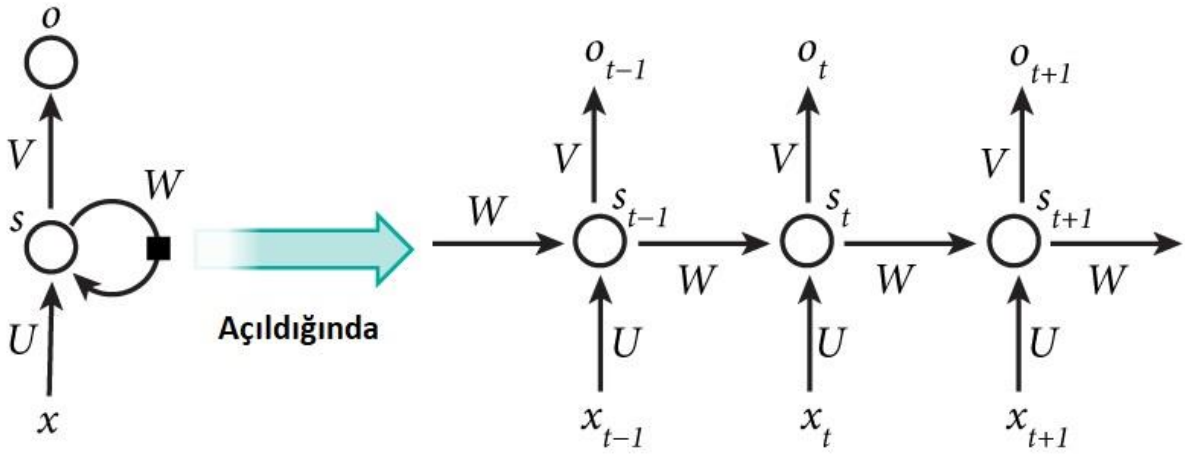
Yapay sinir ağları, bir hesaplamalı problemi çözmek için birbirine bağlı çok sayıda basit hesaplama birimini içeren bilgi işleme sistemleridir. Canlıların sinir sisteminden esinlenilerek geliştirilmiştir. Geleneksel hesaplamalı problem çözme yöntemlerinde her bir problemi çözmek için özel algoritmalar tasarlanmasının gerekmesi problemin tamamen anlaşılmasını gerektirirken, yapay sinir ağları farklı bir yol izleyerek eldeki veriler kullanılarak sistemin eğitilmesini ve hatanın en aza düşürülerek problemin çözümüne giderek yaklaşılmasını hedeflemektedir.

2.2.1. Geçmişi

Yapay sinir ağları ile ilgili ilk çalışma nöronların nasıl çalışabiliyor olduklarına dair Warren McCulloch ve Walter Pitts'in 1943 yılında yazdıkları "The Logical Calculus of the Ideas Immanent in Nervous Activity" makalesi ile başlar. Makalelerinde kullandıkları ağ, sabit bir eşik değeri kullanan ikili sonuç veren basit elemanlardan oluşmaktadır. Bir psikolog olan Donald O. Hebb 1949 yılında yazdığı "The Organization of Behavior" kitabında insanların nasıl öğrendiği konusunda temel bir keşif olan nöron yolarının her kullanıldığında daha da güçlendiğinden bahseder.

2.2.2. Tekrarlayan Yapay Sinir Ağları

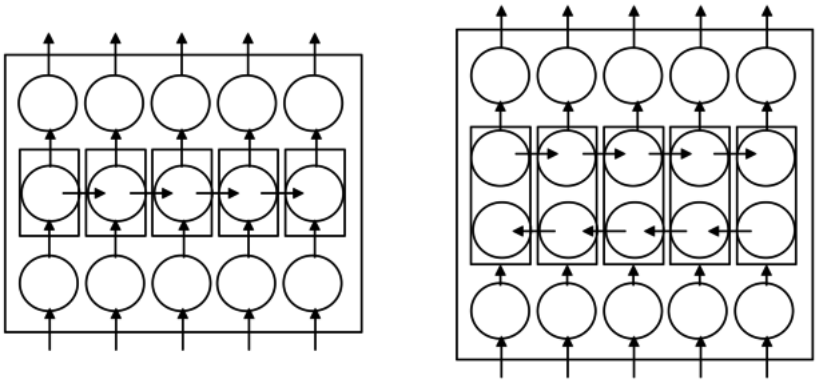
Tekrarlayan yapar sinir ağları ile ardışık bilgilerin sonraki çıktıyı etkilemesi prensibine göre çalışır. Bu yapay sinir ağı çeşidinde çıktıyı hesaplamak için önceki hesaplamanın sonucu da bir girdi olarak kullanılır.



