KISA BİYODİZİLİMLERİN SINIFLANDIRILMASI

CLASSIFICATION OF SHORT BIOSEQUENCES

ALPER TUNGA KALKAN

Başkent Üniversitesi
Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliğinin
BİLGİSAYAR Mühendisliği Anabilim Dalı İçin Öngördüğü
YÜKSEK LİSANS TEZİ
olarak hazırlanmıştır.

2012

“Kısa Biyodizilimlerin Sınıflandırılması” başlıklı bu çalışma, jürimiz tarafından, …/…./…... tarihinde, **BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI'nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Başkan	:
	Doç. Dr. Hamit ERDEM
Üye (Danışman)	:
	Doç. Dr. Hasan Oğul
Üye	:
	Yrd. Doç. Dr. Emre Sümer

ONAY

..../..../.....

Prof. Dr. Emin AKATA

Fen Bilimleri Enstitüsü Müdürü

TEŞEKKÜR

Sayın Doç. Dr. Hasan Oğul'a her zaman yardımcı ve yol gösterici olduğu için…

KISA BİYODİZİLİMLERİN SINIFLANDIRILMASI

ÖZ

Dizilim sınıflandırma biyobilişimin en temel problemlerinden bir tanesidir. Ne olduğu bilinmeyen bir moleküler birimi sadece bu birimin dizilim verisi kullanılarak, daha önceden bilinen bir sınıfa atamak için birçok araç ortaya çıkmıştır. Fakat çıkan bu araçların hepsi ilgili olduğu problemlere özeldir. Ayrıca bu araçlar problemin ait olduğu biyolojik dizilim alfabesine bağlıdır. Bu tezde alfabeden bağımsız yeni bir genel dizilim sınıflandırıcı (TRAINER) java programla dili kullanılarak gerçekleştirilmiştir. Bu araç ile kullanıcılar kendi eğitim veri setleri ve kendi dizilim alfabeleri ile dizilim sınıflandırması yapabilecektir. TRAINER kullanıcıların sistemde tanımlı öğreticiyle (supervised) öğrenme yöntemlerinden istediğini, yöntemin parametrelerini ve önceden tanımlı çeşitli özellik belirtme kalıplarından birini seçerek kullanmasını sağlar. Sistemde eğitilmiş modeller kullanıcının isteğine bağlı olarak sisteme ileride tekrar eğitilmeden kullanılmak üzere kayıt edilebilir. Aday efektör tahmini, mikroRNA hedef tahmini ve nükleolar konumlanma sinyal tahmini çalışmaları ile sistemin DNA, RNA ve protein dizilimleri için verimli bir şekilde çalıştığı gösterilmiştir. Ortaya çıkan sonuçların biyolojik manaları tezde tartışılmıştır.

ANAHTAR SÖZCÜKLER: Dizilim Sınıflandırma, Web Sunucu, En yakın K komşu (k-nearest neighbors) , naive Bayes sınıflandırıcı, destek vektör makinaları (Support Vector Machine).

Danışman: Doç.Dr. Hasan OĞUL, Başkent Üniversitesi, Bilgisayar Mühendisliği Bölümü.

CLASSIFICATION OF SHORT BIOSEQUENCES

ABSTRACT

Classifying sequences is one of the central problems in computational biosciences. Several tools have been released to map an unknown molecular entity to one of the known classes using solely its sequence data. However, all of the existing tools are problem-specific and restricted to an alphabet constrained by relevant biological structure. Here, we introduce TRAINER, a new online tool designed to serve as a generic sequence classification platform to enable users provide their own training data with any alphabet therein defined. TRAINER is implemented by using java programming language. TRAINER allows users to select among several feature representation schemes and supervised machine learning methods with relevant parameters. Trained models can be saved for future use without retraining by other users. Three case studies are reported for effective use of the system for DNA, RNA and protein sequences; Candidate effector prediction, microRNA target prediction and nucleolar localization signal prediction. Biological relevance of the results is discussed.

KEYWORDS: Sequence classification, Web server, K-nearest Neighbors, Naive Bayes Classifier, Support Vector Machine.

Advisor: Assoc. Prof. Dr. Hasan OĞUL, Başkent University, Department of Computer Engineering.

İÇİNDEKİLER LİSTESİ

ŞEKİLLER LİSTESİ

ÇİZELGELER LİSTESİ

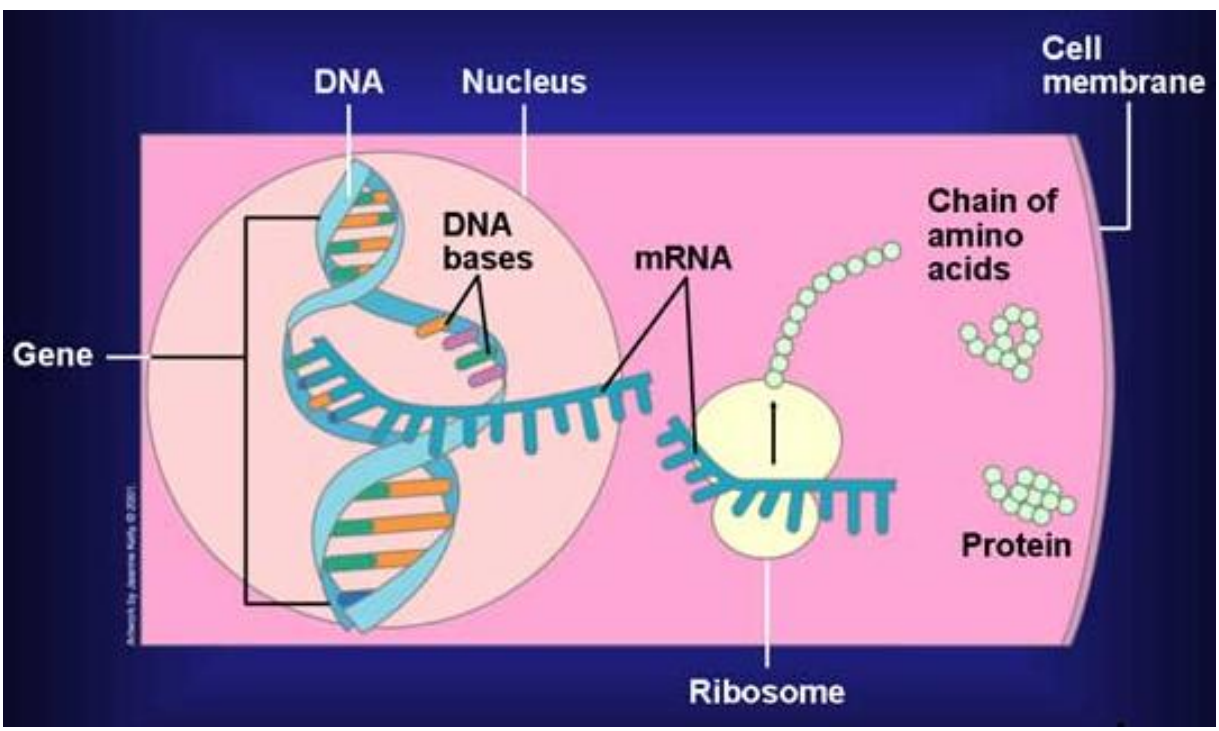
1. GİRİŞ

Hayatımızın sırları DNA, RNA ve protein gibi moleküler yapılara kodlanmıştır (Şekil 1-1). Bu moleküllerden DNA ve RNA nın birlikte çalışması sonucu ortaya çıkan proteinler hayatımızı sürdürmemiz için olmazsa olmaz olan moleküllerdir. Birçok hastalığın temelinde protein sentezinde yaşanan problemler vardır. Protein sentezi en basit hali ile şekil 1-2 de gösterilmiştir. DNA, RNA ve protein gibi moleküllerin yapısı ve fonksiyonları iyi tanımlamış dizilim bloklarına bağlı olduğu için bu dizilimlerde bulunan bilgiyi ortaya çıkarma konusu bilimde çok uzun yıllardır popüler bir konu olmuştur. Bilgisayar yardımı ile yapılan dizilim analiz teknikleri dizilimlerden gerekli bilgileri elde etme noktasında çok değerli katkılar sağlamıştır. Bu teknikler çoğu zaman sadece dizilimin kendisini girdi olarak alıp, dizilim içindeki biyolojik olayları açıklayabilecek desenleri bulmayı amaçlar.

Şekil 1-1 DNA sarmalı

Dizilim analizi alanının en fazla rastlanılan uygulamalarından biri dizilim sınıflandırılmasıdır. Dizilim sınıflandırılması bir dizilimin öncenden bilinen sınıflardan (kategoriler) birine atanması olarak tanımlanabilir. Bu işlem dizilimin biyolojik manası hakkında bilgi verebilir.

Dizilim sınıflandırma problemi ile RNA, DNA, protein ve hatta bir alfabe ile ifade edilebilen sinyallerin sınıflandırması gibi birçok ortamda karşılaşılabilir. Birçok araştırmacı bu problem ile ilgilenmiş ve çeşitli yöntemler geliştirmiştir. [1]. Bazı yöntemler çevrim içi çalışan bazıları ise yalnız çalışan biyolojik uygulamalar olarak gerçekleştirilmiştir. Bu uygulamaların hepsi kullanıcıya kendi elindeki dizilim verisini daha önceden tanımlanmış olan sınıflardan biri ile eşleştirmesi için bir ara yüz sunar.

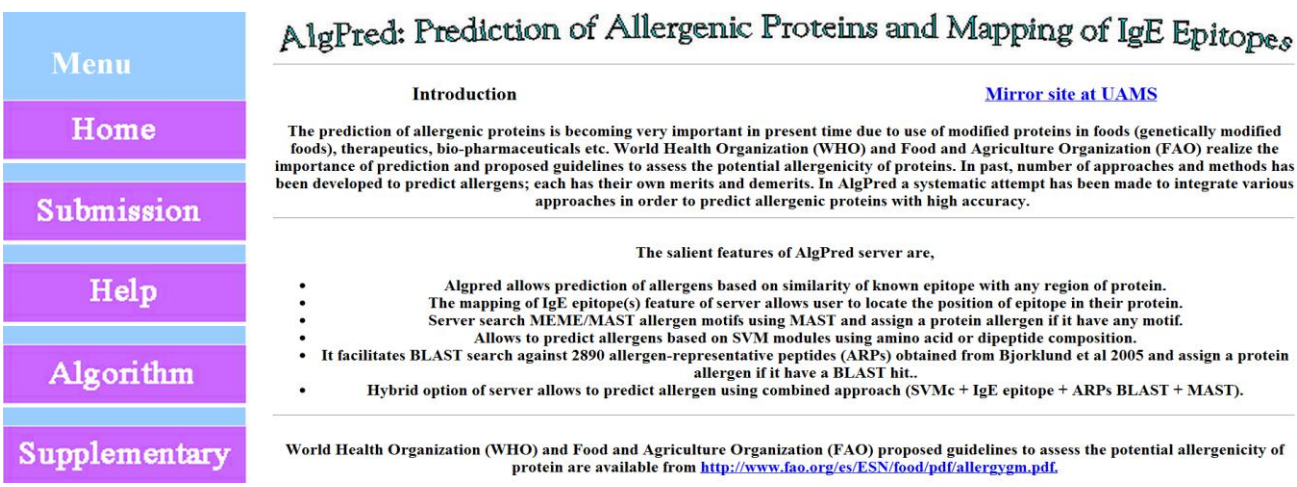


Şekil 1-2 DNA dan RNA, RNA dan protein sentezi

Örnek olarak şekil 1-3'te ana sayfası görülen TargetP protein dizilimlerinden proteinin hücre içi konumlarını tahmin eden bir web sunucudur [2]. Bu sunucu konumları bilinen eukaryotic proteinler için önceden eğitilmiştir.

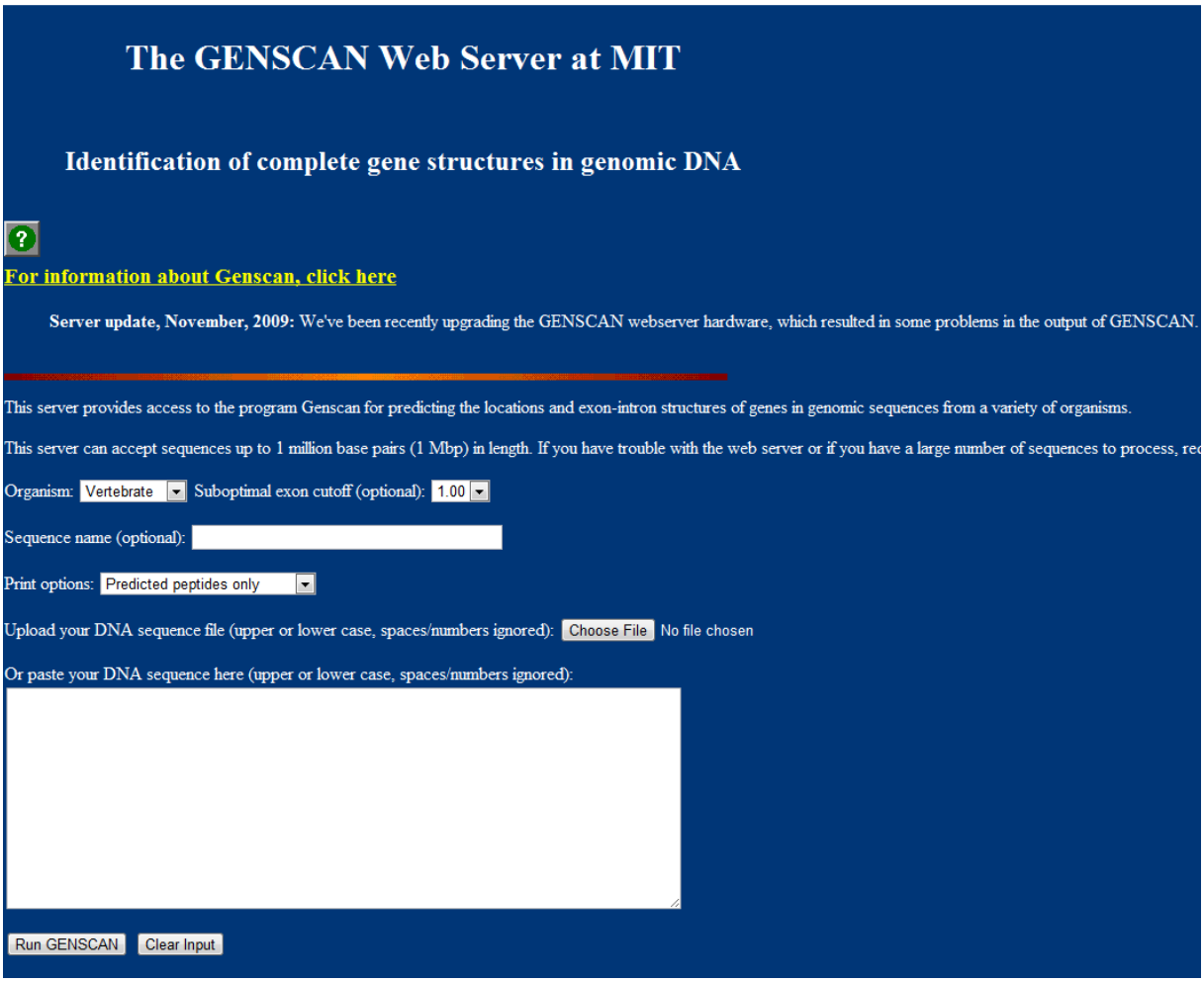
Şekil 1-3 TargetP web aracı ana sayfası

Şekil 1-4 de ana sayfası görülen AlgPred ise proteinlerin alerjen olup olmamasına göre sınıflandıran diğer bir çevrim içi araçtır [3].



