

**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ**

**ÖRÜNTÜ TANIMA VE ÖZNİTELİK SEÇME  
YÖNTEMLERİ KULLANARAK KISA ZAMAN  
SONRAKİ YOL TRAFİK HIZ ÖNGÖRÜSÜ**

**YÜKSEK LİSANS TEZİ  
Müh. Ümit Ülkem YILDIRIM**

**Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ**

**Programı : BİLGİSAYAR MÜHENDİSLİĞİ**

**HAZİRAN 2008**

**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ**

**ÖRÜNTÜ TANIMA VE ÖZNİTELİK SEÇME  
YÖNTEMLERİ KULLANARAK KISA ZAMAN  
SONRAKİ YOL TRAFİK HIZ ÖNGÖRÜSÜ**

**YÜKSEK LİSANS TEZİ  
Müh. Ümit Ülkem YILDIRIM  
(504051533)**

**Tezin Enstitüye Verildiği Tarih : 10 Temmuz 2008  
Tezin Savunulduğu Tarih : 10 Haziran 2008**

**Tez Danışmanı : Doç.Dr. Zehra ÇATALTEPE  
Diğer Jüri Üyeleri Yrd. Doç. Dr. Mustafa KAMAŞAK  
Yrd. Doç. Dr. M. Elif KARSLIGİL (Y.Ü.)**

**HAZİRAN 2008**

## ÖNSÖZ

Bu tez çalışmam süresince yardımlarını esirgemeyen Doç.Dr. Zehra Çataltepe'ye öncelikle teşekkür ediyorum, yüksek lisans programı öncesinde benim için yabancı bir konu olan örüntü tanıma yöntemlerini çok iyi bildiği ve sevdirecek anlattığı için. Çalışmam esnasında beliren bir çok sorunu onun özgün fikirleriyle aştım.

İstanbul Büyükşehir Belediyesi Trafik Kontrol Merkezi yönetimi ve çalışanlarına, kullandığım veriyi temin etmemde yardımlarını esirgemedikleri için ayrıca teşekkür etmek istiyorum.

Tabiki annem, babam ve kardeşime destekleri ve sabırları için çok teşekkür ediyorum, onlar olmasaydı bu çalışmayı yapamazdım.

Son olarak çalışma sırasında vakit ayıramadığım herkese anlayışları için teşekkür ediyorum.

Mayıs 2008

Ümit Ülkem Yıldırım

## **İÇİNDEKİLER**

<b>KISALTMALAR</b>	<b>V</b>
<b>TABLO LİSTESİ</b>	<b>VI</b>
<b>ŞEKİL LİSTESİ</b>	<b>VII</b>
<b>SEMBOL LİSTESİ</b>	<b>IX</b>
<b>ÖZET</b>	<b>X</b>
<b>SUMMARY</b>	<b>XII</b>
<b>1. GİRİŞ</b>	<b>1</b>
<b>2. AKILLI ULAŞIM SİSTEMLERİ VE GEÇMİŞ ÇALIŞMALAR</b>	<b>4</b>
2.1 Akıllı Ulaşım Sistemleri	4
2.1.1 Trafik Sinyal Kontrol Sistemi	4
2.1.2 Transit Yollar Yönetim Sistemi	5
2.1.3 Kaza Yönetim Sistemi	5
2.1.4 Toplu Taşıma Araçları Yönetim Sistemi	5
2.2 Kısa Zamanlı Hız Tahmini Konusunda Yapılmış Çalışmalar	5
2.2.1 Doğrusal Regresyon	5
2.1.2 Kalman Süzgeçi	6
2.1.3 Benzetim	6
2.1.4 Neural Network	7
2.1.5 Karma Yöntemler	8
<b>3. VERİ KÜMESİ VE VERİ HAZIRLAMA</b>	<b>9</b>
3.1 Veri Kümesi	9
3.1.1 Trafik Öngörüsünde Kullanılan Veri Kaynakları	9
3.1.2 Çalışmada Kullanılan Veri Kaynağı	10
3.1.3 Diğer Veri Kaynakları	13
3.1.3.1 BeatTheTraffic	13
3.1.3.2 Tayvan Uluslararası Yollar Yönetim Merkezi (TANFB)	14
3.2 Veri Hazırlama	14
3.3 Veri Temizleme	17
3.4 Veri İşleme	18
3.5 Birbirine Bağımlı Sensörlerin Bulunması	18

<b>4. ÖRÜNTÜ TANIMA YÖNTEMLERİ</b>	<b>22</b>
4.1 Regrasyon Yöntemleri	22
4.1.1 Karar Destek Yöntemi	22
4.1.1.1 Karar Destek Makinası (SVM)	22
4.1.1.2 Karar Destek Regresyon (SVR)	28
4.1.3 k-Nearest Neighbor(KNN)	30
4.2 Öznitelik Seçme Yöntemleri	31
4.3 Örüntü Tanıma Algoritmalarının Performanslarının Ölçümü	31
4.4 mySVM Programı	32
4.4.1 Programın Kullanılması	32
4.4.2 Programın Çalıştırılması	35
<b>5 GELECEĞE YÖNELİK HIZ TAHMİNİ</b>	<b>38</b>
5.2 Başka Sensörlerdeki Verilerin Kullanılması	46
5.3 Birbirlerine bağımlı sensörlerin bulunması	48
5.4 Yüksek Korelasyonlu Sensörlerin Birbirleri Yerine Kullanılması	55
<b>6. SONUÇLAR VE TARTIŞMA</b>	<b>58</b>
<b>KAYNAKLAR</b>	<b>59</b>
<b>ÖZGEÇMİŞ</b>	<b>62</b>