Şekil 2.1. Tekrarlayan Yapay Sinir Ağı

2.2.3. Çift Yönlü Tekrarlayan Yapay Sinir Ağları

Çift yönlü tekrarlayan yapay sinir ağları (BRNN) 1997 yılında Schuster ve Paliwal tarafından bulunmuştur [12]. BRNN yönteminde geçmiş ve gelecek yönlerindeki veriler aynı çıkış katmanına bağlanarak, ağın aynı anda hem gelecek hem de geçmiş verilere erişimi olması sağlanmıştır.



Şekil 2.2. Tekrarlayan Yapay Sinir Ağı ile Çift Yönlü Tekrarlayan Yapay Sinir Ağı

2.3. Deneylerin Yapılması İçin Kullanılan Hesaplama Kümesi Sistemi

Yapay sinir ağı parametre optimizasyonu çok sayıda parametrenin değişik değerleri için hesaplama yapılmasını gerektirdiğinden yüksek işlem gücüne ihtiyaç duymaktadır. Sıradan bir sunucu veya masaüstü bilgisayarda bu kadar çok deneyin yapılması çok uzun süreceğinden bir hesaplama kümesi kullanılması gerekmiştir.

2.3.1. Hesaplama Kümesi Hakkında

Günümüzde veri üretim kaynaklarının sayısındaki artış ve buna bağlı olarak da elde edilen verilerin bir hayli büyümesiyle, bu verilerin işlenmesi için gerekli hesaplama gücü ihtiyacı da artmıştır. Özellikle çok boyutlu işlemlerde veri işleme ve hesaplama

çok uzun süre alabilmektedir. Bu gibi durumlarda aynı anda birden fazla işin çalışabilmesi için çok işlemcili güçlü bilgisayarlar gerekmektedir. Küme hesaplama (cluster computing), hem bu ihtiyacı karşılamak hem de maliyeti düşürmek amacıyla kullanılan yöntemlerden birisidir. Birden fazla bilgisayarın kaynaklarının birleştirilerek yüksek performanslı tek bir veya ağ üzerinden büyük bir paralel bilgisayar gibi davranmasına küme hesaplama denilmektedir. Kümedeki bilgisayarlar birbirlerine fiziksel ve mantıksal olarak bağlıdırlar.

Küme hesaplama genetik, biyoenformatik, veri madenciliği, matematik ve benzeri birçok farklı alanda kullanılmaktadır. Kümeler amaçlarına göre, yüksek erişilebilirlik (high-availability), yüksek performans hesaplama (high performance computing) ve yük dengeleme (load-balancing) olmak üzere üç grupta sınıflandırılırlar. Bu çalışmada kullandığımız küme yapısı yüksek performans hesaplama içerisine girmektedir ve bir Beowulf [13] kümesi olarak nitelendirilebilir. Beowulf ilk olarak 1994 yılında NASA tarafından kurulmuştur. O tarihten sonra heterojen yapıda normal bilgisayarlar tarafından kurulan kümelerin genel ismi olarak kullanılmaktadır. Beowulf bilgisayar kümesi, sunucu ve düğümlerden oluşur. Günümüzde küme yapıları kolay ve az maliyetle kurulabilir hale gelmiştir. Kullandığımız küme yapısı modülerdir, açık kaynak kodludur ve kurulumu kolaydır.

Kurulan bu hesaplama kümesi tamamıyla ücretsiz ve açık kaynak yazılımlardan oluşmaktadır. Küme bir adet küme sunucusu ve sayısı isteğe bağlı değişken olmak üzere hesaplama düğümlerinden oluşmaktadır. Sistemin kullanacağı depolama alanı küme sunucusunda veya ağa bağlı bir depolama aygıtında bulunmaktadır. Bellek ve işlemci ihtiyacını ise hesaplama düğümleri karşılamaktadır. Bu nedenle yönetici olarak kullanılan küme sunucusunda depolama alanı büyük olabilecekken, bellek miktarı ve işlemci gücünün çok yüksek olmasına gerek yoktur. Düğümlerde ise yüksek bellek miktarı ve işlemci gücü olması tercih edilir.