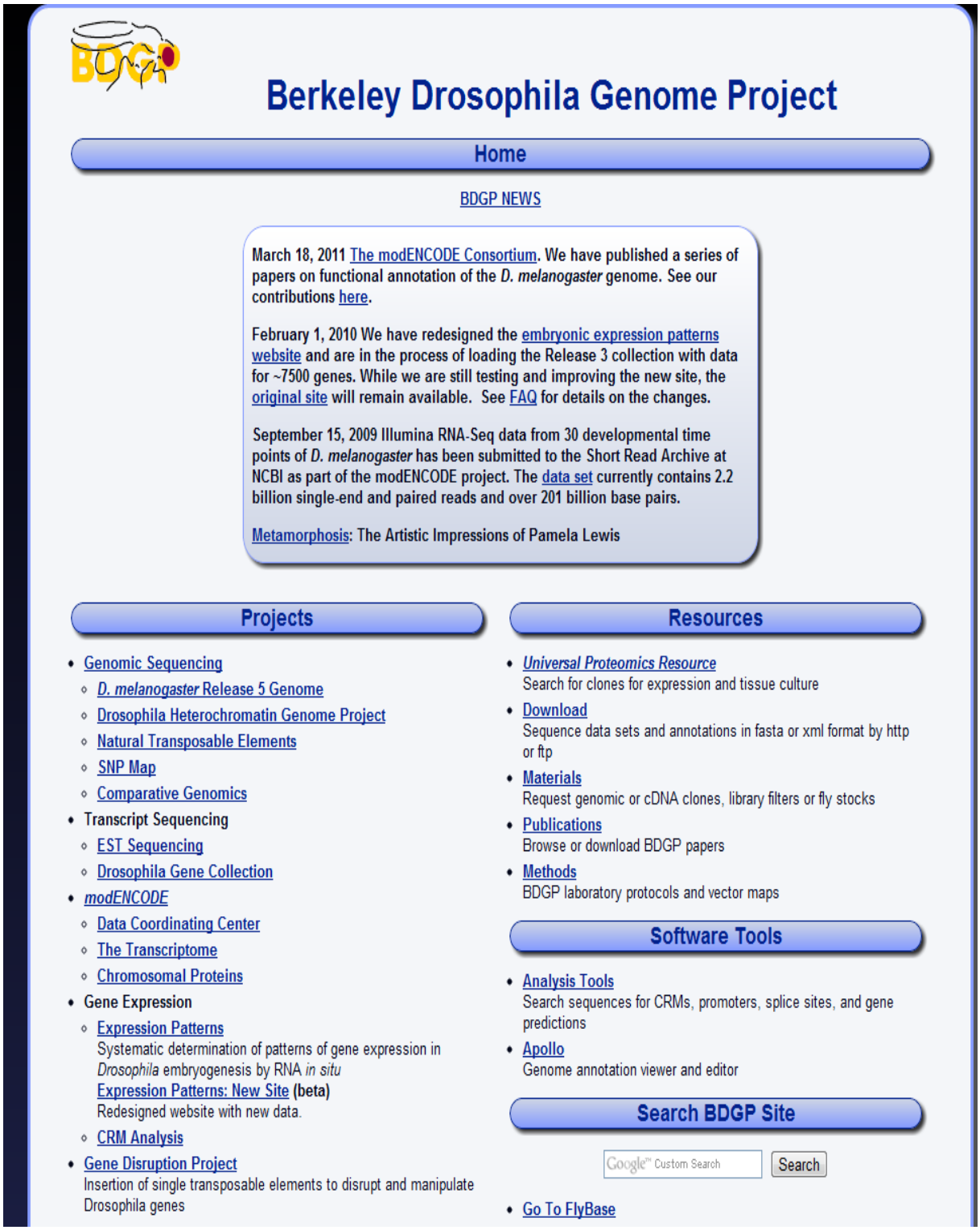
Şekil 1-4 AlgPred web aracı ana sayfası

Şekil 1-5 de ana sayfası görülen Genscan çeşitli organizmalardaki nükleotit dizilimlerini exon veya intron olarak sınıflandıran bir araçtır [6].



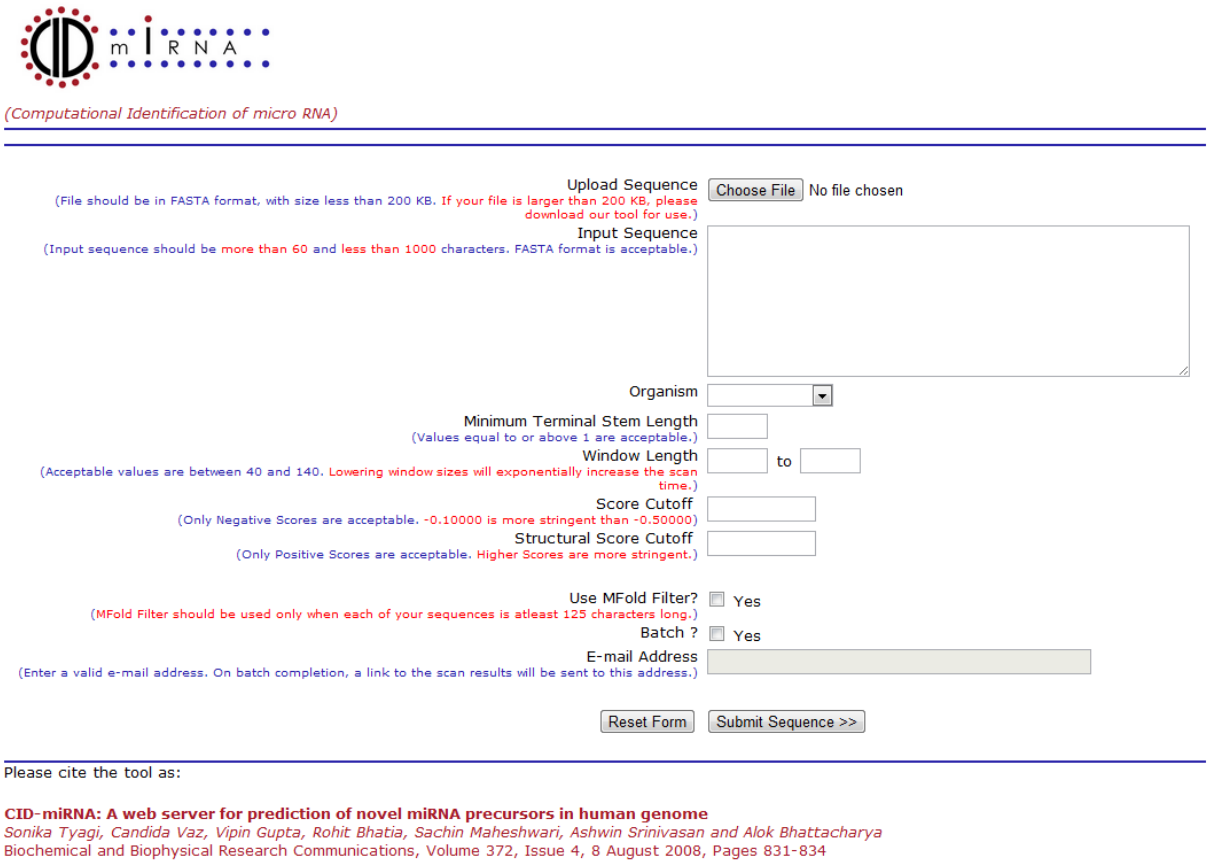
Şekil 1-5 GENSCAN web aracı ana sayfası

Şekil 1-6 da ana sayfası görülen BDGP sunucusu DNA dizilimlerinden muhtemel transkripsiyon promoterları bulmaya olanak veren diğer bir ortamdır [4].



Şekil 1-6 BDGP web aracı ana sayfası

Şekil 1-7 de ana sayfası görülen CID-miRNA insan genomundaki mikroRNAları bulmaya yarayan diğer bir çevrimiçi araçtır [5]. Bu araç girdi olarak verilen RNA dizilimlerini mikroRNA olup olmadığına göre sınıflandırabilir.



Şekil 1-7 CID-miRNA web aracı ana sayfası

Bütün bu çeşitli araçlara rağmen alfabesi ve sınıfları kullanıcı tarafından belirtilen bütün dizilimler için çalışacak genel sınıflandırıcı yoktur.

Biobilimciler çoğu zaman yeni dizilimler üzerinde çalışmak durumunda oldukları için ve şu an sahip olunan araçlar problem veya alfabe özel olduğu için her türlü dizilimle çalışacak ve kullanıcı tarafından eğitilecek genel bir dizilim sınıflandırıcıya ihtiyaç vardır.

Bu tezde bu ihtiyacı kapatacak bir araç geliştirilmeye çalışmıştır. Geliştirilen araç üç temel ve yaygın kullanılan makina öğrenme algoritmasını çok yaygın olarak kabul edilen özellik seçme şemaları ile beraber gerçekleştirmiştir.

Sistemin adı TRAINER dır. TRAINER temelde iki moda sahiptir: (1) Eğitim Modu (2) Yeniden Yükleme Modu. İlk modda dizilimlerin alfabesi, eğitim verisi ve sınıflar kullanıcı tarafından sisteme verilir. İkinci modda ise kullanıcı kendi dizilimlerine uygun olan ve daha önceden eğitilip sisteme kayıt edilmiş olan modeller arasından bir seçim yaparak, elindeki dizilim verisini sınıflandırır.

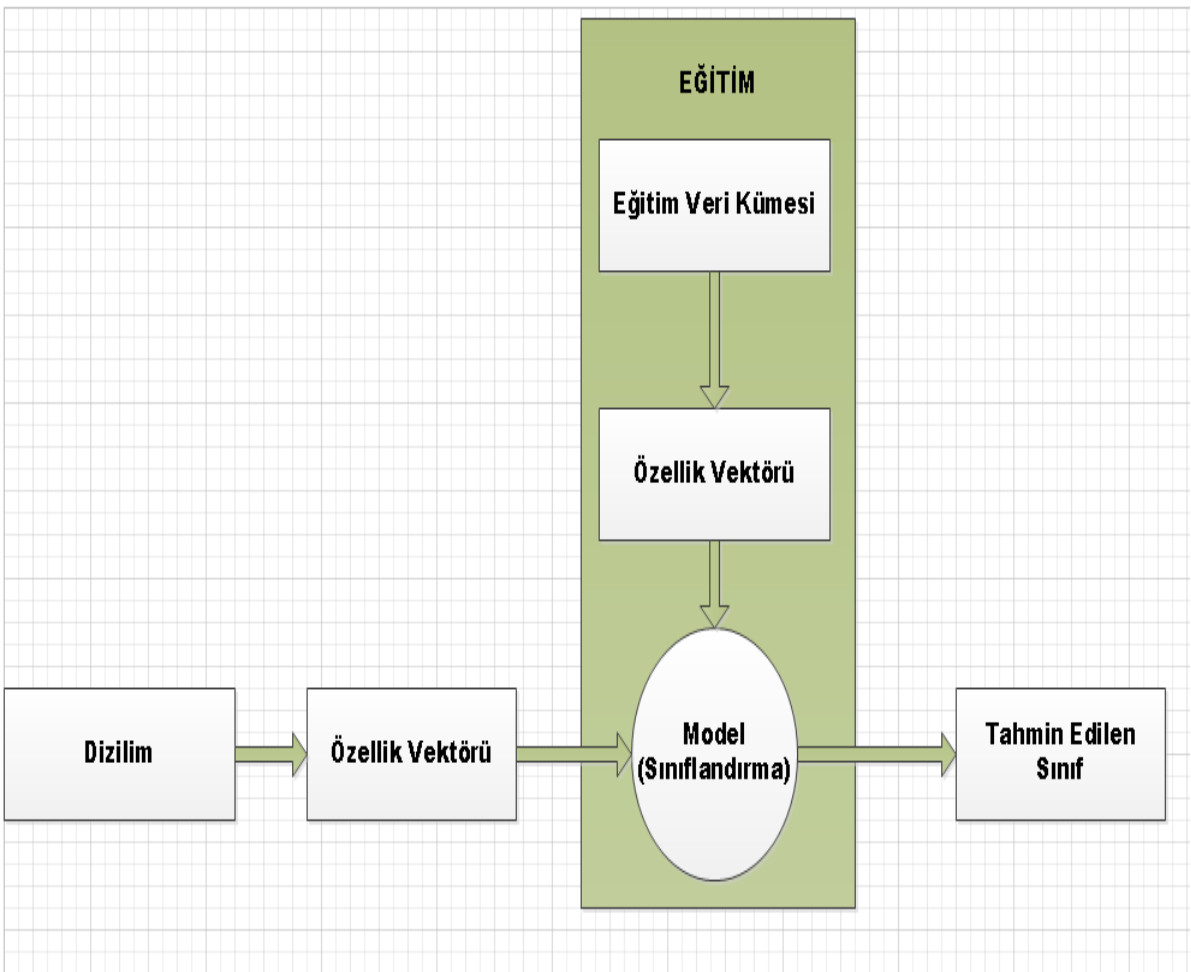
Sistem test amacı ile alfabeleri farklı olan üç örnek çalışma ile denenmiştir. İlk çalışmada nükleotit dizilimlerinin sistem yardımı ile aday efektör olup olmadığı test edilmiştir. İkinci örnek is mikroRNA hedef tahmini probleminin bir alt problemidir. Bu örnekte RNA-RNA çifti yapısının sözde alfabesi kullanılmıştır. Son uygulamada ise amino asit alfabesi kullanılmıştır. Bu uygulamada protein dizilimleri sistemi girdi olarak verilerek onun nükleolar konumları sorgulanmıştır.

Tezde bu üç örnek için sistemin performansının yanında önemli bulgularda belirtilmiştir. Tez raporu beş bölümden oluşmaktadır. Birinci bölümde tezin motivasyonu, temel bilgiler ve benzer çalışmalar tanıtılmıştır. İkinci bölümde TRAINER'da sunulan makine öğrenme yöntemleri, özellik temsil yöntemleri ve web sunucun altyapısı hakkında bilgiler sunulmuştur. Üçüncü bölümde gerçekleştirilen çevrim içi araç olan TRAINER tanıtılmıştır. Dördüncü bölümde örnek çalışmalar ve TRAINER'in bu çalışmalardaki performansı sunulmuştur. Beşinci ve son bölüm olan sonuç bölümünde tezin özeti ve örnek çalışmaların sonuçları hakkında yorumlar sunulmuştur.

2. YÖNTEMLER

2.1 Sınıflandırma

TRAINER çok yaygın kullanılan üç makina öğrenme algoritmasından oluşur. Bunlar naive Bayes, destek vektör makinaları, en yakın k komşu algoritmalarıdır. Bu algoritmaların her biri girdi olarak ait olduğu dizilimi temsil eden bir özellik vektörü alır. Girdi olarak alınan vektörü daha önceden tanımlamış bir sınıfa atamaya çalışır. (Şekil 2-1) Algoritmalar atama işlemlerini belirli bir modele göre yapmaktadır. Bu model algoritmaların atama işlemlerini yapmadan önce öğrenme veri setinden algoritma tarafından çıkarılmaktadır. Öğrenme veri seti problemde belirtilen ve sınıfları bilinen veriler topluluğudur.