## **KISALTMALAR**

<b>SVM</b>	: Support Vector Machine
<b>SVR</b>	: Support Vector Regression
<b>KNN</b>	: k-Nearest Neighbor
<b>RTMS</b>	: Remote Transmission Micro Sensor
<b>ANN</b>	: Artificial Neural Network
<b>IBB-TKM</b>	: İstanbul Büyükşehir Belediyesi Trafik Kontrol Merkezi
<b>FSM</b>	: Fatih Sultan Mehmet
<b>MATLAB</b>	: Matrix Laboratory
<b>SRM</b>	: Structered Risk Minimization
<b>ERM</b>	: Emperical Risk Minimization
<b>ITS</b>	: Intelligent Transportation System
<b>TANFB</b>	: Taiwan Area National Freeway Bureau

## TABLO LİSTESİ

	<u>Sayfa No</u>
<b>Tablo 3.1</b>	Örnek Kayıt Bloğu..... 12
<b>Tablo 3.2</b>	Eğitmen Kümesi Görüntüsü ..... 16
<b>Tablo 3.3</b>	Yüksek bağıntıya sahip sensörler ..... 20
<b>Tablo 4.1</b>	Çekirdek Fonksiyonları ..... 29
<b>Tablo 5.1</b>	Azalan önem sırası ile, üç değişik noktada SVM kullanılarak seçilen öznelikler..... 41
<b>Tablo 5.2</b>	Azalan önem sırası ile, üç değişik noktada KNN kullanılarak seçilen öznelikler..... 41
<b>Tablo 5.3</b>	Yüksek bağıntılı sensörler ..... 49

## ŞEKİL LİSTESİ

		<u>Sayfa No</u>
Şekil 3.1	IBB Trafik Kontrol Merkezi Sayfasında Trafiğin Anlık Durumunu Gösteren Canlandırma .....	11
Şekil 3.2	Beat the traffic sayfasından bir görüntü .....	13
Şekil 3.3	TANFB sitesinden alınan anlık yol durum bilgisi.....	14
Şekil 4.1	İki sınıfı ayıran örnek sınıflandırıcılar.....	23
Şekil 4.2	Karar destek makenizmasında destek vektörleri.....	24
Şekil 4.3	Sınıflandırıcı gösterimi.....	24
Şekil 4.4	Karar destek makenizmasında hata kabülü.....	26
Şekil 4.5	Doğrusal düzlemde sınıflandırma.....	27
Şekil 4.6	Çok boyutlu uzayda sınıflandırma.....	28
Şekil 4.7	SVR ile regresyon.....	29
Şekil 4.8	KNN yöntemi gösterimi.....	31
Şekil 4.9	mySVM programının çıktısı.....	36
Şekil 5.1	s13, s59 ve s68 sensörlerinin konumları.....	38
Şekil 5.2	s268 sensörü için 5 dakika sonraki hız öngörüsü.....	39
Şekil 5.3	s13 sensörü için 5 dakika sonraki hız öngörüsü.....	40
Şekil 5.4	s59 sensörü için 5 dakika sonraki hız öngörüsü.....	40
Şekil 5.5	s268 sensörü için artan sürelerde hız öngörüsü.....	42
Şekil 5.6	s13 sensörü için artan sürelerde hız öngörüsü.....	43
Şekil 5.7	s59 sensörü için artan sürelerde hız öngörüsü.....	43
Şekil 5.8	s268 için 24/12/2007 tarihinde 00:00 ile 23:55 saatleri arasında SVM metodu için 5 dakikalık hız öngörüsü-gerçek hız.....	44
Şekil 5.9	s268 için 24/12/2007 tarihinde 00:00 ile 23:55 saatleri arasında .KNN metodu için 5 dakikalık hız öngörüsü-gerçek hız.....	44
Şekil 5.10	s268 için 24/12/2007 tarihinde 00:00 ile 23:55 saatleri arasında SVM metodu için 60 dakikalık hız öngörüsü-gerçek hız.....	45
Şekil 5.11	s268 için 24/12/2007 tarihinde 00:00 ile 23:55 saatleri arasında KNN metodu için 60 dakikalık hız öngörüsü-gerçek hız.....	45
Şekil 5.12	s13, s59, s268 noktalarından 3/12/2007 Pazartesi günü 00:00 ile 23:55 saatleri arasında alınan hız ölçümleri.....	46
Şekil 5.13	s268 noktasında 60 dakika sonraki hız tahmini için geriye doğru öznitelik seçimi.....	47
Şekil 5.14	s268 noktasında 60 dakika sonraki hız tahmini için geriye doğru öznitelik seçimi .....	48
Şekil 5.15	s95, s73 ve s4'ün harita üzerinde konumları.....	49
Şekil 5.16	s266, ts266'ın harita üzerinde konumları.....	50
Şekil 5.17	s156, ts156'nın harita üzerinden konumları.....	50
Şekil 5.18	s95 in sadece s95 kullanılarak bulunması.. .....	51

<b>Şekil 5.19</b>	s95 in s95 ve s73 sensörleriyle bulunması.....	51
<b>Şekil 5.20</b>	s73 ün sadece s73 sensörleriyle bulunması.....	52
<b>Şekil 5.21</b>	s73 ün s95 ve s73 sensörleriyle bulunması.....	52
<b>Şekil 5.22</b>	Tahmin edilen zamanın arttırılmasıyla s95 SVM metodunun kullanılması.....	53
<b>Şekil 5.23</b>	Tahmin edilen zamanın arttırılmasıyla s95 KNN metodunun kullanılması.....	53
<b>Şekil 5.24</b>	Tahmin edilen zamanın arttırılmasıyla s73 SVM metodunun kullanılması.....	54
<b>Şekil 5.25</b>	Tahmin edilen zamanın arttırılmasıyla ts266 SVM metodunun kullanılması.....	54
<b>Şekil 5.26</b>	Tahmin edilen zamanın arttırılmasıyla s171 SVM metodunun kullanılması.....	55
<b>Şekil 5.27</b>	Tahmin edilen zamanın arttırılmasıyla s95 SVM metodunun kullanılması.....	56
<b>Şekil 5.28</b>	Tahmin edilen zamanın arttırılmasıyla s156 SVM metodunun kullanılması.....	56
<b>Şekil 5.29</b>	Tahmin edilen zamanın arttırılmasıyla ts266 SVM metodunun kullanılması.....	57