Bu çalışmada, depolama alanı küme sunucusu içerisinde ayrı bir bölüm halinde olacak şekilde tasarlanmıştır ve sistemdeki tüm bilgisayarlarda (küme sunucuları ve hesaplama düğümleri) işletim sistemi olarak kümenin kurulum kolaylığı dikkate alınarak Centos 7 tercih edilmiştir. Kümenin dosya sistemi olarak NFS'ten daha hızlı

ve I/O performansının daha iyi olmasından dolayı Lustre [14] seçilmiştir. Kaynak yönetim yazılımı olarak da kullanım kolaylığı ve yapıya uygunluğu sebebiyle Slurm [15] kullanılmıştır. Bu küme heterojen bir yapı kurulmasına uygun olup, hesaplama düğümleri sunucu, masaüstü ve dizüstü bilgisayarlardan seçilebileceği gibi sanal bilgisayarlardan ve Raspberry Pi gibi düşük bütçeli mini bilgisayarlardan da seçilebilir.

2.3.2. Hesaplama Kümesi Kurulumu

Küme sunucusu kurulumunda, öncelikle sunucuda Selinux ve Firewall devre dışı bırakılır ve sunucunun ismi (hostname) ayarlanır. Gerekli Lustre paketleri kurulur. Lustre, Linux işletim sistemleri üzerinde kullanılabilen, paylaşımlı dağıtık bir dosya sistemidir. Lustre için gerekli yapılandırmalar yapıldıktan sonra bir dizin oluşturulup, kümenin depolama alanı olarak kullanılacak disk bölümü Lustre formatında formatlanır ve dizine bağlanır. Daha sonrasında ise Slurm yazılımının kullanacağı MariaDB veri tabanı yönetim sistemi ile yine Slurm için gerekli olan kimlik doğrulama eklentisi Munge kurulur. Munge için gerekli yapılandırmalar yapıldıktan sonra Slurm yazılımının paketleri yüklenir ve Slurm yapılandırılır. Kümeyi kullanacak tüm kullanıcılar için kullanıcı yetkilendirmeleri yapılmalıdır. Kümedeki tüm bilgisayarların aynı sistem saatine sahip olması amacıyla NTP yazılımı kurulup çalıştırılır. Slurm yazılımı çalıştırıldıktan sonra küme sunucusunda yapılacak işler şimdilik tamamlanmış demektir.

2.3.3. Hesaplama Düğümlerinin Kurulumu

Hesaplama düğümlerinin kurulumunda, sunucu kurulumunda olduğu gibi Selinux ve Firewalld devre dışı bırakılır ve sunucunun ismi ayarlanır. Gerekli Lustre paketleri kurulur ve hesaplama sunucusunda oluşturulan Lustre formatlı depolama alanı düğüme bağlanır. Böylece sunucudaki depolama alanına hesaplama düğümü de erişebilir hale getirilir. Bir sonraki aşamada hesaplama düğümüne de Munge ve Slurm yazılımları kurulur. Her ikisi için de gereken yapılandırma ayarları yapılır. NTP

yazılımı kurulup sunucuyla aynı yapılandırmaya sahip olacak şekilde ayarlanır. Sunucu tarafına geçilip Slurm yapılandırmasına hesaplama düğümü eklenir ve düğümün Slurm servisi başlatılır. Daha sonra sunucunun Slurm servisi yeniden başlatılır. Sunucudaki Slurm servisinin yeniden başlatılması -varsa- o anda çalışan işleri etkilememektedir. Bu durum kümeye istenilen zamanda yeni bir düğüm eklenebileceği anlamına gelmektedir.

Bir önceki adım tekrar edilerek istenilen sayıda düğüm kümeye eklenebilir. Kümeye en az bir düğüm eklendikten sonra slurm batch script'i kullanılarak kümeye iş gönderilebilir. Kümede çalıştırılmak istenilen işler ilave program kurulumuna ihtiyaç duyuyorsa tüm düğümlere ilgili programları yüklemek gerekmektedir. Bu yükleme işinin tüm düğümlere tek tek yapılması yerine Chef, Puppet veya Ansible gibi konfigürasyon yönetim yazılımlarıyla pratik bir şekilde yapılabilir.