Şekil 2-1 Genel sınıflandırma şeması

2.1.1 Naive Bayes sınıflandırıcı

NaiveBayes sınıflandırıcı bayes istatistiğine dayanan bir öğreticiyle öğrenme (supervised) sınıflandırma tekniğidir. Bayesian istatistiği veri seti altında yatan olasılıksal bir modelin varlığını varsayar. Bu model ortaya çıkabilecek sonuçlara göre model hakkında belirsizliği ortaya koyar.

Problemdeki özellik vektörü elemanlarının (feature) gerçekte birbiri ile ilişkisi olmasına rağmen bu elemanların birbirinden bağımsız olduğunu kabul eder. Böylece her elemanın problemin çözümüne diğer elemanlardan bağımsız olarak katkı sağladığını farz eder. İsmindeki Naive sıfatını bu kabulden dolayı almıştır. Yöntemdeki bu "naive" varsayıma rağmen, çoğu gerçek hayat problemlerinde performansı yüksektir.

Naive bayes modellerde parametre tahminini en yüksek olasılık (maximum likehood) kullanılarak yapılır. Girdinin boyutu yüksek olduğu zamanlarda bu yöntem iyi bir seçimdir. Çok az sayıda veri içeren bir eğitim veri seti ile parametre tahmini yapılabilir. Naive Bayes sınıflandırıcının çıktısı, girdi olarak verilen verinin, problemdeki sınıflara göre o sınıfta olma olasılığıdır. Veri en yüksek olasılığı sahip sınıfa atanır.

Matematiksel tanımı şöyle yapılabilir [27].

P(A) : A olayın B olayı hakkında hiçbir bilgi sahibi olmadan olma olasılığı

P(B) : B olayın A olayı hakkında hiçbir bilgi sahibi olmadan olma olasılığı

P(A | B) : A' nın B olayı verildiği zamanki koşullu olasılığı

P(B | A) : B' nın A olayı verildiği zamanki koşullu olasılığı

$$P(A \mid B) = \frac{P(B \mid A)\,P(A)}{P(B)} \tag{2.1}$$

Bir örnek üzerinde naive Bayes sınıflandırıcının nasıl çalıştığını aşağıda açıklanmıştır.

Görünüş	Sıcaklık	Nem	Rüzgârlı	Oyun oynanabilir
Güneşli	Sıcak	Yüksek	Hayır	Hayır
Güneşli	Sıcak	Yüksek	Evet	Hayır
Kapalı	Sıcak	Yüksek	Hayır	Evet
Yağışlı	Orta	Yüksek	Hayır	Evet
Yağışlı	Soğuk	Normal	Hayır	Evet
Yağışlı	Soğuk	Normal	Evet	Hayır
Kapalı	Soğuk	Normal	Evet	Evet
Güneşli	Orta	Yüksek	Hayır	Hayır
Güneşli	Soğuk	Normal	Hayır	Evet
Yağışlı	Orta	Normal	Hayır	Evet
Güneşli	Orta	Normal	Evet	Evet
Kapalı	Orta	Yüksek	Evet	Evet
Kapalı	Sıcak	Normal	Hayır	Evet
Yağışlı	Orta	Yüksek	Evet	Hayır
Güneşli	Soğuk	Yüksek	Hayır	???

Çizelge 2-1 Örnek veri seti

Çizelge 2-1 deki son veri için sınıflandırma olasılıkları aşağıdaki gibi bulunabilir. İlk olarak verilen özelliklerden bağımsız olarak Hiçbir bilgi olmadan oyun oynanabilirlik ihtimalleri (prior probabilities) aşağıdaki gibi olur.

$$P(Evet) = P(C_1) = \frac{9}{14} \quad (2.2)$$

$$P(Hayır) = P(C_2) = \frac{5}{14}$$

Bütün özellikler birbirinden bağımsız düşünüldüğü hesaba katılırsa özellik vektörü için formül aşağıdaki gibi belirtilebilir.

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i) \quad (2.3)$$

Bu ifade oyun oynanma durumları olan "evet" ve "hayır"a göre açılırsa 2.4 ve 2.5 deki ifadeler ile karşılaşılır.

$$P(X = (Güneşli, Soğuk, Yüksek, Hayır) \mid Evet) =$$

$$P(x_2 = Güneşli \mid Evet) * P(x_3 = Yüksek \mid Evet) *$$

$$P(x_4 = Hayır \mid Evet) * P(x_1 = Hayır \mid Evet) \quad (2.4)$$

$$= \frac{1}{9} * \frac{2}{9} * \frac{4}{9} * \frac{3}{9}$$

$$= \frac{24}{6561}$$

$$P(X = (Güneşli, Soğuk, Yüksek, Hayır) \mid Hayır) =$$

$$P(x_2 = Güneşli \mid Hayır) * P(x_3 = Yüksek \mid Hayır) *$$

$$P(x_4 = Hayır \mid Hayır) * P(x_1 = Hayır \mid Hayır) \quad (2.5)$$

$$= \frac{3}{5} * \frac{2}{5} * \frac{4}{5} * \frac{2}{5}$$

$$= \frac{48}{625}$$

2.5 ve 2.4 deki ifadelerin sonuçları 2.1 formülünde ayrı ayrı yerine koyulursa 2.6 ve 2.7 deki sonuçlar elde edilir.

$$P(X = (Güneşli, Soğuk, Yüksek, Hayır) \mid Evet) \quad (2.6)$$

$$= \frac{24}{6561} * \frac{9}{14}$$

$$= 0{,}002351$$

$$P(X = (Güneşli, Soğuk, Yüksek, Hayır) \mid Hayır) \qquad (2.7)$$

$$= \frac{48}{625} * \frac{5}{14}$$

$$= 0{,}027428$$

Yukarıdaki değerlerden 2.7 deki büyük olduğu için naive Bayes sınıflandırıcı ile sınıflandırılan veri “Hayır” olarak sınıflandırılır.

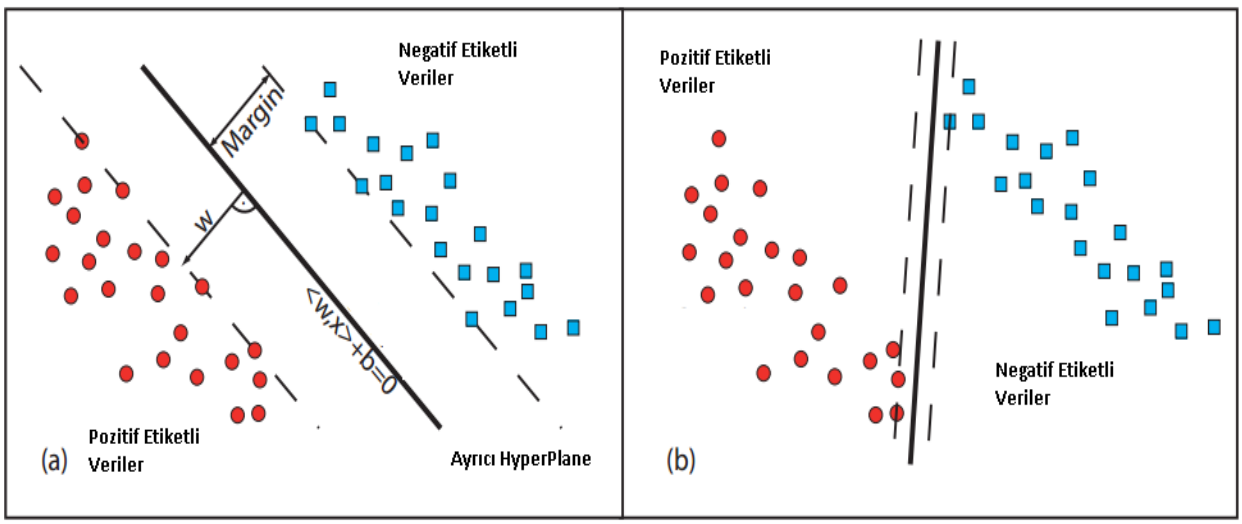
2.1.2 Destek vektör makinaları

Destek vektör makinaları (SVM) birçok araştırmacı tarafından hali hazırdaki en sofistike sınıflandırıcı olarak nitelenmiştir. Örneğin meme kanseri sınıflandırılması için birçok teknik öne sürülmüştür. Mamografi meme kanserini tespit etmekte en çok kullanılan yöntemlerden bir tanesidir. Fakat sonuçlar uzmanlar tarafından yorumlanırken birçok kez farklı sonuçlar ortaya çıkmıştır. Bu sebepten dolayı mamografi sonuçlarını yorumlayacak otomatik bir sisteme yani sınıflandırıcıya ihtiyaç duyulmuştur. Bu sınıflandırıcı için birçok farklı yöntem denenmesine rağmen SVM genel görüş olarak diğer bütün yöntemlerden daha iyi sonuç vermiştir. [17]

Destek vektör makinaları iki sınıf arasında tahmin yapan bir sınıflandırıcıdır. Bu sınıflandırıcı yapısal riskleri en aza indirme prensibine göre çalışır. SVM n boyutlu girdi verisini lineer olmayan bir şekilde daha yüksek bir boyuta taşır.

Taşıdığı bu yüksek boyutta lineer bir sınıflandırıcı oluşturur. SVM yöntemi sınıflandırma aşamasına geldiği zaman, sınıflandırılacak dizilim için eğitim safhasındaki gibi dizilimi temsil eden bir özellik vektörüne ihtiyaç duyar. Bu özellik vektörü test veri setindeki her bir veri için ayrı ayrı oluşturulmalıdır.

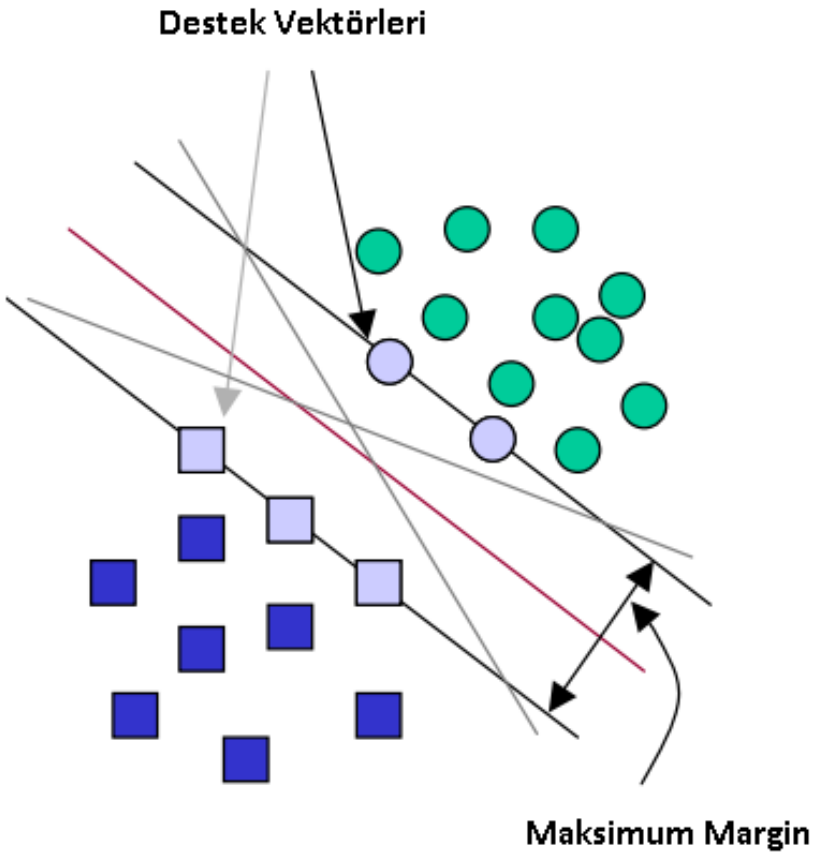
Tanım olarak SVM verilen iki veri kümesi arasında veriyi bir birinden ayıran optimum hiper düzlemi bulur. Aşağıda şekil 2-2 b de veri kümesi iki boyutlu olduğu için verileri ayıran hiper düzlem çizgidir. Verileri ayıran bir çok çizgi olmasına rağmen a'daki ayırım göz ile de fark edilebileceği gibi en optimum ayırımdır.



Şekil 2-2 İki farklı hiper düzlem

Yine şekilde görüldüğü gibi veriyi birbirinden ayıran birçok düzlem olmasına rağmen SVM nin amacı şekil 2-3 deki gibi maksimum margin ile ayırım yaparak destek vektörlerini bulmaktır.

SVM çıktısı test edilecek veriye ait discriminant skorudur. İki sınıf arasında sınıflandırma yapan sınıflandırıcılarda pozitif skor verinin o sınıfa ait olduğuna işaret eder. Sistemde sıfırdan büyük değerler iyi bir skor olarak kabul edilmiştir. Radial temelli kernel fonksiyonu tez kapsamında yapılan örnek çalışmalarda kullanılmak üzere LIBSVM [7] paketindeki varsayılan parametreleri ile birlikte gerçekleştirilmiştir. LIBSVM herkesin kullanıma açık bir SVM paketidir.



Şekil 2-3 Destek Vektörleri

Matematiksel olarak SVM aşağıdaki gibi tanımlar.