## SEMBOL LİSTESİ

<b>V(t)</b>	: t anında ölçülen hız
<b>V(t + Δt)</b>	: t anından Δt dakika sonra öngörülen hız
<b>V<sub>x</sub>(t)</b>	: x sensörü için t anında ölçülen hız
<b>s</b>	: Özniteliklerin hangi sensörden ve hangi zaman aralığından alınacağını belirleyen bir zaman serisidir.
<b>SS<sub>xy</sub></b>	: x ile y'nin birbiri ile korelasyonu
<b>M</b>	: Margin genişliği
<b>ε<sub>k</sub></b>	: SVM'de hata oranı
<b>C</b>	: SVM'de hata Kabul katsayısı
<b>k</b>	: KNN'de k katsayısı

## ÖRÜNTÜ TANIMA VE ÖZİNİTELİK SEÇME YÖNTEMLERİ KULLANARAK KISA ZAMAN SONRAKİ YOL TRAFİK HIZ ÖNGÖRÜSÜ

### ÖZET

Trafik 20. yüzyılda hayatımıza girdiğinden bugüne kadar önemini giderek arttırmıştır. Bugün ise büyük kentlerde trafiğin hayatı kolaylaştırdığından değil, hayatı ne kadar zor hale getirdiğinden bahsedilmektedir. Bunun temel nedeni trafikte harcanılan zamanın giderek artması ve trafik sıkışıklığı gibi bir çok soruna neden olarak insanların hayatlarını zorlaştırmasıdır. Modern dünyanın neden olduğu bu soruna yine modern yöntemlerle çözüm bulabilmek için akıllı ulaşım sistemlerinin geliştirilmesine ve kullanılmasına büyük kentlerde önem verilmeye başlanmıştır. Bu akıllı servisler içinde trafiğin durumuna göre trafik lambalarının yanma süresini değiştiren veya yolcu otobüslerinin konumlarına göre ne zaman durakta olacaklarını bildiren sistemler sayılabilir. Bu tezde üzerinde çalışılan trafik hızı öngörüsü de akıllı ulaşım sistemleri içinde çözülmeye çalışılan konulardan bir tanesidir. Bu öngörünün insanlara duyurulmasıyla yolcular programlarını daha düzenli yapabilecek, trafik sıkışıklığı yoğun olabilecek bölgeden uzaklaştırılarak alternatif yollara yönlendirilebilecektir. Trafik sıkışıklığının azalması sonucu hem zaman kazancı olacak, hem de araçlar daha az yakıt harcayacağından maddi getirisi de olacaktır.

Trafik hızı öngörüsü, kullanıcıların geçeceği yollarda oluşabilecek hızı öngörmeyi amaçlamaktadır. Bunun için yolculuğun süreceği rota boyunca yerleştirilen sensörlerden alınan hız bilgileriyle her sensör için genel hız davranış durumunun çıkarılması amaçlanmaktadır. Genel hız davranış durumunun öğrenilmesinden sonra yapılması gereken şu anki hızı bu duruma göre yorumlayarak x dakika sonraki oluşabilecek hızı tahmin etmektir.

Trafiğin genel durumu haftanın hangi gününde olduğu, hangi mevsimde olduğu, trafiğe çıkan araç sayısına gibi bir çok parametrenin etkilemesi sonucu oluşmaktadır. Bu nedenle bir noktada oluşabilecek hızı bulmak karmaşık bir işlem haline gelmektedir. Bu çalışmada karar destek regresyon(SVR) yöntemi kullanılarak trafik hızı öngörüsü yapılacaktır. SVR yöntemi yerel sınırlara yakalanmaması ve zaman serileri analizinde başarılı sonuçlar verdiği için bu çalışmada kullanılmıştır. SVR ile yapılan testleri karşılaştırmak için k-En Yakın Komşu metodu kullanılmıştır.

Çalışmada kullanılan veriler İstanbul Büyükşehir Belediyesi Trafik Kontrol Merkezinden alınmıştır. Bütün İstanbul'a yayılmış 327 sensörün verileri 1-2 dakika aralıklarla 45 gün boyunca alınmış ve çalışmada kullanılmıştır. Çalışmada trafik hızı öngörüsü için temel olarak 4 yöntem kullanılmıştır.

- a) Sadece ilgili sensörün verileri kullanılarak 5 dakikadan 60 dakikaya kadar trafik hızı öngörüsü yapılmıştır. SVR ve KNN yöntemlerinin performansı karşılaştırılmıştır.
- b) Sensöre yakın konumlardaki ve bağlı yollardaki diğer sensör verileri kullanılarak 5 dakikadan 60 dakikaya kadar trafik hızı öngörülme çalışılmıştır. SVR ve KNN yöntemlerinin performansı karşılaştırılmıştır.
- c) Sensöre yüksek korelasyonla bağlı sensörler bulunarak ilgili sensör için trafik hızı öngöründe yardımcı olarak kullanılmıştır. SVR ve KNN yöntemlerinin performansı karşılaştırılmıştır.
- d) İlgili sensör verileri kullanılmadan sadece yüksek korelasyonla bağlı bulunan sensörlerin verileri kullanılarak trafik hızı öngörüsü yapılmıştır. İlgili sensörden veri alınmadığında oluşabilecek hızları tahmin için kullanılabilir.

Yapılan testler sonucunda genel olarak SVR yönteminin KNN yönteminden daha başarılı olduğu görülmüştür. Fakat yöntemlerin başarımları oranları sensörün konumu, bağlı olan yolların durumu, trafik yoğunluğunun sıkça olduğu bir yerde olup olmadığı gibi birçok parametreye bağlı olduğundan değişiklik göstermektedir. İlgili birden çok sensör verisi kullanılarak yapılan tahminlerin ise bir sensör verisi kullanılarak yapılan tahminlerden daha iyi sonuç verdiği görülmüştür. İlgili sensörün bozulduğu varsayılarak onun yerine korelasyonu yüksek diğer sensör verilerinin kullanıldığında sonuçların ilgili sensörün kullanıldığı duruma gayet yakın sonuçlar verdiği görülmüştür.