2.3.4. Küme Mimarisi

Yapmakta olduğumuz biyoenformatik çalışmalarında protein ikincil yapı tahmininde kullanılmak üzere farklı protein veri kümeleri üzerinde yapısal profiller üretilmektedir. Yapısal profil matrisi üretmek üzere hazırlanmış olan algoritmanın veri kümesindeki her bir protein için çalıştırılması gerekmekte ve her protein için bu işlem ilgili proteinin uzunluğuna göre birkaç saat sürebilmektedir. Hal böyle olunca bu işlemi işlemcilere bölmeden ve tek bir bilgisayar üzerinde gerçekleştirmek günlerce hatta aylarca sürebilmektedir. Kullandığımız diğer sunuculardaki ve grid sistemlerindeki yoğunluk ve hız durumunu da göz önüne alındığında eldeki imkânlar doğrultusunda bir hesaplama kümesi oluşturulmaya karar verilmiştir. Kullanıma uygun veya atıl durumda bulunan bilgisayarlar tespit edilip bunlar kümeye dâhil edilmiştir. Ayrıca bu mimari sayesinde mevcut işler çalışmakta iken onlara zarar gelmeden kümeye hesaplama düğümü eklenip çıkarılabilecek durumdadır.

Çizelge 2.1. Küme Sunucusu

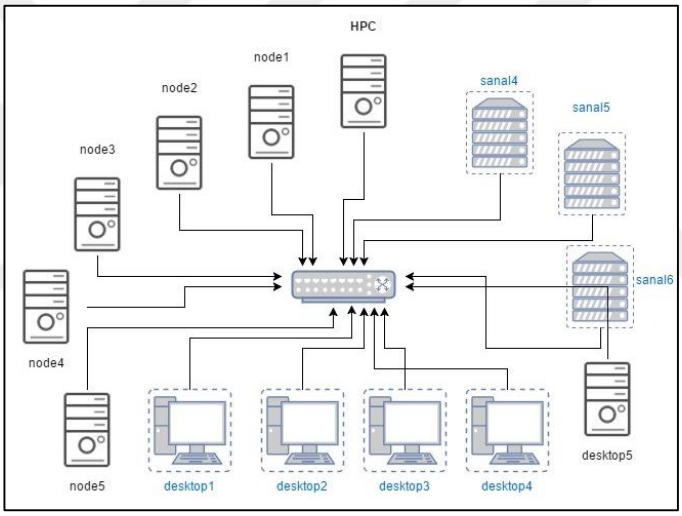
CPU	İşlemci Marka/Model	Bellek (MB)	Tip	Düğüm İsmi
4	Intel(R) Core(TM) i7-4790 3,60 GHz	3664	Masaüstü	hpc

Çizelge 2.2. Hesaplama Düğümleri

CPU	İşlemci Marka/Model	Bellek (MB)	Tip	Düğüm İsmi
4	Intel(R) Core(TM) i7-4790 3,60 GHz	3664	Masaüstü	node1
4	Intel(R) Core(TM) i7-4790 3,60 GHz	3664	Masaüstü	node2
4	Intel(R) Core(TM) i7-4790 3,60 GHz	3664	Masaüstü	node3
4	Intel(R) Core(TM) i7-4790 3,60 GHz	3664	Masaüstü	node4
4	Intel(R) Core(TM) i7-4790 3,60 GHz	3664	Masaüstü	node5
10	Intel(R) Xeon(R) E5-2660 2,20 GHz	19916	Sanal (Proxmox)	sanal4
8	Intel(R) Xeon(R) E5-2660 2,20 GHz	19917	Sanal (Proxmox)	sanal5
8	Intel(R) Xeon(R) X5450 3,00 GHz	14878	Sanal (Proxmox)	sanal6
4	Intel(R) Core(TM) i5-4590 3,30 GHz	2000	Sanal (VirtualBox)	desktop1
4	Intel(R) Xeon(R) W3550 3,07 GHz	4203	Sanal (VirtualBox)	desktop2
4	Intel(R) Core(TM) i7-4790 3,60 GHz	2000	Sanal (VirtualBox)	desktop3
4	Intel(R) Core(TM) i7-4790 3,60 GHz	15955	Sanal (VirtualBox)	desktop4
4	Intel(R) Core(TM) i5 760 2,80 GHz	7820	Masaüstü	desktop5

Oluşturulan kümede 1 adet küme sunucusu ve 13 adet düğüm bulunmaktadır. Küme sunucusunun özellikleri Çizelge 2.1'de, düğümlerin özellikleri ise Çizelge 2.2'de verilmiştir. Düğümler masaüstü bilgisayarlar, masaüstü bilgisayarlar içerisine kurulan sanal bilgisayarlar (VirtualBox) ve Proxmox sunucular [16] içerisinde bulunan sanal bilgisayarlar olmak üzere üç tipten oluşmaktadır.