$\{x_i, y_i\}$, i = 1,…N, olarak verilen eğitim setinde her örnek d tane özelliğe sahiptir ($x_i \in R^d$). Sınıfları belirten y_i sadece iki değer olabilir ($y_i \in \{1,-1\}$). D boyutlu bu uzayda bütün hiper düzlemler bir vektör ve bir sabit sayı ile belirtilir. Bu ifadeyi aşağıdaki gibi belirtebiliriz. [22]

$$w * x + b = 0 \tag{2.8}$$

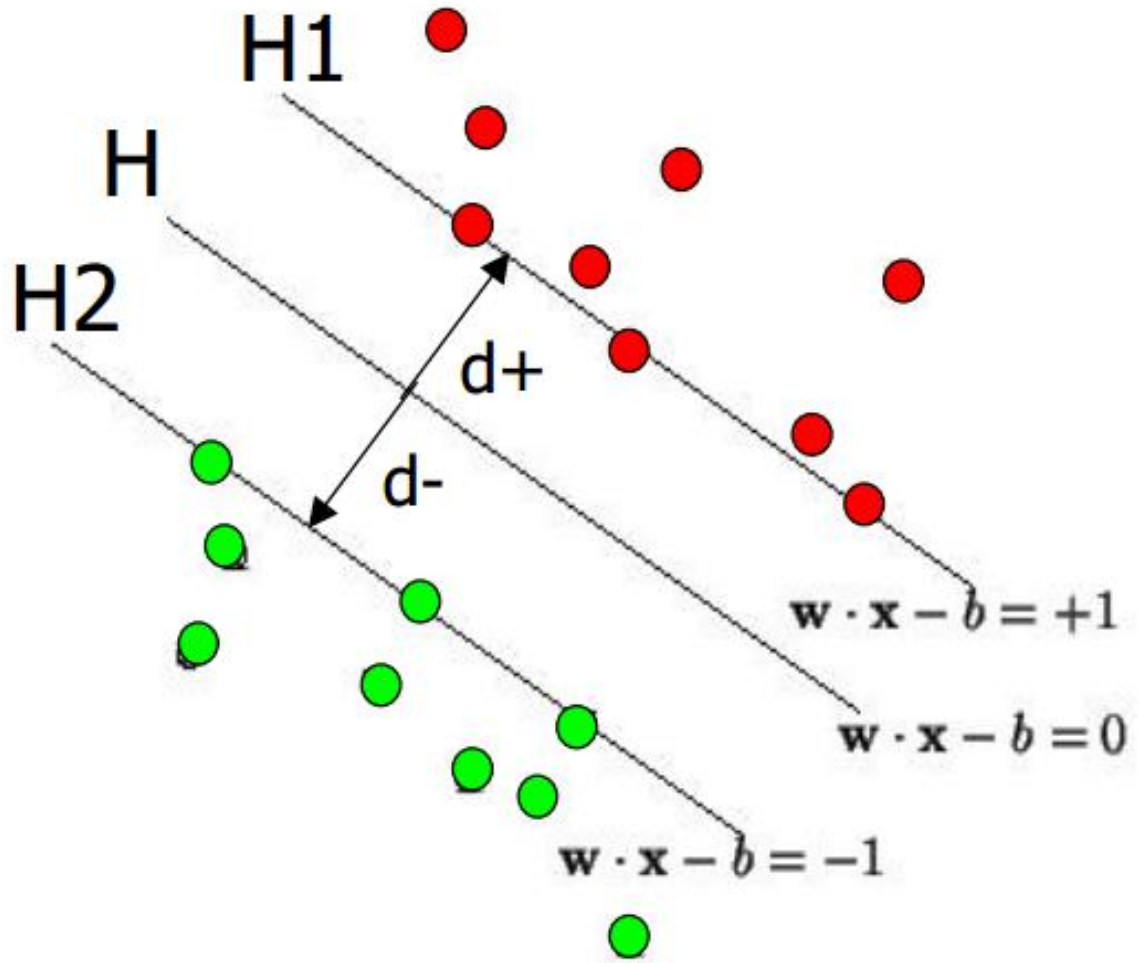
Not olarak w vektörü hiper düzleme dik bir vektördür. Bu formülü SVM nin sınıflandırma fonksiyonu olarak kullanırsak aşağıdaki formüle ulaşırız

$$f(x) = sign(w * x + b) \tag{2.9}$$

Veri setinde bulunan herhangi örnek olan x_i formülde yerine koyulursa şekil 2-4 de görüldüğü üzere şöyle bir sonuç ortaya çıkar.

$$x_i * w + b \geq +1 \text{ İse } y_i = +1 \tag{2.10}$$

$$x_i * w + b \leq -1 \text{ İse } y_i = -1 \quad (2.11)$$



Şekil 2-4 Geometrik olarak hiper düzlem

Veya daha sade olarak:

$$y_i(x_i * w + b) \geq 1 \quad (2.12)$$

İfadesi veri setindeki her örnek içi doğru olur.

Geometrik olarak x_i noktasının hiper düzleme olan uzaklığı hesaplanırken w nin değerini normalize edilmelidir. Böylece x_i noktasının hiper düzleme olan uzaklığı basitçe aşağıdaki gibi formülüze edilebilir.

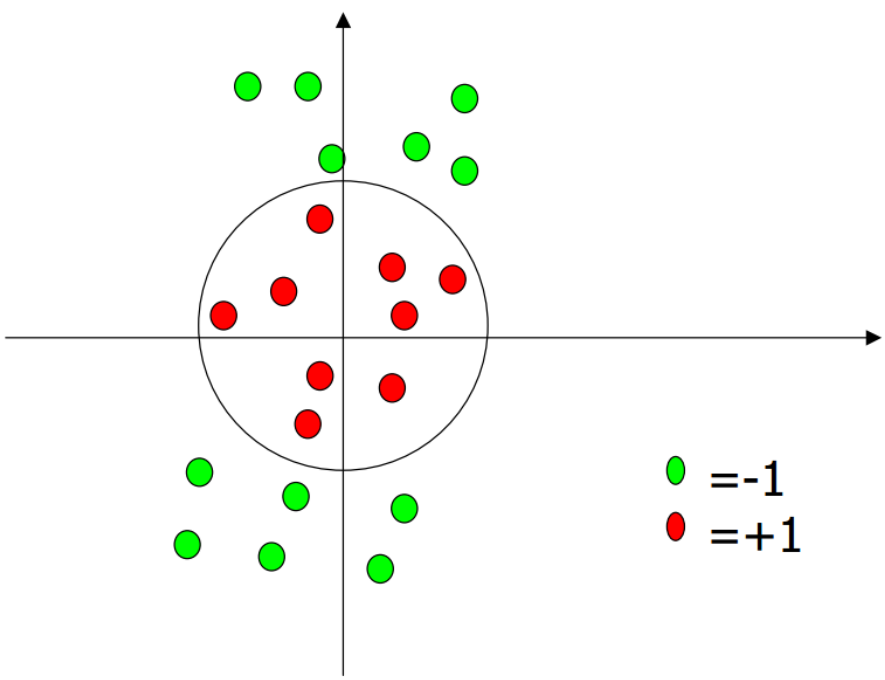
$$d\big((w,b),x_i\big) = \frac{y_i(x_i * w + b)}{||w||} \geq \frac{1}{||w||} \quad (2.13)$$

Bu noktanın hiper düzleme olan uzaklığını maksimize edilmek istendiği için yukarıdaki formüldeki ||w|| ifadesi minimize edilmesi gerekmektedir. Bu ifadenin minimize edilmesinde kullanılan başlıca yöntem Vapnik de belirtildiği gibi Lagrange çarpanlarıdır [23]. Bu yöntem kullanılarak ifade aşağıdaki ifadenin minimize edilmesine dönüştürülür.

$$W(\alpha) = -\sum_{i=1}^{N} \alpha_i + \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j (x_i^T * x_j) \quad (2.14)$$

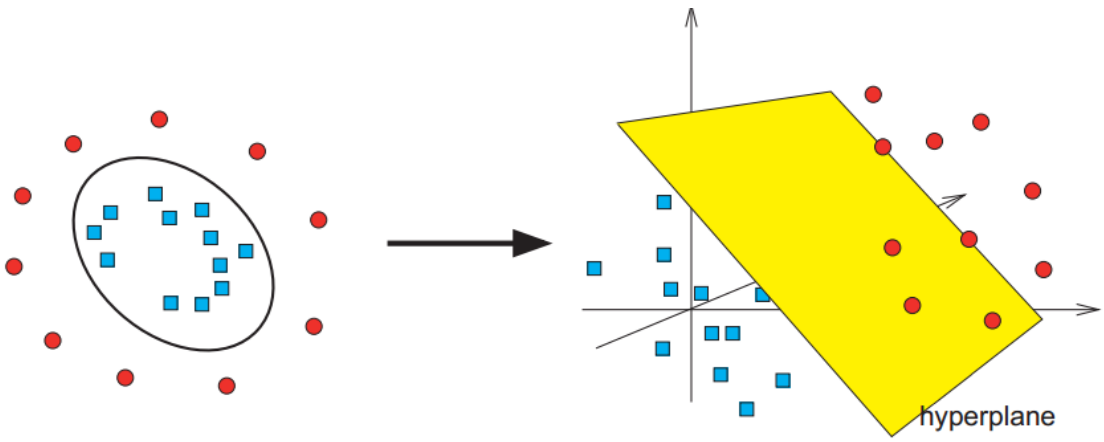
Bu ifadenin minimize edilmesi ile her veri için bir tane olmak üzere toplam L tane α değeri bulunur [25]. Bulunan alfa değerlerinden sıfırdan büyük olanlar destek vektörleri olarak tanımlanmıştır. Örnek olarak 1000 verilik bir eğtim setinde çıkan α değerlerinin birçoğu sıfır olacaktır [25]. Bu noktalar veriyi ayıran maximum margin ile tanımlamış hiper düzlemin dışında kalan noktalardır. Fakat α_i değeri sıfırdan büyük ise bu değerin ait olduğu x_i vektörü destek vektörü olarak tanımlanır. Destek vektörlerinin bulunması ile lineer olarak ayrılan veriler için maximum margine sahip hiper düzlem bulunmuş olur.

Şu ana kadar veri setinin lineer olarak ayırabilindiği farz edilmiştir. Ama karşılaşılan problemlerin birçoğunda veri setinde aşağıdaki şekil 2-5 de olduğu gibi lineer olarak ayrılamaz.



Şekil 2-5 Lineer olarak ayrılamayan veri seti

Uygun bir Φ fonksiyonu ile veri setinin lineer olarak ayrılabileceği yüksek boyutlu bir sisteme taşındığını farz edilirse, yeni oluşan çok boyutlu uzay özellik uzayı H olarak adlandırabilir. Bu uzayda bulunan bir hiper düzlem ile mevcut veriler lineer olarak ayrılacaktır [26] (Şekil 2-6)



Şekil 2-6 Veri setinin hiper düzlemde lineer olarak ayrılması

Lineer olarak ayrılamayan veriler için ulaşılan optimum hiper düzlem'in formülü, Lineer olarak ayrılabilen veriler için olan ile birebir aynıdır. Tek fark formüldeki x_i vektörlerinin d boyut olması yerine, $\Phi(x_i)$ vektörünün daha yüksek belkide sonsuz boyutta olmasıdır.

$$W(\alpha) = -\sum_{i=1}^{N} \alpha_i + \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j \left(\Phi(x_i^T) * \Phi(x_j)\right) \quad (2.15)$$

Formül incelendiği zaman görülen en önemli nokta yüksek boyutlu uzaydaki vektörlerin nokta çarpımları ile ilgilidir. Elimizdeki vektörlerin yüksek boyutlu uzaya taşınmış halindeki $\Phi(x_i^T) * \Phi(x_j)$ nokta çarpımını yüksek boyutlu uzayda yapılması çok maliyetli bir işlemdir [25]. Hatta bazı durumlarda veri sonsuz boyutlu uzaya taşındığı için bu işlemi yapmak gerçek manada imkânsızdır. Bu noktada kernel fonksiyonları verinin transfer edilmiş uzaydaki nokta çarpımlarını verirler. Kernel fonksiyonları yardımı ile verinin transfer edildiği yüksek boyutlu uzay hakkında hiçbir şey bilinmese bile bu uzaylar kullanabilir [25]. Bu durum aşağıdaki gibi formülüze edebilir.

K kernel fonksiyonu ve Φ vektörleri yüksek boyuta taşıma fonksiyonu olmak üzere

$$k(x, y) = \Phi(x) * \Phi(y) \quad (2.16)$$

Bu durumun direk bir sonucu olarak vektörleri yüksek boyuta taşıyan fonksiyon hakkında hiçbir şey bilinmese bile kernel fonksiyonları ile destek vektör makinalarını verimli bir şekilde kullanılabilir. [24]

Kernel fonksiyonları SVM nin en önemli ve anlaşılması zor konusudur. [20] Hangi Kernel fonksiyonun seçileceği probleme bağlı olarak değişir. Doğru kernel bulunsa bile bu kernel parametreleri seçme işi zor ve can sıkıcı olabilir. Kernel seçme işini otomatik olarak yapma konusunda çeşitli çalışmalar yapılmıştır.[21] Sistemde sunulmuş olan kernel fonksiyonları aşağıda incelecektir.

2.1.2.1 Lineer kernel

Lineer kernel fonksiyonları içinde en basit olanı olarak düşünülebilir. Vektörlerin iç çarpımlarına sabit bir değer ekleyerek bulunur. Verinin lineer olarak ayrılamadığı durumlarda kullanılması iyi bir seçim değildir. Aşağıdaki gibi formülüze edilebilir [18].

$$k(x, y) = x^T * y + c \tag{2.17}$$

2.1.2.2 Polynomial kernel

Eğitim veri setinde ki bütün değerlerin normalize edildiği problemler için iyi bir seçimdir. Aşağıdaki gibi formülüze edilebilir [18].

$$k(x, y) = (\alpha x^T y + c)^d, \ \alpha > o \tag{2.18}$$

2.1.2.3 Radial temelli kernel

RBF kernel çoğu zaman ilk seçim olarak kullanılır. Bu kernel datayı lineer olmayan bir şekilde yüksek boyuta taşır. Lineer kernel aksine datanın lineer olarak ayrılamadığı koşullarda verimli bir şekilde çalışabilir. Özellik vektörünün sayısının çok yüksek olduğu durumlarda kullanılması tavsiye edilmez. [18] RBF aşağıdaki gibi formülüze edilebilir [18].

$$k(x, y) = exp\left(-\alpha||x_i - x_j||^2\right), \ \alpha > o \tag{2.19}$$

2.1.2.4 Çoklu sınıflandırma ve SVM

SVM nin en belirgin problemi sadece ikili sınıflandırma (binary classification) yapmasıdır. Örnek olarak SVM iki kanser arasında sınıflandırma yapabilirken

birden fazla kanser türü arasında sınıflandırma yapamamaktadır. Bu sorunu aşmak için çeşitli yollar önerilmiştir. Bunlar içinde en basit olanı elimizde ki bütün kanser çeşitlerinden her seferde birini dışarıda tutarak bu geri kalan bütün türleri aynı tutmaktır. Böylece her bir seferde elimizde verileri ayrı tutulan kanser türü olup olmadığına göre sınıflandırabiliriz. Bu yöntemden hariç birçok daha karmaşık yaklaşımlar ile SVM yi birden fazla sınıfa göre sınıflandırma yapacak şekle dönüştürebiliriz. [19]

2.1.3 En yakın k komşu

En yakın k komşu (kNN) parametrik olmayan veri temelli bir sınıflandırıcıdır. Bu sınıflandırıcı veriyi kendine en fazla benzeyen k tane verinin sınıflarının dağılım oranına bakarak yapar. K tane veride hangi sınıf çoğunlukta ise veri o sınıfa atanır. Sistemde bu tahmin edilmiş sınıf olarak belirtilmiştir. KNNde bir eğitim aşaması yoktur. Eğitim aşaması yerine sınıflandırılacak olan veri, eğitim veri seti ile sınıflandırmanın yapıldığı anda karşılaştırılıp karar verilir. Bu sebepten eğitim verisinin özellik matrisi sınıflandırma için bir modeldir. K değeri kullanıcı tarafından içeriğe bağlı olarak seçilir. Sistemde veriler arasında benzerlik üç farklı yolla ölçülür. Bu yollar Euclidean, Manhattan ve Chebyshev uzaklık ölçümleridir.