## **SHORT TIME TRAFFIC SPEED PREDICTION USING PATTERN RECOGNITION AND FEATURE SELECTION METHODS**

### **SUMMARY**

The importance of traffic has been increased since 20th century. However, nowadays the bad reputation of traffic is greater than its good reputation, because of the time wasting especially in big cities. This problem is created by modern world so modern methods are necessary for solve this problem. So, intelligent transformation system is developed and used in metropol to solve traffic problem. These intelligent systems can change the time of traffic lamb according to traffic situation or understand the public bus positions and give information to their waiting passengers. Short time traffic speed prediction, which is subject of this thesis, is one of the subjects of intelligent transportation systems. When passengers learn short time traffic situation, they can make their plan more correctly and can be guided to alternative roads. If traffic conjection is decreased, people can arrive their target more rapidly and they can decrease their fuel usage.

Short time traffic speed prediction estimates traffic speed on passengers' road when they travel. To make this, they calculate general speed situation for every sensor location on passenger route. After finding general speed situation, current traffic speed is used to predict x minutes later traffic speed.

Situation of traffic is effected by lots of parameters like the day of the week, the season of current day, number of vehicles in traffic. Because of this, predicting the speed in one location becomes a complex problem. In this working the short time speed prediction is calculated by using Support Vector Regrassion (SVR) method. Since support vector machines have greater generalization ability and guarantee global minima for given training data, it is believed that support vector regression will perform well for this problem. To compare result of SVR, k-Nearest Neighbor (KNN) method is also used.

The data used in this thesis is obtained from the Istanbul Municipality Traffic Control Center's web site. Speed data of 327 different sensors is taken by 1 or 2 minutes throughout 45 days. In this thesis, 4 main methods are used to make traffic prediction.

a) Only one sensor's data is used to make speed prediction between 5 minutes to 60 minutes. SVR method performance ise compared with KNN method performance.

b) Other sensor data's is also used which is near to main sensor and near to connected road's sensors and maked prediction between 5 minutes to 60 minutes. SVR method performance is compared with KNN method performance.

c) With finding most correlated sensors to main sensor, their data is used to help prediction. SVR method performance is compared with KNN method performance.

d) To simulate sensor failure, only most correlated near sensor is used to predict failure sensor speed.

According to test results, SVR method is successful than KNN method for this problem. However, the success of methods is changed by lots of parameters like sensor location, connection road which is near sensor and traffic situations. Tests show us that using more than one sensor's data is better than using only the main sensor's data. Also tests show us that, using near or most correlated sensor's data is good to predict failure sensor's speed.

## 1. GİRİŞ

Şehirlerin büyümesi ve nüfusun kalabalıklaşmasıyla beraber insanların şehrin içinde bir noktadan bir noktaya araçlarıyla ulaşmaları giderek daha da zorlaşmaktadır. Günün belirli vakitlerinde oluşan olağan trafik sıkışıklığının yanı sıra beklenmeyen trafikte bekleme süreleri insanların hayatlarını olumsuz şekilde etkilemektedir. Her ne kadar belediyeler bu durumu çözmek çeşitli önlemler alsa da, bunlar hem çok fazla maliyete neden olmakta hem de özellikle şehirlerin eski yerleşim birimlerinde yeni yolları açmak mümkün olamaması gibi çeşitli etkenler sonucu bu çabalar çoğu zaman sonuçsuz kalmaktadır. Akıllı trafik sistemleri bu noktada yardımımıza koşmakta ve trafik işaretleri kontrol sistemi, elektronik bilet toplama sistemi, transit yollar yönetim sistemleri gibi bir çok uygulamayı yolcuların ve trafik kontrol servislerinin daha akıcı ve güvenli bir trafik hizmeti sunmasını sağlamaktadır (**U.S. Highway Administration, 1999**).

Kısa zamanlı trafik hızı tahmini yolcuların seyahat sürelerini en doğru şekilde öngörmelerini sağlamaktadır. Yola çıktıktan hedefe varıncaya kadar gidilecek rota üzerinde oluşabilecek trafik sıkışıklığını önceden öngörülmesiyle kullanıcılar başka rotalara kaydırılması sağlanmakta ve trafiğin daha homojen yayılması sağlanabilmektedir. Böylece insanların trafikte geçirdikleri sürenin en aza indirilmesi hedeflenmektedir.

Trafik uzun süreli gözleendiğinde günün belirli saatlerinde birbirine benzer hız eğrileri gözlenmektedir. Bu hız eğrilerinin iniş çıkışları genelde insanların günlük rutinlerini yansıtmaktadır. Örneğin genelde yollarda sabah saat 7 ile 9 arasında insanların işlerine gitmesi nedeniyle bir yoğunluk oluşmakta, yine akşam 18 ile 20 saatleri arasında işten dönüş zamanı olması nedeniyle tekrar bir yoğunluk yaşanmaktadır. Bu beklenen trafik yoğunluğu dışında yolun çeşitli özelliklerinden doğan rutin sıkışıklıklar oluşmaktadır. Örneğin taraftarı fazla olan bir takımın stad yolları periyodik olarak maç zamanı tıkanıklık yaşamaktadır. Bunun dışında genelde yollarda hafta içi trafik sıkışıklığı daha fazla olmaktadırken hafta sonları çalışan insanların sayısının azalması ve insanların daha

çok kendi ihtiyaçları için trafiğe çıkması sonucu trafiğin yoğunluk miktarı ve saatleri değişmektedir (**Chrobok ve diğ., 2000**). Beklenen trafik sıkışıklıkları dışında oluşan kaza, yol yapım çalışması gibi beklenmeyen çeşitli olaylar neticesinde yine trafikte bekleme süreleri artmaktadır. Bunun yanı sıra binlerce öğrencinin olduğu bir şehirde okulların açık kalma zamanlarına göre mevsimsel olarak trafiğin durumu değişmektedir. Ayrıca görüş mesafesinin fazla olduğu güneşli bir günde trafiğin açık olduğu görülürken, yağmurlu bir günde fren mesafesinin artması ve görüşün azalması gibi etkiler nedeniyle kullanıcılar daha güvenli bir şekilde hedeflerine ulaşmak için hızlarını düşürmekte, bu da trafiğin daha ağır ilerlemesine neden olmaktadır.