Şekil 2.3.'de hesaplama kümesinin mimarisi görülmektedir. Kümede bulunan tüm bilgisayarlar yerel alan (LAN) ağında dağıtık vaziyette, her biri farklı yerlerde fakat birbiriyle haberleşebilecek durumdadır. Bilgisayarların aynı ortamda bulunmaları gerekmez, aynı ağa bağlı olmaları yeterlidir.



Şekil 2.3. Küme Mimarisi

3. ARAŞTIRMA BULGULARI VE TARTIŞMA

Çok katmanlı yapay sinir ağı ve çift yönlü tekrarlayan yapay sinir ağı deneyleri 7 katlamalı çapraz doğrulama (7 fold cross validation) yöntemiyle yapılmış olup sonuçları aşağıda verilmiştir.

3.1. Çok Katmanlı Yapay Sinir Ağı Sonuçları

Çizelge 3.1. MLP 1.Fold İçin En İyi 10 Başarı Oranına Sahip Parametreler

Gizli Katmanlar	Gizli Katmandaki Nöron Sayısı	Öğrenme Hızı	Momentum	Korelasyon Katsayısı	Seyreltme Değeri	Devir Sayısı	Başarı
3	75	0,1	0,001	0,0001	0	5	80,44
4	75	0,1	0,01	0,0001	0	5	79,83
4	75	0,1	0,1	0,0001	0	5	79,62
3	75	0,1	0,001	0,001	0	5	79,52
1	200	0,1	0,1	0,001	0	10	79,5
1	200	0,1	0,5	0,001	0	5	79,49
2	100	0,1	0,01	0,0001	0	5	79,48
4	200	0,1	0,01	0,0001	0	5	79,46
1	75	0,1	0,9	0,001	0	50	79,45
5	100	0,1	0,5	0,0001	0	5	79,45

Çizelge 3.2. MLP 2.Fold İçin En İyi 10 Başarı Oranına Sahip Parametreler

Gizli Katmanlar	Gizli Katmandaki Nöron Sayısı	Öğrenme Hızı	Momentum	Korelasyon Katsayısı	Seyreltme Değeri	Devir Sayısı	Başarı
1	200	0,01	0,5	0,01	0	5	78,3
1	200	0,01	0,1	0,01	0	5	78,21
1	200	0,01	0,01	0,01	0	5	78,21
1	200	0,01	0,9	0,01	0	10	78,16
1	75	0,01	0,01	0,01	0	5	78,14
1	200	0,01	0,01	0,01	0	10	78,01
1	200	0,01	0,1	0,01	0	50	77,99
1	100	0,01	0,9	0,01	0	5	77,98
1	100	0,01	0,1	0,01	0	10	77,98
1	100	0,01	0,1	0,01	0,5	5	77,98

Çizelge 3.3. MLP 3.Fold İçin En İyi 10 Başarı Oranına Sahip Parametreler

Gizli Katmanlar	Gizli Katmandaki Nöron Sayısı	Öğrenme Hızı	Momentum	Korelasyon Katsayısı	Seyreltme Değeri	Devir Sayısı	Başarı
1	200	0,001	0,1	0,001	0	50	76,2
1	200	0,001	0,01	0,001	0	5	76,19
1	200	0,001	0,5	0,001	0,5	50	76,14
1	200	0,001	0,9	0,001	0	25	76,13
1	200	0,001	0,001	0,001	0,5	25	76,1
1	200	0,001	0,1	0,001	0	25	76,04
1	200	0,001	0,5	0,001	0	25	76,03
1	200	0,001	0,1	0,001	0	10	76,03
1	200	0,001	0,01	0,001	0	10	76,03
1	200	0,001	0,001	0,001	0	50	76,02

Çizelge 3.4. MLP 4.Fold için en iyi 10 başarı oranına sahip parametreler

Gizli Katmanlar	Gizli Katmandaki Nöron Sayısı	Öğrenme Hızı	Momentum	Korelasyon Katsayısı	Seyreltme Değeri	Devir Sayısı	Başarı
1	200	0,001	0,9	0,001	0	50	79,13
1	200	0,001	0,9	0,001	0	25	79,04
1	200	0,001	0,01	0,001	0	50	79,01
1	200	0,001	0,01	0,001	0	25	78,98
1	200	0,001	0,001	0,001	0	25	78,98
1	200	0,001	0,1	0,001	0	50	78,95
1	200	0,001	0,1	0,001	0	10	78,93
1	200	0,001	0,1	0,001	0	25	78,93
1	200	0,001	0,001	0,001	0,5	50	78,93
1	200	0,001	0,5	0,001	0	25	78,9