2.1.3.1 Euclidean uzaklık ölçümü

Euclidean uzaklık iki nokta arasındaki cetvel ile ölçülecek sıradan uzaklıktır. Kartezyen koordinat sisteminde p = (p1,p2,….,pn,) ve q = (q1,q2,….,qn) gibi iki nokta arasındaki euclidean uzaklık d(p,q) olarak tanımlanır. Matematiksel olarak aşağıdaki gibi ifade edilebilir [28].

$$d(p,q) = d(q,p) =$$

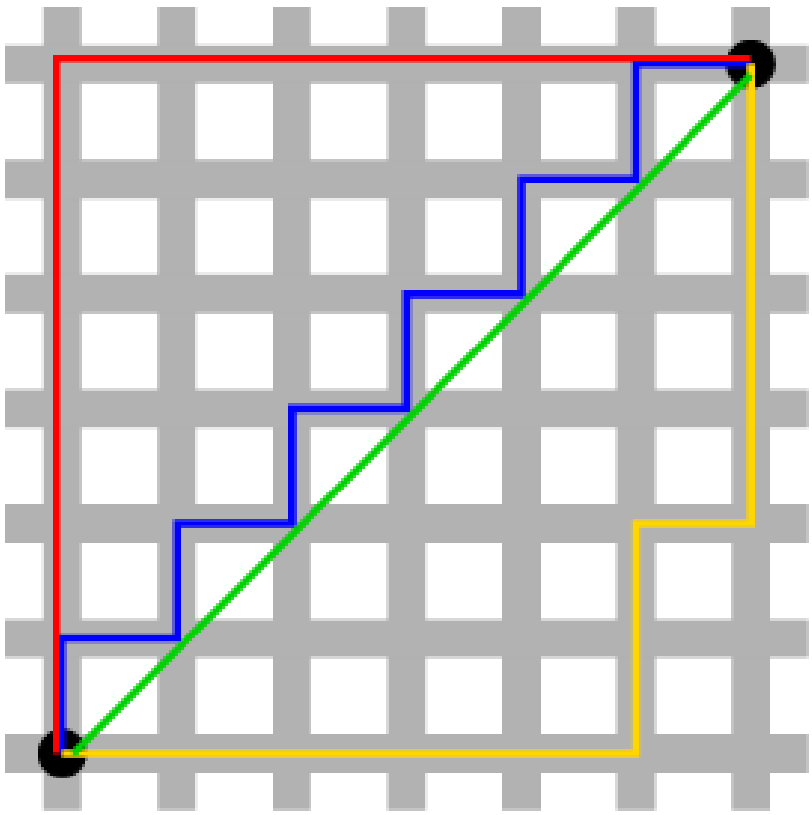
$$\sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \qquad (2.20)$$

2.1.3.2 Manhattan uzaklık ölçümü

Manhattan uzaklık ölçümü kartezyen düzlemdeki n boyutlu iki noktanın her bir koordinatı arasındaki mutlak farkın toplamıdır. Taxicab geometrisi olarak bilinir bu ismi Manhattan yarım adasındaki iki nokta arasındaki en kısa uzunluğu ölçtüğü için almıştır.

Matematiksel ifadesi Kartezyen koordinat sisteminde p = (p_1,p_2,…..,p_n) ve q = (q_1,q_1,….,q_n) gibi iki nokta arasındaki Manhattan uzaklık d(p,q) olarak tanımlanır. Şekil 2-7 de euclidian ve manhattan uzaklık ölçümleri gösterilmiştir. Matematiksel olarak aşağıdaki gibi ifade edilebilir.

$$d(p,q) = \sum_{i=1}^{n} |p_i - q_i| \tag{2.21}$$



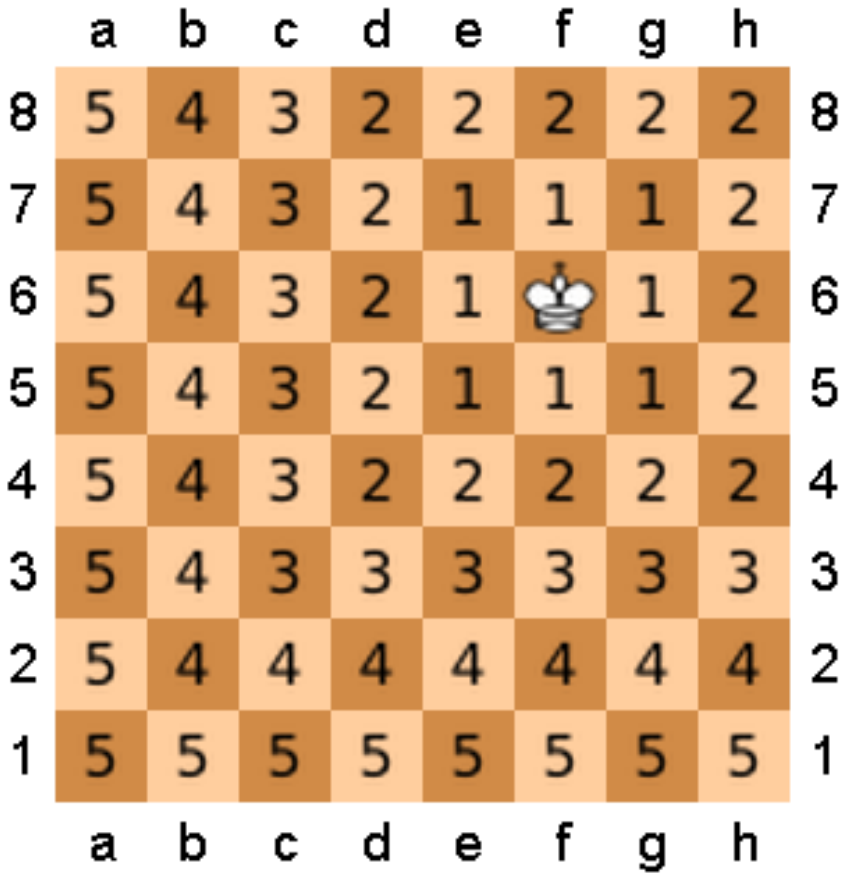
Şekil 2-7 Euclidian ve Manhattan uzaklık ölçüm gösterimi

2.1.3.3 Chebyshev uzaklık ölçümü

Satranç tahtası uzaklığı olarak da bilinir. iki vector arasındaki Chebyshev uzaklık iki vektörün koordinatları arasındaki farklardan en büyüğüdür. Örnek olarak satranç tahtasındaki F6 ve E2 noktaları arasındaki uzaklık F ve E için 1 iken 6 ve 2 arasındaki uzaklık 4 olduğu için Chebyshev 4 olarak hesaplanır. Matematiksel olarak aşağıdaki gibi ifade edilebilir [29].

$$d(p,q) = \max(|p_i - q_i|), \; i = 1,2,3,\dots,n \tag{2.22}$$

Şekil 2-8 de gösterilen Kralın hareketleri Chebyshev uzaklık ölçümünün bir örneğidir. Ayrıca Satranç tahtasındaki Filin hareketleri Euclidian uzaklığa, kalenin hareketleri Manhattan uzaklığa birer örnektir



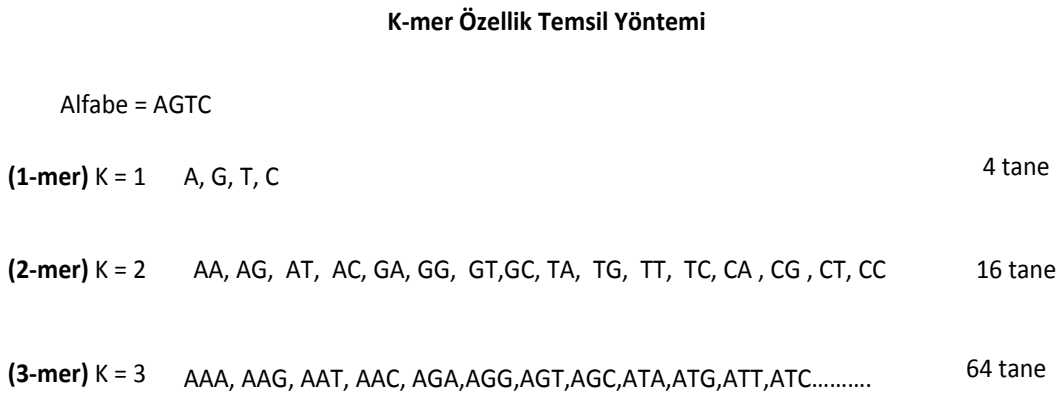
Şekil 2-8 Chebyshev uzaklık ölçümü örneği

2.2 Özellik Temsili

Sistemde tanımlı olan sınıflandırıcıların hepsi için girdi olarak sabit uzunlukta bir özellik vektörü gerekir. Fakat biyolojik dizilimler farklı uzunlukta olduğu için dizilim direk olarak sınıflandırıcıya verilemez.

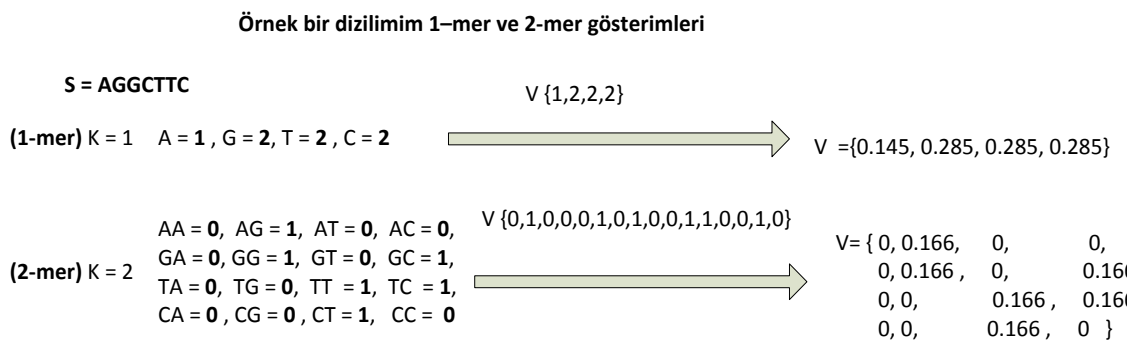
Dizilim kendisi yerine, dizilimi temsil eden sabit sayıda sayısal özellik, diğer bir deyişle özellik vektörü verilir. Birçok yaklaşımın arasından, dizilimlerin kompozisyonel özelliklerinin verimli bir dizilim temsil yöntemi olduğu gösterilmiştir. TRAINER kullanıcılara üç farklı temsil şeması ve bunların kombinasyonları arasında seçme fırsatı sunar. Bu şemalar 1-mer, 2-mer ve 3-mer dir. $A=\{a_1,a_2,...,a_n\}$ gibi bir alfabe üzerinden tanımlanmış bir $S=s_1s_2...s_m$ dizilimi $S_1=\{fa_1,fa_2,...,fa_n\}$ ile temsil edilmiştir. 1-mer modelinde, fa_i a_i'nci elemanın S

stringindeki frekansını gösterir. Bu temsilin genelleştirilmesi k-mer model olarak tanımlanmıştır (Şekil 2-9).



Şekil 2-9 k-mer temsil yöntemi

K-mer modelde S olarak tanımlanmış dizilim $S_k=\{fk_1,fk_2,...\}$ olarak temsil edilmiştir. Bu temsilde fk_i A alfabesinden K uzunlukta oluşturulabilecek bütün stringlerden i.nin frekansıdır. Örnek olarak 2-mer özellik temsil şemasında, n uzunluğundaki A alfabesinden 2 (k = 2) uzunluğunda oluşturulan n^2 tane stringin frekansı mevzu bahistir (Şekil 2-10). Bu şema ile dizinin kendisi ve dizideki alfabe elemanlarının sıraları ifade edilmiş olur. Benzer şemalar başarılı olarak farklı alanlarda uygulanmıştır [16].

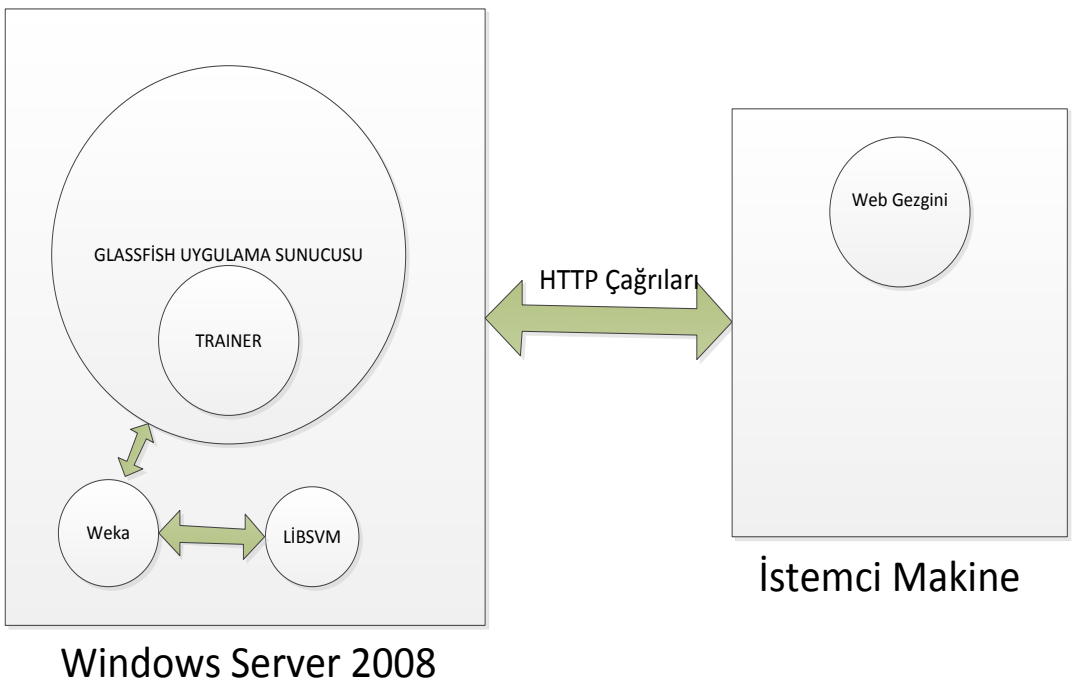


Şekil 2-10 Örnek bir dizilimin 1-mer ve 2-mer özellik temsilinde gösterimi

2.3 Web Sunucu Gerçekleştirimi

Web sunucu Java Server Faces (JSF) framework, WEKA ve LIBSVM kullanılarak gerçekleştirilmiştir. WEKA makina öğrenme algoritmaları topluluğudur. WEKA veri ön işleme, sınıflandırma, kümeleme, ilişki kuralları ve görsel gösterim için

kullanılan araçlar sağlar. Weka'da sınıflandırma yapabilmek için gereken uygulama geliştirme ara yüzünün (API) temel bilgileri EK-A da sunulmuştur. Sistemimiz Glassfish uygulama sunucusu üzerinde koşmaktadır. Bu özellik sistemi herhangi bir işletim sisteminde çalışabilir duruma getirmektedir. Şu anki sürümü Windows Server 2008 üzerinde çalışmaktadır. Şekil 2-11 de sisteminin genel tasarımı gösterilmiştir.