Görüldüğü gibi bir çok dış etken trafiğin seyir durumunu değiştirmektedir. İnsanlar evlerinden çıkıp varmak istedikleri yere gitmek istediklerinde onlarca dış faktör onların trafikte geçirecekleri zamanı etkilemektedir. Bu noktada trafik hızı öngörüsü yapabilmek için ilk yapılması gereken, araçların anlık hız bilgisini ölçmektir. Trafik hızı ölçümü için geliştirilen bir çok cihaz ile anlık hız ölçümü yapılabilmektedir. Bu cihazların başlıcaları Remote Traffic Microwave Sensor (RTMS) cihazları (**Petty ve diğ., 1998**), kılavuz araçlar (**Chen ve Chien, 2004**), yeniden araç tanınabilmesi (**Zhang ve Rice, 2001**) ve video kameralardır (**Balcılar ve Sönmez, 2008**). RTMS cihazları yol kenarında çeşitli yüksek yerlere konularak hızı ölçülen cihazlardır (**EIS Electronic, 2003**). Kılavuz araçlar belirli periyotlarla trafik hızı ölçülmek istenilen yolda seyir eden araçlardır. Bu cihazlardaki GPRS sistemi sayesinde her noktadaki hız ölçülmektedir (**Chen ve Chien, 2004**).

Yolun çeşitli noktadaki anlık hız bulunduktan yol uzunluğunun hıza bölünmesiyle oluşan seyahat süresinin hesaplanabilir. Fakat trafiğin durumunun yola çıktıktan varış noktasına ulaşmaya kadar değişmesi yola çıkıldığında alınan hız bilgileriyle hesaplanan tahmini varış süresi yanıltıcı olmasına neden olmaktadır. Bu yüzden tahmini varış süresini hesaplamak için, seyahat süresince oluşabilecek tahmini hızların hesaplaması zorunludur.

Seyahat süresince oluşabilecek tahmini hızları hesaplamak için çeşitli yöntemler ortaya çıkmıştır. Bu amaçla ilk denenilen yöntemlerden biri doğrusal yöntemdir (**Rice ve Zwet, 2001; Sun ve diğ., 2003; Van Lint ve diğ., 2000; Zhang ve diğ., 2001**). Doğrusal

yöntemlerin hız değerleri arasındaki karmaşık ilişkiyi tam olarak verememesinden dolayı yapılan tahminler 15 dakikadan kısa periyotlarda iyi sonuç vermekte, daha uzun zamanda yapılan tahminlerde ilgili zaman diliminde geçmiş zamanın ortalaması alınarak yapılan ölçümler daha iyi sonuç vermektedir (**Kwon ve diğ., 2000; Hobeika ve diğ., 1994**). Yine trafik öngörüsünde kullanılan ve zamana bağlı parametreler kullanan Kalman filtresi yöntemi yine ileriki zaman dilimi tahminlerinde hatalı sonuçlar verebilmektedir (**Park ve Rilett, 1999**).

Problemin karmaşıklığı göz önüne alınarak bir çok çalışmada SVM (Support Vector Machine, Karar Destek Makinesi) ve ANN (Artificial Neural Network, Yapay Sinir Ağları) yöntemleri, bu yöntemlerin belirli bir fonksiyon şeklinde olmaması ve başlangıçta belirli parametrelere ilk değer verme zorunluğu olmaması özellikleri nedeniyle kullanılmıştır. ANN'ler insan beyninin hesaplama yapısını modelleyerek çok katmanlı bir hesaplama birimi oluşturmaktır ve trafik hızı öngörüsü bir çok çalışmada kullanılmıştır (**Lingras ve Mountford, 2001; Mark ve Sadek, 2004; Park ve Rilett, 1999; Van Lint ve diğ., 2000**). Fakat SVM'de kullanılan yapısal risk küçültme yöntemi (SRM), ANN'de kullanılan deneysel risk küçültme (ERM) yönetimine göre, yerel sınırlara takılmaması ve genel değerleri daha iyi yakalaması nedeniyle daha iyi sonuçlar vermektedir (**Wu ve diğ., 2004; Bin ve diğ., 2006**).

Bu çalışmada SVM ve kNN (k-En Yakın Komşu) yöntemleri ve İstanbul'un 327 noktasından 45 gün boyunca toplanılan hız verileri kullanılarak kısa zamanlı hız tahmini yapılmaktadır. Hem aynı noktada yapılan ölçümler, hem de o noktaya bağlanan yollardan alınan ölçümler ve öznitelik seçme yöntemleri kullanılarak hız tahmini yapılmıştır. Tezin geri kalanı şu şekilde düzenlenmiştir. 2. bölümde akıllı ulaşım sistemleri ve trafik hızı öngörüsü için geçmişte yapılan çalışmalar anlatılmıştır. 3. bölümde kullanılan veri kümesi ve bu kümenin nasıl önişlendiği hakkında bilgi verilmiştir. 4. bölümde kullanılan örüntü tanıma yöntemleri ve öznitelik seçme yönteminden bahsedilmiştir. 5. bölüm bir ve birden fazla noktadan alınan hız bilgileriyle yapılan trafik hızı öngörüsü deney sonuçlarını içermektedir. 6. Bölümde ise genel olarak varılan sonuçları ve gelecekte yapılabilecek çalışmalar anlatılmıştır.