Çizelge 3.5. MLP 5.Fold için en iyi 10 başarı oranına sahip parametreler

Gizli Katmanlar	Gizli Katmandaki Nöron Sayısı	Öğrenme Hızı	Momentum	Korelasyon Katsayısı	Seyreltme Değeri	Devir Sayısı	Başarı
3	200	0,1	0,5	0,0001	0	5	78,31
1	200	0,01	0,01	0,001	0	5	78,25
1	200	0,01	0,001	0,001	0,5	5	78,2
1	200	0,01	0,1	0,001	0	5	78,19
1	200	0,01	0,9	0,001	0	5	78,09
1	200	0,01	0,01	0,001	0	10	78,07
1	200	0,01	0,9	0,001	0,5	5	78,02
1	200	0,01	0,01	0,001	0,5	5	78,02
1	75	0,01	0,001	0,001	0	5	78
1	200	0,01	0,5	0,001	0	5	78

Çizelge 3.6. MLP 6.Fold için en iyi 10 başarı oranına sahip parametreler

Gizli Katmanlar	Gizli Katmandaki Nöron Sayısı	Öğrenme Hızı	Momentum	Korelasyon Katsayısı	Seyreltme Değeri	Devir Sayısı	Başarı
4	200	0,1	0,001	0,0001	0	10	75,37
4	100	0,1	0,01	0,0001	0	5	74,76
1	200	0,01	0,5	0,001	0	10	73,93
4	50	0,1	0,1	0,0001	0	5	73,79
1	200	0,01	0,1	0,001	0	10	73,7
1	75	0,01	0,001	0,001	0	5	73,6
1	200	0,01	0,1	0,001	0	5	73,56
1	100	0,01	0,01	0,001	0	5	73,41
1	200	0,01	0,5	0,001	0	5	73,41
1	100	0,01	0,1	0,001	0,5	5	73,4

Çizelge 3.7. MLP 7.Fold için en iyi 10 başarı oranına sahip parametreler

Gizli Katmanlar	Gizli Katmandaki Nöron Sayısı	Öğrenme Hızı	Momentum	Korelasyon Katsayısı	Seyreltme Değeri	Devir Sayısı	Başarı
1	200	0,1	0,5	0,0001	0	5	76,97
1	200	0,1	0,9	0,0001	0	5	76,63
5	200	0,1	0,001	0,0001	0	10	76,59
5	75	0,1	0,1	0,0001	0	5	76,51
5	200	0,1	0,5	0,0001	0,5	5	76,36
3	200	0,1	0,1	0,0001	0	5	76,24
5	100	0,1	0,01	0,0001	0	5	76,23
4	75	0,1	0,9	0,0001	0	5	76,22
5	200	0,1	0,001	0,0001	0,5	5	76,18
2	100	0,1	0,9	0,0001	0	5	76,15

3.2. Çift Yönlü Tekrarlayan Yapay Sinir Ağı Sonuçları

Çizelge 3.8. BRNN Parametre Testi – Fold 1 İçin En İyi 10 Sonuç

Başarı oranı	İterasyon Sayısı	Gizli Katman Sayısı	İleri ağ ve Geri ağ Pencere Genişliği	İleri ağ ve Geri ağ Çıkış Sayısı	İleri ağ ve Geri ağ Gizli Katman Düğüm Sayısı	Öğrenme Hızı
83,8844	6000	20	8	8	9	0,4
83,8325	5500	5	8	8	9	0,8
83,8066	6000	5	8	8	9	0,8
83,7936	6000	10	8	8	9	0,4
83,7677	5500	10	8	8	9	0,6
83,7547	5000	5	8	8	9	0,8
83,7417	3000	5	8	8	9	0,8
83,7417	4500	10	8	8	9	0,8
83,7417	5500	10	8	8	9	0,4
83,7158	5000	20	8	8	9	0,6

Çizelge 3.9. BRNN Parametre Testi – Fold 2 İçin En İyi 10 Sonuç

Başarı oranı	İterasyon Sayısı	Gizli Katman Sayısı	İleri ağ ve Geri ağ Pencere Genişliği	İleri ağ ve Geri ağ Çıkış Sayısı	İleri ağ ve Geri ağ Gizli Katman Düğüm Sayısı	Öğrenme Hızı
81,1128	5000	10	8	8	9	0,6
81,099	5500	10	8	8	9	0,6
81,0852	5000	10	8	8	9	0,8
81,0852	6000	20	8	8	9	0,4
81,0852	6000	10	8	8	9	0,8
81,0714	5500	10	8	8	9	0,8
81,0714	6000	20	8	8	9	0,6
81,0576	6000	30	8	8	9	0,6
81,0576	6000	10	8	8	9	0,6
81,0023	4500	10	8	8	9	0,6