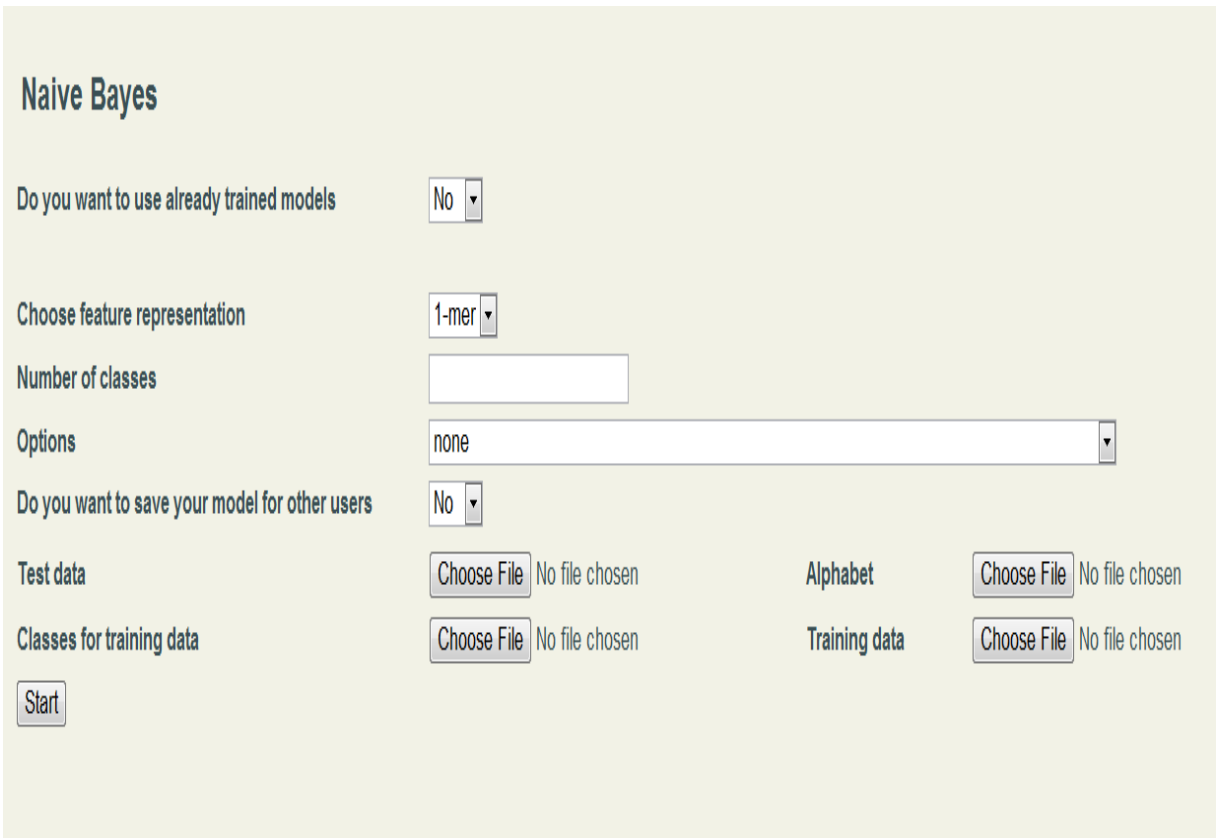
Şekil 2-11 Sistemin genel tasarımı

3. GELİŞTİRİLEN ARAÇ

TRAINER web tabanlı bir dizilim sınıflandırma sistemidir. Kullanıcılar kendi veri setlerini sisteme yükleyebilir ve kendi oluşturdukları modelleri daha sonra kullanmak için kayıt edebilirler. Sisteme http://www.baskent.edu.tr/~hogul/TRAINER adresinden ulaşılabilir.

3.1 Kullanıcı Ara Yüzleri

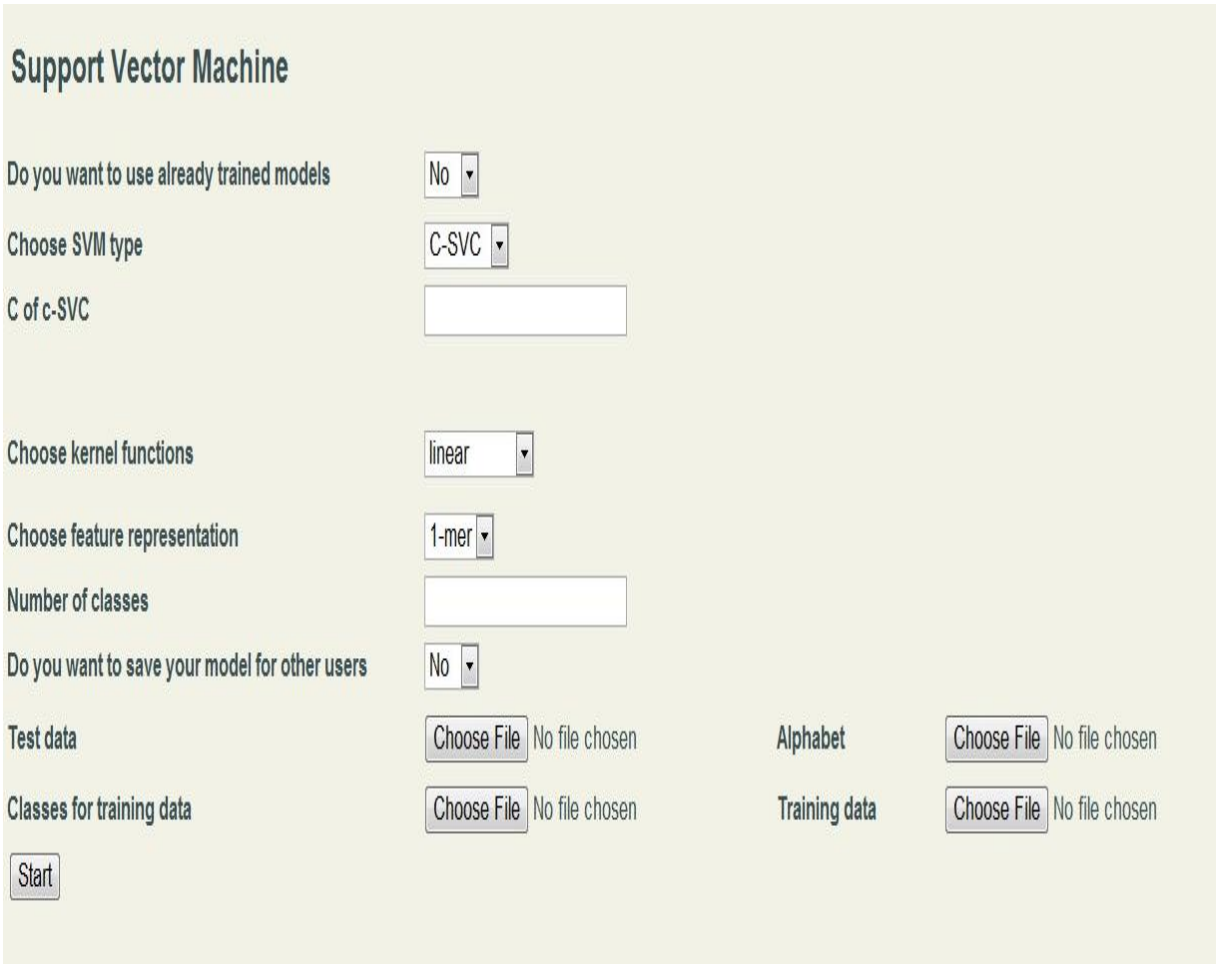
Kullanıcı ara yüzleri üç farklı makina öğrenme yöntemine ulaşım sağlar. Her yöntem farklı parametrik yapılarından dolayı farklı bir ara yüze sahiptir. Naive Bayes sınıflandırıcı için normal dağılım varsayılan olarak belirlenmiştir. Şekil 3-1 de naive Bayes sınıflandırıcının eğitim modu ara yüzü gösterilmiştir.



Şekil 3-1 TRAINER naive Bayes eğitim modu ara yüzü

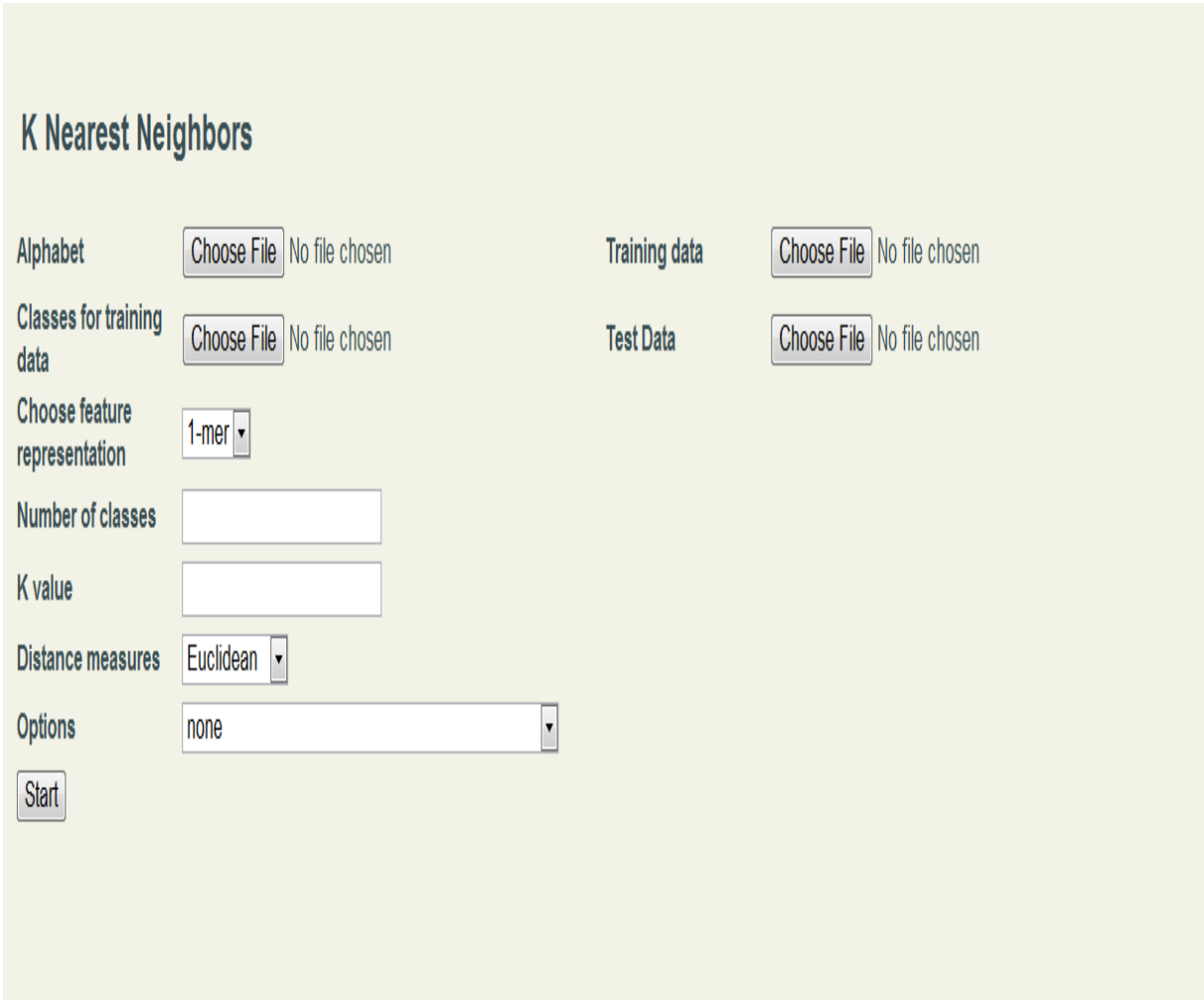
SVM ara yüzünde üç tane yönteme özel parametre bulunur. Bunlar sırası ile SVM tipinin seçildiği C-SVC ve nu-SVC değerleri, SVM tip seçimine göre c veya nu değerleri ve kernel tipleridir.

Kernel tipleri linear, polynomial veya radial basis olabilir. Kernel tipine bağlı olarak da seçilen kernel parametreleri kullanıcı tarafından girilir. Şekil 3-2 de destek vektör makinalarının eğitim modu ara yüzü gösterilmiştir.



Şekil 3-2 TRAINER Destek vektör makinaları eğitim modu ara yüzü

KNN ara yüzünde üç parametre kullanıcı tarafında set edilir. Bunların ilki K değeri bu değer pozitif bir tam sayıdır. İkincisi uzaklık fonksiyondur. Euclidean uzaklık ölçümü varsayılan olarak ayarlanmış olsa bile kullanıcı bu değeri Manhattan veya Chebyshev uzaklık ölçümlerinden biri ile değiştirebilir. Son olarak kullanıcı seçenek kısmından isteğe bağlı olarak özellikleri aynı olan verilerin görmezden gelinmesini sağlayabilir. Şekil 3-3 de en yakın k komşu sınıflandırıcısının eğitim modu arayüzü gösterilmiştir.



Şekil 3-3 TRAINER en yakın k komşu eğitim modu ara yüzü

3.2 Eğitim Modu

Sistemdeki üç algoritma için kullanıcıların kendi eğitim verileri ile sistemi eğitme şansını sahiptir. Sistemde eğitme işlemini yapmak için zaten eğitilmiş modelleri kullan seçeneği kullanıcı tarafından hayır olarak set edilmelidir.

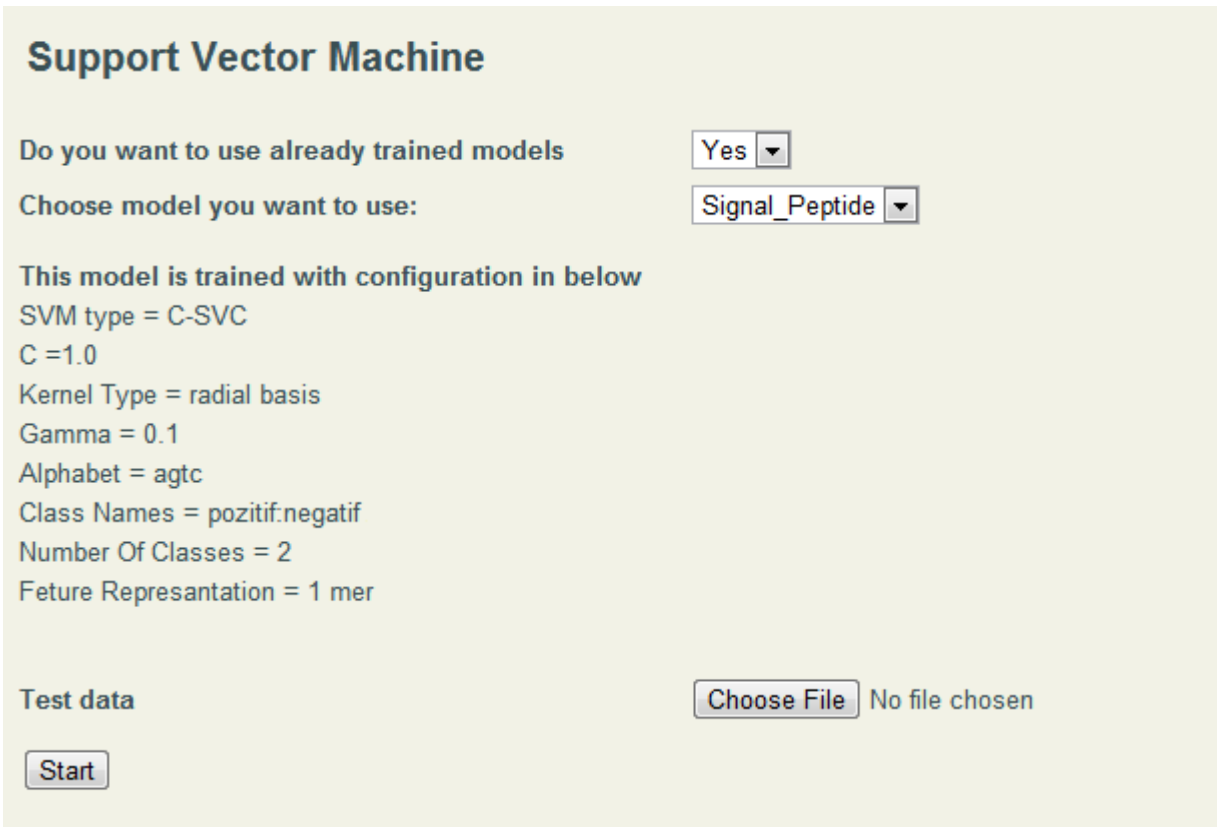
Bu adımdan sonra kullanıcı özellik temsil şemalarından birini seçmelidir. Bu şemalar 1-mer, 2-mer, 3-mer veya hepsinin birleşimidir. Daha sonraki adımda kullanıcı eğitim ve test veri setini oluşturan dizilimlerini Fatsa formatında uzantısı txt olan bir dosya ile sisteme yüklemelidir. Bunların yanında kullanıcı sisteme eğitim dizilimlerinin alfabesi ve sınıflarını belirten txt uzantılı iki dosya daha yüklemesi gerekir. Son olarak kullanıcı “eğitilmiş modeli sisteme kayıt etmek istermisiniz” sorusuna cevap verecektir. Eğer kullanıcı bu adımda evet derse sistem kullanıcından veri seti ve problemi ile ilgili bir isim girmesini ister. Bu isim sistemi, gelecekte yeniden yükleme modunda kullanacak kullanıcılara bu modeli

seçmeleri için listede sunulacaktır. KNN algoritmasında oluşturulan bir model olmadığı için modeli kayıt et gibi bir seçenek yoktur.