## **2. AKILLI ULAŞIM SİSTEMLERİ VE GEÇMİŞ ÇALIŞMALAR**

### **2.1 Akıllı Ulaşım Sistemleri**

Şehirlerdeki araç sayısının giderek artmasıyla beraber trafik sıkışıklığı artmaya başlamış, kaza sayısında artış gözlenmiş ve mali açıdan bir çok kayıp meydana gelmiştir. Bu sorunu çözmek için akıllı ulaşım sistemleri kavramı ortaya atılmış ve trafiği çeşitli bilimsel yöntemlerle gözleyerek onun düzenlenebileceği savı üzerine bir çok sistem hayata geçirilmiştir. Akıllı trafik sistemleri (ITS) trafik sinyal kontrol sistemi, kaza yönetim sistemi gibi bir çok uygulamayı yolcuların ve trafik kontrol servislerinin hizmetine sunarak daha akıcı ve güvenli bir trafik hizmeti sunmasını sağlamaktadır. Yol durumu hakkında gerçek zamanlı bilgilendirme, acil durumlarda hızlı müdahale ve yaşam kalitesinin artması için bu servisler desteklenmektedir. Bu sistemleri 1984 yılından beri kullanan California kenti %13 yakıt tasarrufu sağlamış, %41 araç duruş sayısını azaltmış, %18 seyahat etme süresini indirmiş, %16 ortalama seyahat hızını arttırmış ve trafikte gecikme oranını %44 indirmiştir (**U.S. Highway Administration, 1999**).

#### **2.1.1 Trafik Sinyal Kontrol Sistemi**

Trafik sinyallerinin trafiğin durumuna göre otomatik değişmesini hedeflemektedir. Trafik ışıkları önüne konan kameralar ve diğer cihazlar yardımıyla trafik durumu gerçek zamanlı olarak izlenmekte ve durum değişikliklerinde trafik ışıklarının zamanlaması değiştirilmektedir. Böylece yollardaki araç akışı sürekli hale getirilmeye çalışılmaktadır. San Diegoda kurulu SCOOT sistemi sayesinde yoğun saatlerde trafik yoğunluğu %11 azaltılmıştır (**U.S. Highway Administration, 1999**).

### **2.1.2 Transit Yollar Yönetim Sistemi**

Araçlara konulan OGS tarzı cihazlarla radyo dalgalarıyla araç kimliği tespit edilmekte ve ücretlendirme otomatik olarak yapılmaktadır. Böylece para ödeme gişelerinde araç kuyrukları oluşmamaktadır (U.S. Highway Administration, 1999).

### **2.1.3 Kaza Yönetim Sistemi**

Kaza olduğunu araçların normal hızlarındaki değişime göre algılamakta ve kaza olan yere gerekli araçları yollamaktadır (U.S. Highway Administration, 1999). Ayrıca kaza olan yerde trafiği rahatlatmak için gerekli uyarıları ve yönlendirmeleri yapmaktadır. Kaza olan yerleri istatistiksel olarak işaretleyerek kaza olma riski yüksek yerleri belirlemektedir. Sistem şu an San Fransisco kentinde kullanılmaktadır. Kullanıcı bilgilendirme sistemi sayesinde sürücüler bilmedikleri yollarda daha az zaman harcayarak hedeflerine ulaşmaktadır (U.S. Highway Administration, 1999). Ayrıca internet ve çeşitli telekomünikasyon araçlarıyla on-line olarak yol durumunu görebilmektedir. Acil durumlarda (deprem vs.) bu gibi bilgilendirmeler çok daha hayati olabilmektedir.

### **2.1.4 Toplu Taşıma Araçları Yönetim Sistemi**

Toplu taşıma araçlarının durumlarını ve yolcuların aktivitelerini gözetleyen bir sistemdir (U.S. Highway Administration, 1999). Toplu taşıma araçlarının otomatik olarak yerlerinin belirlenmesi sonucu olası problemlere müdahale edilebilir.

## **2.2 Kısa Zamanlı Hız Tahmini Konusunda Yapılmış Çalışmalar**

Kısa zaman sonra hız öngörüsü üzerine bir çok çalışma bulunmaktadır. Aşağıdaki bölümlerde bu çalışmalarda kullanılan yöntemler hakkında detaylar verilmiştir.

### **2.2.1 Doğrusal Regresyon**

Doğrusal regresyon yöntemleri trafik zaman öngörüsü çalışmalarında ilk kullanılan yöntemlerdendir (Rice ve Zwet, 2001; Sun ve diğ., 2003; Van Lint ve diğ., 2000; Zhang ve diğ., 2001). Şu anki hız değeri ile gelecekte oluşacak hız değeri arasında

doğrusal bir bağlantı olduğu varsayımına göre çalışmaktadır. Aradaki doğrusal ilişki bulunurken geçmiş hız değerleri kullanılır.

(2.1)'deki gibi doğrusal fonksiyon kullanılarak gelecekteki hız tahmin edilmek istenmektedir.

$$V(t + \Delta t) = \alpha(t, \Delta t)V(t) + \beta(t, \Delta t) + \varepsilon \quad (2.1)$$

Burada  $V(t)$  şu anki seyahat hızı, alfa ve beta ve epsilon zamanla değişen katsayılarıdır.  $V(t + \Delta t)$  ise  $\Delta t$  dakika sonra tahmin edilmek istenen hız değeridir. Burada alfa ve beta değerlerini ölçülmek istenen zamana göre en doğru şekilde bulmak önem kazanmaktadır ve bu bir minimizasyon yöntemini kullanarak başarılabilir (**Rice ve Zwet, 2001**).