Çizelge 3.10. BRNN Parametre Testi – Fold 3 İçin En İyi 10 Sonuç

Başarı oranı	İterasyon Sayısı	Gizli Katman Sayısı	İleri ağ ve Geri ağ Pencere Genişliği	İleri ağ ve Geri ağ Çıkış Sayısı	İleri ağ ve Geri ağ Gizli Katman Düğüm Sayısı	Öğrenme Hızı
83,7596	5000	30	8	8	9	0,4
83,717	4500	30	8	8	9	0,4
83,717	5500	30	8	8	9	0,4
83,7028	5500	30	8	8	9	0,2
83,6885	3000	30	8	8	9	0,4
83,6743	4000	30	8	8	9	0,6
83,6743	6000	30	8	8	9	0,2
83,6743	6000	30	8	8	9	0,8
83,6601	4000	30	8	8	9	0,4
83,6601	4500	30	8	8	9	0,6

Çizelge 3.11. BRNN Parametre Testi – Fold 4 İçin En İyi 10 Sonuç

Başarı oranı	İterasyon Sayısı	Gizli Katman Sayısı	İleri ağ ve Geri ağ Pencere Genişliği	İleri ağ ve Geri ağ Çıkış Sayısı	İleri ağ ve Geri ağ Gizli Katman Düğüm Sayısı	Öğrenme Hızı
83,2196	5500	20	8	8	9	0,8
83,2034	5000	20	8	8	9	0,8
83,2034	6000	30	8	8	9	0,6
83,1709	6000	20	8	8	9	0,8
83,1546	5500	30	8	8	9	0,6
83,1546	6000	30	8	8	9	0,8
83,1384	6000	30	8	8	9	0,4
83,1222	4500	20	8	8	9	0,8
83,1059	6000	20	8	8	9	0,6
83,0897	5000	30	8	8	9	0,6

Çizelge 3.12. BRNN Parametre Testi – Fold 5 İçin En İyi 10 Sonuç

Başarı oranı	İterasyon Sayısı	Gizli Katman Sayısı	İleri ağ ve Geri ağ Pencere Genişliği	İleri ağ ve Geri ağ Çıkış Sayısı	İleri ağ ve Geri ağ Gizli Katman Düğüm Sayısı	Öğrenme Hızı
84,5419	6000	5	8	8	9	0,4
84,5004	6000	30	8	8	9	0,4
84,4727	4000	5	8	8	9	0,6
84,4727	6000	5	8	8	9	0,6
84,4589	4000	5	8	8	9	0,4
84,4589	5500	5	8	8	9	0,4
84,4451	3500	5	8	8	9	0,4
84,4451	4000	30	8	8	9	0,4
84,4451	5000	30	8	8	9	0,4
84,4451	5000	5	8	8	9	0,4

Çizelge 3.13. BRNN Parametre Testi – Fold 6 İçin En İyi 10 Sonuç

Başarı oranı	İterasyon Sayısı	Gizli Katman Sayısı	İleri ağ ve Geri ağ Pencere Genişliği	İleri ağ ve Geri ağ Çıkış Sayısı	İleri ağ ve Geri ağ Gizli Katman Düğüm Sayısı	Öğrenme Hızı
81,946	1500	10	8	8	9	0,4
81,9337	1000	10	8	8	9	0,6
81,9337	2500	30	8	8	9	0,2
81,9337	3000	30	8	8	9	0,2
81,9213	1000	20	8	8	9	0,6
81,9213	1500	30	8	8	9	0,8
81,909	2000	30	8	8	9	0,4
81,8967	1000	10	8	8	9	0,8
81,8843	3000	10	8	8	9	0,2
81,872	3500	30	8	8	9	0,2

Çizelge 3.14. BRNN Parametre Testi – Fold 7 İçin En İyi 10 Sonuç

Başarı oranı	İterasyon Sayısı	Gizli Katman Sayısı	İleri ağ ve Geri ağ Pencere Genişliği	İleri ağ ve Geri ağ Çıkış Sayısı	İleri ağ ve Geri ağ Gizli Katman Düğüm Sayısı	Öğrenme Hızı
82,8643	5000	30	8	8	9	0,6
82,8643	6000	30	8	8	9	0,4
82,8523	5500	20	8	8	9	0,6
82,8523	6000	20	8	8	9	0,6
82,8404	5500	30	8	8	9	0,4
82,8404	6000	30	8	8	9	0,2
82,8284	6000	30	8	8	9	0,6
82,8045	5000	30	8	8	9	0,8
82,8045	5500	30	8	8	9	0,6
82,7925	2500	5	8	8	9	0,6