3.3 Yeniden Yükleme Modu

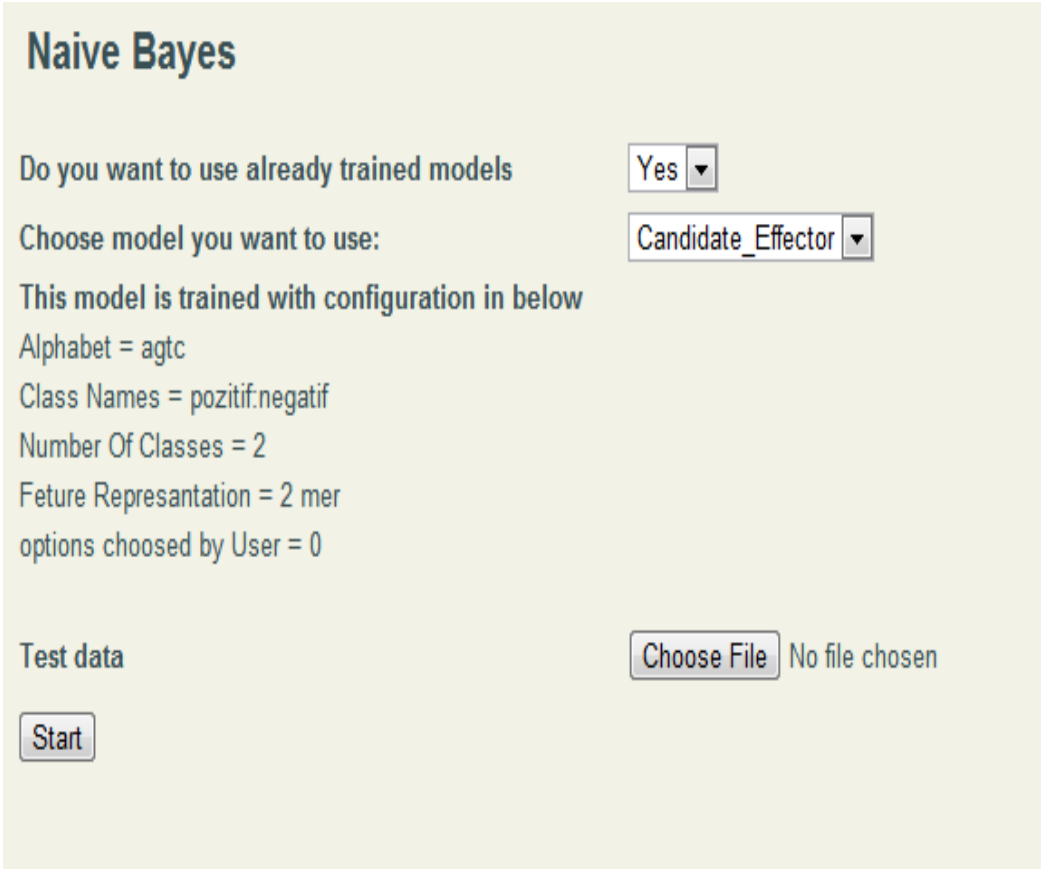
Bu mod naive Bayes ve SVM yöntemleri içindir. En yakın k komşu sınıflandırıcısı tembel (lazy) bir öğrenme algoritması olduğu için bir eğitim aşaması yoktur. Bu sebepten en yakın k komşu algoritması için yeniden yükleme modu yoktur. Kullanıcı daha önceden eğitilmiş modellerden birini kullanmak istediği zaman, sistem daha önceden eğitilmiş modeller listeler. Kullanıcı bunlar arasından birini seçebilir.

Herhangi bir model seçildiği zaman sistem bu model hakkında çeşitli bilgileri listeler. Bu bilgiler Model ismi, modeldeki sınıf sayısı, Sınıfların isimleri, Alfabe, özellik temsili ve kullanılan sınıflandırma yönteminde bağlı olarak bazı ek parametreleridir. Şekil 3-4 de destek vektör makinaları sınıflandırıcısı için yeniden yükleme modu ve bu modda yüklenecek model hakkındaki bilgileri sunan ara yüz gösterilmiştir.



Şekil 3-4 TRAINER Destek vektör makinaları yeniden yükleme modu ara yüzü

Şekil 3-5 de naive Bayes sınıflandırıcısı için yeniden yükleme modu ve bu modda yüklenecek model hakkındaki bilgileri sunan ara yüz gösterilmiştir.



Şekil 3-5 TRAINER Naive Bayes yeniden yükleme modu ara yüzü

4. ÖRNEK ÇALIŞMALAR

Üç örnek çalışma ile TRAINER'ın performansı raporlanmıştır. Bu örnek çalışmalar aday efektör tahmini, micrRNA-hedef bağlanma tahmini ve nükleolar konumlanma sinyal tahminidir. Örnek çalışmalarda sınıflandırıcıların performansları hata matrisi (confusion matrix) kullanılarak gösterilmiştir.

Hata matrisi üzerinde tanımlı olan gerçek pozitif (TP) ifadesi sistemin pozitif örneklerden kaçını doğru olarak sınıflandırdığını belirtir. Sahte pozitif (FP) ise sistem tarafından kaç tane negatif örneğin yanlış sınıflandırma sonucu pozitif olarak sınıflandırdığını belirtir. Aynı şekilde sahte negatif (FN) ifadesi ise sistemin kaç tane pozitif örneği yanlış sınıflandırma sonucu negatif örnek olarak sınıflandırdığını belirtir. Son olarak gerçek negatif (TN) ifadesi sistemin negatif örneklerden kaçını doğru olarak yani negatif olarak sınıflandırdığını belirtir.

Çizelge 4-1 de gösterilen hata matrisi üzerinde doğruluk formül 4-1 deki gibi tanımlanmıştır. Doğruluk (accuracy) sistemin hangi yüzde ile doğru tahminler yaptığını hakkında bilgi verir.

$$doğruluk = \frac{Gerçek\ pozitif + Gerçek\ negatif}{tüm\ elemanlar\ ın\ sayısı} * 100 \qquad (4\text{-}1)$$

$$= \frac{TP + TN}{TP + FP + TN + FN} * 100$$

Özgüllük (specificity) ise sistemin negatif örnekleri hangi doğruluk yüzdesi ile sınıflandırdığın bilgisidir. Formül 4-2 deki gibi tanımlanmıştır.

$$\text{Özgüllük} = \frac{\text{Gerçek negatif}}{\text{Gerçek negatif} + \text{Sahte pozitif}} * 100 \qquad (4\text{-}2)$$

$$= \frac{\text{TN}}{\text{TN} + \text{FP}} * 100$$

Duyarlılık (sensitivity) ise sistemin pozitif örnekleri hangi doğruluk yüzdesi ile sınıflandırdığın bilgisidir. Formül 4-3 deki gibi tanımlanmıştır.

$$Duyarlılık = \frac{Gerçek\ Pozitif}{Gerçek\ Pozitif + Sahte\ Negatif} * 100 \qquad (4\text{-}3)$$

$$= \frac{TP}{TP + FN} * 100 \qquad (4\text{-}4)$$

		Tahmin Edilen Sınıf	
		Pozitif	Negatif
Gerçekte olan Sınıf	Pozitif	Gerçek Pozitif (TP)	Sahte Negatif (FN)
	Negatif	Sahte Pozitif (FP)	Gerçek Negatif (TN)

Çizelge 4-1 Hata matrisi

4.1 Aday Efektör Tahmini

Biotropik bitki patojenleri büyük miktarlardaki tarım ürününe zarar vermektedir. Bu sebeple patojenin sebep olduğu salgın hastalıklar bütün dünyada ekonomik ve stratejik olarak çok büyük önem taşımaktadır. Arpa, buğday, mısır, süpürge darısı, çavdar ve yulaf gibi bitkileri etkileyen Powdery mildew, çok sayıda bitkiyi etkilediği için özellikle tehlikeli bir patolojik organizmadır [11]. Powdery mildew tamamen asalak bir organizmadır. Kendisine gerekli olan besini asalak olarak yaşadığı organizmanın içine tesir eden ve besini emmesini sağlayan organları sayesinde yapar. Bu organlar sinyal peptide içeren efektör proteinler (virulence veya avirulence proteinleri) üretirler. Efektör proteinleri üzerinde yaşadığı bitkinin hücrelerine tesir ederek burada yaşam alanı bulur. [13; 14]. Üzerinde asalak olarak yaşayacağı bitkinin, bu asalak organizmaya direnci sadece uygun direnç faktörlerine sahip olmasına veya olmamasına bağlıdır [11]. Efektörler bazı belli başlı karakteristik özelliklere sahiptir. Bu özellikler sinyal peptide dizilimini içermeleri ve küçük proteinler olmalarıdır. Dizilimlerin birbirinden farkı çok az olsada, bazı intron-extron yapıları bazı benzerlikleri ortaya çıkarabilir [11]. Barley Powdery mildew patojeninin genomu gen sayısı olarak artmasına rağmen bazı belli başlı genleri kaybetmesi daha verimli bir asalaklık olarak sonuçlanmıştır [12].

Bütün ihtimal dâhilindeki efektör proteinler, sinyal peptidelara sahip olduğu ve aynı zamanda kısa proteinler olduğu için efektör protein araması yaparken temel kriterler kısa olması ve sinyal peptideların varlığıdır.

Sistemi eğitmek için sinyal peptide içeren kısa protein dizilimleri ve sinyal peptide içermeyen kısa protein dizilimlerine ihtiyaç duyulmuştur. Sinyal peptide içerdiği

deneysel olarak kanıtlamış 99 tane aday efektör Blumeria granis organizmasından alınmıştır [11]. Peptide içeren efektörler pozitif içermeyenler ise negatif olarak adlandırılmıştır. Elde edilen pozitif verileri doğrulamak için SignalP [15] kullanarak efektörleri bütün genom ve EST dizilimleri içinde tekrar aratılmıştır. Veri seti, SignalP kullanarak bütün genome içinde aratmak pozitif veri setindeki dizilimler ile aynı uzunluktaki ama sinyal peptide içermeyen efektörlerin bulunmasını sağladı.

Bu işlem sonucunda oluşan çok sayıdaki negatif verilerden, pozitif veri setinden kolayca ayrılabilecek olanları elemek için, blast arama aracını kullanıldı. Böylece pozitif sete en fazla benzeyen negatif set elde edilmiş oldu. Bu işlemlerin sonucunda 90 tane pozitif ve 99 tane negatif veri elde edilmiştir.

Yöntem	Duyarlılık (sensitivity)				Özgüllük (specificity)				Doğruluk (accuracy)			
	1-mer	2-mer	3-mer	Hepsi	1-mer	2-mer	3-mer	Hepsi	1-mer	2-mer	3-mer	Hepsi
NaiveBayes	%76	%89	%93	%91	%77	%72	%77	%76	%76	%78	%82	%81
kNN	%61	%74	%35	%51	%77	%63	%62	%58	%68	%67	%53	%56
SVM (RBF)	%60	%73	%68	%100	%88	%80	%81	%100	%79	%78	%77	%100

Çizelge 4-2 TRAINER sonuçları - Aday efektör tahmini

Seçilmiş veri seti ve bu problem için SVM yöntemi doğru tahmin yapma konusunda diğer yöntemlere gözle görülebilir bir üstünlük sağladı (Çizelge 4-2). Veri setindeki sınıf dağılımları iyi bir şekilde dengeli olduğu için, bu sonuç SVM nin aday efektör tahmininde diğer yöntemlere göre üstün olduğunu gösterir. Diğer dikkat çekici bir noktada özellik temsillerinin hepsinin bir arada kullanımı ile yapılan deneylerde %100 bir doğruluk ortaya çıktı. Bu sonuç dizilimlerde nükleotitin frekansının ve sırasının dizilimin efektör protein olup olmaması hakkında büyük bir etkiye sahip olduğu sonucunu ortaya çıkarmıştır.

4.2 MikroRNA-Hedef Bağlanma Tahmini

MikroRNAların gen exprees değerleri üzerinde önemli bir düzenleyici oldukları son zamanlarda ortaya çıkmıştır. Bu sebep ile gen düzenleme mekanizmasını anlamak için mikro-RNA hedeflerinin belirlenmesi kritik bir önem taşımaktadır.

Hedef şeçme olayı, mikroRNA ve onun hedefinin belirli bir lokasyonu arasında oluşan çiftli yapı ile ilgili olduğu için, mikroRNA hedef tanımlama problemi olgun mikroRNA dizilimi ve onun bağlanacağı varsayımsal mRNA arasındaki bağı tahmin etmeye indirilebilir.

Bu problem birçok kez çalışmalara konu olmuştur örneğin RNAHybrid [9], mirTif [10] mirProb [8]. Bu tezde TRAINER aynı problem için denenmiştir. Veri seti deneysel olarak bağlanma durumu ispat edilmiş 138 pozitif örnek ve bağlanma durumu olmadığı bilinen 38 negatif örnekten oluşmaktadır.

Ogul vd. [8] tarafından belirtildiği gibi bütün örnekler bir boyutlu dizilimlere indirgenmiştir. Çiftli yapıdaki herhangi bir bağlanma noktası bütün bağlanma olasılıklarından biri olarak tanımlanıştır. Aynı zamanda çiftli yapıdaki bütün bağlanma olasılıkları bu ikili yapının alfabesini oluşturmaktadır. Bu alfabe 8 farklı sembolden oluşur. Veri setinin beş katlı çapraz doğrulama tekniği kullanılarak sistemde denenmesi ile oluşan sonuçlar aşağıda belirtilmiştir.