Doğrusal regresyon yöntemlerinin temel zaafı şu anki hız ile gelecekteki hız arasındaki ilişki her zaman doğrusal olmadığından dolayı genellikle 15-20 dakika aralığı gibi kısa zamanlı hız öngörülerinde başarılı olmalarıdır. Daha uzun zaman aralıklarında yapılan çalışmalarda tarihsel ortalamalar daha başarılı sonuçlar vermiştir (**Kwon ve diğ., 2000; Hobeika ve diğ., 1994**).

### 2.1.2 Kalman Süzgeci

Kalman süzgeci veri kaybı ve gürültüsü çok olan dinamik sistemler için tasarlanmış etkili, kendini tekrarlayan bir süzgeçtir. Kendini tekrarlayan bir süzgeç olması nedeniyle şu anki durum sadece bir önceki duruma ve şu anki ölçümlere dayanmaktadır. Doğrusal en küçük kareler yöntemini kullanarak en uygun parametreleri bulmaya çalışmaktadır. Trafik hızı öngörüsünün yapısı bu tanıma uygun olduğundan bazı çalışmalar bunu kullanmıştır (**Sheng, 2002; Okutani ve Stephanedes, 1984**). Fakat bu metotta doğrusal yöntemdeki gibi uzun zaman tahminlerinde yanılma payı fazla olmaktadır. 1-2 kendini tekrarlayan döngü (10-15 dakika) için ise oldukça iyi sonuçlar vermektedir (**Park ve Ritett, 1998**).

### 2.1.3 Benzetim

Benzetim trafik hızı bulma yöntemlerinde bir benzetim sistemi tasarlanarak, trafik yol durumu gerçek zamanlı girilerek geleceğe dair hız/zaman tahmini yapılmaktadır

(Chrobok, 2001; Wahle ve diğ., 2000). Ajan tabanlı yazılımlarda bu kategoride değerlendirilebilir (Zhang ve Levinson, 2004). Belirli konumlarda ölçülen hız ve araç sayısı değerleri daha önce sistem üzerinde oluşturulmuş yollara girilmekte ve araçlar gerçek zamanlı olarak benzetim gerçekleştirilmektedir (Chrobok, 2001). Ayrıca geleceğe yönelik tahmin yapmak için geçmiş hız değerlerinden de yararlanılmaktadır. Bunun için yollarda geçen araç sayısı – hız bilgisi günlük ve mevsimsel olarak kayıt edilmekte ve bunlar sürekli güncellenmektedir. Günlük olarak trafik durumunu gruplarken benzer araç trafiği gösterdiği için pazartesi – perşembe, cuma, cumartesi ve pazar günleri kendi içinde gruplandırılmaktadır. Mevsimsel olarak araç trafiğini çıkarmak için her aya trafik yoğunluğuna gösteren bir katsayı verilmektedir. Maç gibi özel durumlarda trafik durumunun normal seyirinde seyretmeyeceği varsayılarak ayrıca sisteme girilmektedir. Benzetimde oluşturulan yolların gerçek yollarla birebir aynı özellikleri göstermesi başarıyı arttırmaktadır. Bu özellikleri sağlaması için yollar, kavşaklar ve geçiş yolları da benzetimle gerçekleştirilmektedir. Simüle edilen her araç yolun özelliklerine göre hızlanabilmekte, diğer yollara geçiş yapabilmekte ve önünde bulunan araçların durması durumunda yavaşlayabilecek şekilde programlanmaktadır. Benzetim sistemlerinin avantajı eğer veriler doğru hesaplanıp girilirse herhangi bir anda yol üzerindeki herhangi bir noktanın yol durumunu göstermesidir. Benzetim sistemlerinin kötü tarafı ise hesaplanacak çok fazla parametre olması ve bilgilerin gerçek zamanlı işlenmesi gerektiğinden hızlı sistemlere ihtiyaç duymasıdır.

#### 2.1.4 Neural Network

ANN (Alpaydın, 2004) insan beyninin hesaplama yapısını modelleyerek çok katmanlı bir hesaplama birimi oluşturmaktadır. Giriş çıkış katmanları arasında gizli katmanlar konularak örüntü üzerindeki karmaşık ilişkiler öğrenilmeye çalışılır. ANN, SVM’de olduğu gibi istatistik yöntemlerinin tersine hesaplanması gereken nokta sayısı arttığında işlem zamanı da ona paralel olarak artmaz, çok daha az bir hesaplama gücü ile sonuca ulaşılabilir (Mark ve Sadek, 2004). Fakat SVM’e göre kötü tarafı kayıp verilere karşı daha duyarlıdır ve veri eksikliği durumunda sonuçlar kötüleşir (Wu ve diğ., 2004). Ayrıca ANN’ye başlangıç değeri verme zorunluluğu farklı başlangıç değerleri verildiğinde yerel sınırlara takıldığından farklı sonuçların oluşmasına neden olmaktadır. Ayrıca ANN ile

çalıřırken deęiřkenlere ilk deęerlerin verilmesi, gizli katmanlarının boyutlarının ayarlanması ve öğrenme oranının belirlenmesi önemlidir. **Wu ve dię. (2004), Bin ve dię. (2006)** çalıřmalarında SVM'in ANN'ye göre daha iyi sonuç verdięi gösterilmiřtir.

### **2.1.5 Karma Yöntemler**

Çeřitli yöntemleri ortak kullanarak trafik hızı öngörüsünde daha iyi sonuç elde edilmek istenmiřtir. Örnek olarak **Sheng (2002)** çalıřmasında kalman süzgeçi ve ANN yöntemleri ortak kullanılarak daha iyi başarıml sağlandıęı belirtilmiřtir. **Zou ve dię. (2007)** çalıřmasında ise doğrusal regresyon yöntemiyle geçmiş zamanın ortalamasının birleřimi kullanılmıřtır.