4. SONUÇ

Bu tezde protein ikincil yapı tahmini deneyleri çok katmanlı yapay sinir ağı ve çift yönlü tekrarlayan yapay sinir ağı kullanılarak yapılmıştır. Her iki ağın parametre optimizayonu yapıldığında çift yönlü tekrarlayan yapay sinir ağlarının daha iyi bir başarım sağladığı görülmüştür. Tekrarlayan çift yönlü yapay sinir ağlarında, ağın geçmiş ve gelecek bilgiye aynı anda erişim sağlamasının nedeninin, proteinlerin ikincil yapılarının belirlenmesinde amino asit diziliminde her bir amino asitin kendinden önce gelen ve kendinden sonra gelen aminoasitlerin hangileri olduğunun belirleyici olduğu düşünülebilir.

KAYNAKLAR

[1] E. Frank, M. Hall, L. Trigg, G. Holmes, and I. H. Witten, "Data mining in bioinformatics using Weka," *Bioinformatics*, vol. 20, no. 15, pp. 2479–2481, 2004.

[2] "RCSB Protein Data Bank." [Çevrimiçi]. Adresi: http://www.rcsb.org/pdb/ngl/ngl.do?pdbid=1MBN. [Erişim Tarihi: 17-07-2018]

[3] M. G. N. M. Luscombe, D. Greenbaum, "What is Bioinformatics? A Proposed Definition and Overview of the Field," pp. 346–358, 2001.

[4] "Amino acid." [Çevrimiçi]. Adresi: https://simple.wikipedia.org/wiki/Amino_acid. [Erişim Tarihi: 17-07-2018].

[5] B. Al-Lazikani, J. Jung, Z. Xiang, and B. Honig, "Protein structure prediction," *Curr. Opin. Chem. Biol.*, vol. 5, no. 1, pp. 51–56, 2001.

[6] P. Baldi and G. Pollastri, "The Principled Design of Large-Scale Recursive Neural Ağ Architectures-DAG-RNNs and the Protein Structure Prediction Problem," *J. Mach. Learn. Res.*, vol. 4, pp. 575–602, 2003.

[7] L. H. Holley and M. Karplus, "Protein secondary structure prediction with a neural network," vol. 86, no. January, pp. 152–156, 1989.

[8] G. Pollastri, P. Baldi, P. Fariselli, and R. Casadio, "Improved prediction of the number of residue contacts in proteins by recurrent neural networks," *Bioinformatics*, vol. 17, no. SUPPL. 1:S234-42, 2001.

[9] B. Rost and C. Sander, "Combining evolutionary information and neural networks to predict protein secondary structure," *Proteins Struct. Funct. Bioinforma.*, vol. 19, no. 1, pp. 55–72, 1994.

[10] C. Bishop, "A Fast Procedure For Retaining Multilayer Perceptron," *Int. J. Neural Syst.*, vol. 2, no. 3, pp. 229–236, 1991.

[11] E. Kiliç, "Lineer Olmayan Dinamik Sistemlerin Yapay Sinir A ğ ları ile Modellenmesinde Comparison of MLP and RBF Structures in Modeling of Nonlinear Dynamic Systems with Artificial Neural Networks," pp. 694–698, 2012.

[12] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, 1997.

[13] T. Sterling, D. J. Becker, C. Park, J. E. Dorband, U. A. Ranawake, and C. V Packer, “BEOWULF : A PARALLEL WORKSTATION FOR SCIENTIFIC COMPUTATION,”, In Proceedings of the 24th International Conference on Parallel Processing, 1995, pp. 11–14.

[14] “Lustre dosya sistemi.” [Çevrimiçi]. Adresi: http://lustre.org. [Erişim Tarihi: 27-07-2018].

[15] “Slurm workload manager.” [Çevrimiçi]. Adresi: http://slurm.schedmd.com. [Erişim Tarihi: 27-07-2018].

[16] “Proxmox Server Solutions.” [Çevrimiçi]. Adresi: http://proxmox.com. [Erişim Tarihi: 27-07-2018].