Yöntem	Duyarlılık (sensitivity)				Özgüllük (specificity)				Doğruluk (accuracy)			
	1-mer	2-mer	3-mer	hepsi	1-mer	2-mer	3-mer	hepsi	1-mer	2-mer	3-mer	hepsi
NaiveBayes	%88	%78	%84	%83	%7	%53	%21	%21	%75	%73	%76	%75
kNN	%96	%93	%97	%95	%2	%0	%2	%0	%83	%77	%81	%79
SVM (RBF)	%80	%79	%81	%81	%36	%15	%18	%20	%72	%68	%70	%71

Çizelge 4-3 TRAINER sonuçları - MikroRNA-hedef bağlanma tahmini

Çizelge 4-3 kNN yönteminin yüksek doğruluk ile çalıştığını göstermektedir. Fakat bu problemde pozitif ve negatif örnekler eşit bir şekilde dağılmadığı için duyarlılık ve özgüllük arasında bir denge gerekmektedir. Bu bakış açısı ile sonuçlar incelediğinde en iyi sonuçlar naive Bayes yönteminin 2-mer ve 3-mer özellik temsil şeması ile kullanılmasından ortaya çıkmıştır. Genel olarak, 1-mer özelik temsil şeması çiftli yapıdaki sıra bilgisini içermediği için iyi bir şema olarak düşünülmez. TRAINER'ın sonuçlarını daha önce bahsedilen çalışmaların sonuçları ile kıyasladığı zaman RNA Hybrid sonuçları ile yakın fakat mirTif ve mirProbdan daha

kötüdür. Bunun sebebi mikroRNA hedef tahmininde dizilim içeriğinin tek etken olmamasıdır. MirTif ve mirProb dizilim bilgisi dışında başka bilgilerde göz önüne almaktadır.

4.3 Nükleolar Sinyal Tahmini

Proteinler hücre içinde bir kaç farklı konumda bulunabilir. Bu konumlar arasında, nucleolusdaki konumlanma tam olarak anlaşılamamıştır. Scott ve arkadaşları deneysel olarak doğrulanmış nükleolar konumlanma sinyalleri (NoLS) toplamışlardır. Nöral network analizlerini kullanarak bu sinyallerin dizilimlerini karakterize etmeye çalışmışlardır. NoLSler diğer nuclear dizilimler ile içerik olarak çok benzer olduğu için bunlar arasında ayrım yapmak zor bir iştir. Bu örnek çalışmada NoNLS tahmininde TRAINERda bulunan sınıflandırma yöntemlerinin performanslarını inceledik. Veri setinde ki negatif örnekler non-NoLS nuclear dizilimlerinden elde edilmiştir. Veri seti 31 pozitif ve 46 negatif örnek içermektedir.

Yöntem	Duyarlılık (sensitivity)				Özgüllük (specificity)				Doğruluk (accuracy)			
	1-mer	2-mer	3-mer	hepsi	1-mer	2-mer	3-mer	hepsi	1-mer	2-mer	3-mer	hepsi
NaiveBayes	%53	%73	%67	%71	%47	%37	%18	%31	%52	%58	%47	%55
kNN	%81	%64	%93	%62	%29	%33	%0	%36	%60	%51	%55	%51
SVM (RBF)	%68	%71	%82	%64	%45	%50	%34	%47	%59	%62	%63	%57

Çizelge 4-4 TRAINER sonuçları - Nükleolar sinyal tahmini

Çizelge 4-4 de gösterildiği gibi kNN düşük özgüllük ile çalışırken SVM ve Naive Bayes yöntemleri nükleolar sinyal tanımada yakın sonuçlar vermiştir. Özgüllük ve duyarlılık arasındaki denge SVM sonuçlarında çok daha gözle görülebilirdir. Çizelgeden çıkan diğer bir sonuç ise bütün özellik temsil şemalarının kombinasyonunu kullanmanın iyi bir sonuç vermemesidir. Bu sonucun sebebi nükleolar konumlanmayı belirleyen nükleotit dizilimlerin tekrarsız bazı dizilimler olması olabilir. Diğer bir deyişle alfabeden oluşturulabilen bütün ihtimal dâhilindeki nükleotit dizilimlerinin büyük bir bölümü, problem için önem taşımazken sadece belirli bir kısmı önem taşımaktadır.

5. SONUÇ

Bu tezde yeni bir yöntem (metodoloji) önermek yerine biyodizilimlerin analizi için başlıca orijinal yöntemleri içeren yeni bir çevrim içi araç gerçekleştirilmiştir.

TRAINER kullanıcının kendi eğitim veri setini yükledikten sonra sırası ile problemin kendi alfabesini, problemdeki sınıfların sayısını ve ismini girdikten sonra son olarak da özellik temsil şemalarından birini seçerek dizilim sınıflandırması yapabileceği esnek bir çevrim içi araç olarak tasarlanmıştır. TRAINER ara yüzü tecrübesiz internet kullanıcıları düşünülerek kullanıcı dostu bir ara yüz olarak tasarlanmıştır. Sistem elli ve daha az harften oluşan dizilimleri içeren veri setlerinde, set binlerce dizilimden oluşsa bile saniyeler içinde sonuç üretmektedir. Sistemin eğitilmiş modelleri kayıt edebilme özelliğinin gelecekte çok büyük bir eğitilmiş modeller veri tabanı oluşmasına sebep olacağı tahmin edilmektedir. Sisteminin herhangi bir biyolojik dizilim ile doğru sonuç verdiği tezde gösterilmiştir. TRAINER bilindiği kadar ile web tabanlı çalışan ve eğitim seçeneği sunan ilk genel dizilim sınıflandırıcısıdır.

Patojen efektör proteinlerin tespit edilebilmesi bitki patojenleri ve bitki arasındaki etkileşimi inceleyen çalışmalar için önemli bir iştir. Bu tezde sadece nükleotit diziliminden aday efektör tahmini yapan ilk sistematik çalışmanın sonuçları sunulmuştur. Sistemin pozitif ve negatif verilerin ayrışımındaki yüksek doğruluk oranları TRAINER ın ilerde efektör tahmini alanında kullanılmasını özendirecektir.

TRAINER Micro-RNA hedef bağlanma tahmini probleminde, hâlihazırda mevcut olan yöntemlerin sonuçlarına göre bir iyileştirme sağlamasa bile gelecek çalışmalar için detaylı bir alt yapı hazırlamıştır.

Nükleolar proteinleri diğer nuclear proteinlerden dizilimlerine göre ayırmak makina öğrenme alanında zor bir görevdir. Bu ayırımı yapmaya çalışan bilinen işlemsel bir çalışma yoktur. Bu çalışma bir başlangıç noktası olarak bu problemi TRAINER da kayıtlı olan üç yaygın yöntem ile deneyerek sonuçlarını sunmuştur.

TRAINER herhangi bir problem için üç makina öğrenme sınıflandırıcısı ve değişik özellik temsil şemaları açısından bir karşılaştırma noktası olarak düşünülebilir. Uygulamada herhangi bir yöntemin veya özellik temsil şemasının üstünlüğünü

kanıtlayacak sürekli sonuçlar ile karşılaşmamıştır. Bu sebep ile yöntem ve özellik şeması, problemin doğasına özel olarak seçilmelidir. Yöntem ve özellik şeması açısından karşılaştırmalı çalışmalar bazı biyolojik gerçeklerin kapısını aralayabilir ve aynı zamanda yeni test edilebilir hipotezlerin ortaya çıkmasına yardımcı olabilir.

Çalışmanın devamı olarak TRAINER'e yeni temsil şemaları ve yeni sınıflandırma algoritmaları eklenebilir. Yine çalışmanın devamı olarak TRAINER diğer hali hazırdaki çevrim içi araçlarında kullanılabileceği bir alt yapı haline dönüştürülebilir.

Bu tez çalışması, TÜBİTAK tarafından EEEAG 110E160 nolu proje ile desteklenmiştir.

KAYNAKLAR LİSTESİ

Bilimsel periyodikler:

[1] Z. Xing, J. Pei, E.J.Keogh, A brief survey on sequence classification, ,SIGKDD Explorations, s.40-48, 2010.

[2] O.Emanuelsson, S.Brunak, G.Heijne, H.Nielsen, Locating proteins in the cell using TargetP, SinyalP, and related tools. Nature Protocols 2, s.953-971, 2007.

[3] S.Saha and G. P. S. Raghava, AlgPred: prediction of allergenic proteins and mapping of IgE epitopes, NucleicAcids Res, s34, 2006.

[4] M.G.Reese, Application of a time-delay neural network to promoter annotation in the Drosophila melano gastergenome, ComputChem ,s.51-6,2001.

[5] S.Tyagi, C.Vaz, V.Gupta, R.Bhatia, S.Maheshwari, A.Srinivasan A.Bhattacharya, CID-miRNA: A web server for prediction of novel miRNA precursors in human genome, Biochemical and Biophysical Research Communications, vol.372, no 4, 2008, s.831-834, 2008.

[6] C. Burge, S. Karlin, Prediction of complete gene structures in human genomic DNA, J. Mol. Biol, vol 268, s.78-94. 1997

[7] C.Chang, C.Lin, LIBSVM: a library for support vector machines, ACM Transactions on Intelligent Systems and Technology, vol.2.27, s.1-27, 2011.

[8] H.Oğul, S.U.Umu, Y.Y.Tuncel, M.S.Akkaya, A probabilistic approach to microRNA-targetbinding, Biochemical and Biophysical Research Communications vol.413, s.111-115, 2011.

[9] M. Rehmsmeier, P. Steffen, M. Hochmann vd. ,Fast and effective prediction of microRNA/target duplexes, RNA 10, s.1507-1517, 2004.

[10] Y. Yang, Y.P. Wang, K.B. Li, MiRTif: a support vector machine-based microRNA target interaction filter, BMC Bioinformatics vol.9 s4, 2008.

[11] D.Godfrey, H.Böhlenius, C.Pedersen, vd., Powdery mildew fungal effector candidates share N-terminal Y/F/WxC-motif, BMC Genomics,vol11,s.317 2010.

[12] P.D.Spanu, J.C.Abbott, J.Amselem, vd. , Genome Expansion and Gene Loss in Powdery Mildew Fungi Reveal Tradeoffs in Extreme Parasitism, Science, vol.330, s.1543-1546, 2010.

[13] CJ.Ridout, P.Skamnioti, O.Porritt, vd., Multiple avirulence paralogues in cereal powdery mildew fungi may contribute to parasite fitness and defeat of plant resistancei, The Plantcell, vol.18, s.2402-2414,2006.

[14] Z.Zhang, C.Henderson, E.Perfect, vd., Of genes and genomes, needles and haystacks: Blumeria graminis and functionality, Molecular plant pathology, vol.6, s.561-575, 2005.

[15] JD.Bendtsen, H.Nielsen, G.Heijne G. Von, S.Brunak, Improved prediction of sinyal peptides: SinyalP3.0, Journal of molecular biology, vol 340, s.783-795, 2004.

[16] H.Oğul, E.Mumcuoğlu, A discriminative method for remote homology detection based on n-peptide compositions with reduced amino acid alphabets, Biosystems, vol.87, s.75-81, 2007.

[17] M.Hussain, S.K.Wajid, A.Elzaart, M.Berbar, A Comparison of SVM Kernel Functions for Breast Cancer Detection, Eighth International Conference Computer Graphics, Imaging and Visualization, 2011.

[18] C.Hsu, C.Chang, C.Lin , A Practical Guide to Support Vector Classification, 2003

[19] W.S Noble, What is a support vector machine? ,Nature Biotechnology, vol.24, s.1565-1567, 2006.

[20] R.Kong, B.Zhang, Autocorrelation Kernel Functions for Support Vector Machines, Third International Conference on Natural Computation, 2007.

[21] T.Howley, M.G.Madden,The Genetic Kernel Support Vector Machine:

Description and Evaluation, Artificial Intelligence Review, vol.24, no.3-4, s.379-395, 2005.

[22] D.Boswell, Introduction to Support Vector Machines, 2002.

[23] V.Vapnik, C.Cortes, Support vector networks, Machine Learning, vol. 20, s. 273-297, 1995

[24] D. Huson, SVMs and Kernel Functions, Algorithms in Bioinformatics II SoSe'07 ZBIT, s.265, 2007

Görsel Kaynaklar:

[25] Y.Abu-Mostafa, Lecture 14 - Support Vector Machines, 2012, http://www.youtube.com/watch?v=eHsErlPJWUU

[26] Y.Abu-Mostafa, Lecture 15 - Kernel Methods, 2012, http://www.youtube.com/watch?v=XUj5JbQihlU&feature=relmfu%20kernel%20methods

Kitaplar:

[27] R.O.Duda, P.E.Hart, D.G.Stork, John, Pattern Classification, 2.Baskı, John Wiley & Sons, Bölüm 2 s.4.,2001.

[28] E.Deza, M.M.Deza, Encyclopedia of Distances, s.94, Springer, 2009.

[29] D.M.J. Tax, R.Duin, D.Ridder, Classification, Parameter Estimation and State Estimation: An Engineering Approach Using MATLAB, John Wiley and Sons, s.360, 2009.

EKLER LİSTESİ

EK A WEKA Java Api Kullanımı Hakkında Temel Bilgiler

1) Data için özellik belirtimi yaratma

```
FastVectorfvWekaAttributes = new FastVector(ABCsize);
for (int i = 0; i <ABCsize; i++) {
Attributetemp Attribute = new Attribute("Numeric" + i);
fvWekaAttributes.addElement(tempAttribute);
  }
FastVectorfvClassVal = new
FastVector(Integer.parseInt(this.getClassNumber()));
for (int i = 0; i <Integer.parseInt(this.getClassNumber()); i++) {
fvClassVal.addElement("Class" + (1 + i));
      }
      Attribute ClassAttribute = new Attribute("theClass", fvClassVal);
fvWekaAttributes.addElement(ClassAttribute);
```

2) Data oluşturma

```
      Instance iExample = new DenseInstance(ABCsize + 1);
      for (int i = 0; i <ABCsize; i++) {
iExample.setValue((Attribute) fvWekaAttributes.elementAt(i), 1]);
       }
```

3) Datalardan oluşan veri seti yaratma

Instances isTrainingSet = new Instances("Rel", fWekaAttributes, NumberOfTrainningSamples);

4) **Datalardan oluşan Veri setinde Sınıfı belirten özelliğin indexinin set edilmesi**

isTrainingSet.setClassIndex(ABCsize);

5) **Sınıflandırıcı fonksiyonların yaratılması**

LibSVM SVM = new LibSVM();
NaiveBayes Bayes= new NaiveBayes();
LinearNNSearchKnn = new LinearNNSearch();

6) **Sınıflandırıcı fonksiyonların parametrelerinin set edilmesi**

DistanceFunctiondf = new EuclideanDistance();

Knn.setDistanceFunction(df);

Knn.setOptions(Knnoptions);

SVM.setCost(Double.parseDouble(this.getcValue()));

SVM.setNu(Double.parseDouble(this.getNuValue()));

SVM.setCoef0(Double.parseDouble(this.getCoef0()));

SVM.setDegree(Integer.parseInt(this.getDegree()));

SVM.setGamma(Double.parseDouble(this.getGamma()));

Bayes.setOptions(Naiveoptions);

7) **Sınıfladırıcı Fonksiyonun data set ile Modeli oluşturması**

SVM.buildClassifier(isTrainingSet);

```
Bayes.buildClassifier(isTrainingSet);
```

8) Sınıflandırılması için Verilen datanın Sınıflandırma işlemi

```
double[] fDistribution = Bayes.distributionForInstance(iUse);
double[] fDistribution = SVM.distributionForInstance(iUse);
Instances neighbors = Knn.kNearestNeighbours(iUse, knumber);
```

EK B Tezde İsmi Geçen Web Araçların Adresleri

TargetP : http://www.cbs.dtu.dk/services/TargetP/

Algpred : http://www.imtech.res.in/raghava/algpred/

Genscan : http://genes.mit.edu/GENSCAN.html

BDGP: http://www.fruitfly.org/

Mirna : http://mirna.jnu.ac.in/cidmirna/index.html