### 3. VERİ KÜMESİ VE VERİ HAZIRLAMA

#### 3.1 Veri Kümesi

##### 3.1.1 Trafik Öngörüsünde Kullanılan Veri Kaynakları

Trafik hızı öngörüsü için temel veri bir noktada araçların geçmiş hız değerleridir. Araçların geçmiş hız değerleri çeşitli şekillerde ölçülmekte ve kayıt edilmektedir. Hız değerlerini ölçmek için kullanılan cihazlar tek-çevrim dedektörler, çift-çevrim dedektörler **Petty ve diğ. (1998)**, kameralar **Balcılar ve Sönmez (2008)** ve vekil araçlardır **Chen ve Chien (2004)**. Kamera kullanılması durumunda çeşitli görüntü işleme yöntemleri kullanılarak cihazların hızları tespit edilir. Aynı zamanda güvenlik ve çeşitli gözetleme işlemleri içinde yolların kenarına sıkça yerleştirilen kameralardan elde edilen verilerle hız tespit edilmesi yararlı olmaktadır. Çift-çevrim dedektörler belirli süre aralıklarla trafik hızını ölçüp ana merkeze gönderen cihazlardır. Tek-çevrim dedektörler, Çift-çevrim dedektörler farklı olarak yalnızca akış ve yoğunluğu ölçmektedir (**Petty ve diğ., 1998**). Akış ve yoğunluk kullanılarak araçların ortalama hızı hesaplanmaktadır. Trafik hızını tespit etmek için ilk kullanılan dedektörlerden olduğundan eski alt yapıların çoğunda bulunmaktadır. Bu yüzden veriler bu tip dedektörlerden alındığında hızları elde etmek için bir ön işleme gerekmektedir (**Wang ve Nihan, 2003**).

Trafik hızı öngörüsü yapıldığında öngörü yapılan yol boyunca olabildiğince çok noktada hız ölçümü yapılması yararlı olmaktadır. Sensör aralıklarının fazla olduğu yerlerdeki hız değişimleri çok hassas şekilde hesaplanamayacağından tahmin edilen hız değerleri ile gerçekte oluşacak hız değerleri arasında fark oluşabilmektedir. Bu sebepten olabildiğince çok noktadan hız verisi alınması başarıyı arttıracaktır.

Trafik hızı öngörüsünde veri toplamak için kullanılan bir diğer yöntem ise gezici araçlardır. Şehrin çeşitli noktalarında gezen araçlar sürekli merkeze yerlerini GPRS aracılığıyla göndermekte gönderilen değerler kullanılarak araçların geçtiği noktalardaki anlık trafik hızı hesaplanmaktadır. Gezici araçla trafik hızı ölçüm yöntemi diğer hız

ölçüm yöntemlerine göre daha masraflı olduğundan çok fazla tercih edilmemektedir (**Chen ve Chien, 2004**). Ayrıca benzer şekilde araçları tanıyabilen algılayıcılarda hız tespitinde kullanılabilir. 2 ayrı noktaya yerleştirilen bu cihazlar ile araçların 2 algılayıcı arasındaki geçiş zamanı kullanarak araçların hızları hesaplanmaktadır (**Petty ve diğ., 1998**). Bu yöntemin araçları tanıma konusunda başarımı yolun durumuna göre değişiklik gösterdiğinden ve maliyeti çok yüksek olduğundan kullanım oranı azdır.

### 3.1.2 Çalışmada Kullanılan Veri Kaynağı

Bu çalışmada kullanılan veriler İstanbul Büyükşehir Belediyesi Trafik Kontrol Merkezinin (IBB-TKM) sayfasından alınmaktadır. Trafik kontrol merkezi bütün İstanbul'a yayılmış 327 RTMS (Remote Traffic Microwave Sensor) ile ilgili şeritlerden geçen araçların hız ortalamalarını aralıkla 1-2 dakika aralıklarla web sitesinden yayınlamaktadır. RTMS cihazları çift çevrim sensörlerin bir çeşididir (**EIS Electronic, 2003**). Trafik sensörlerinin sıkça doğru sonuç vermemesine neden olan sensör açığı değişimleri ve önüne sensörü engelleyici yabancı cisimlerin gelmesi (ağaç dalı vs.) gibi durumlar Trafik Kontrol Merkezi çalışanlarının yaptığı düzenli bakım çalışmaları sayesinde önlenmektedir.

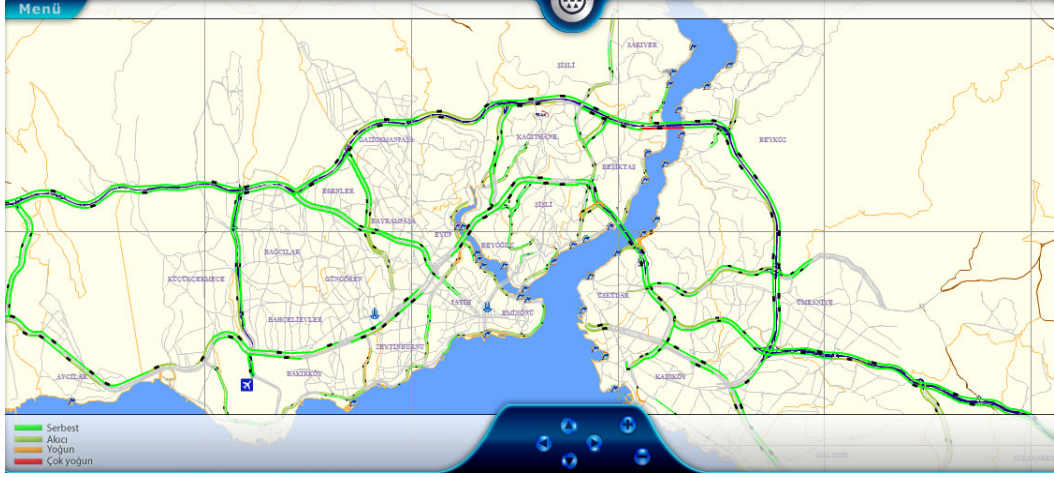
Trafik kontrol merkezi sayfasında bulunan canlandırma sayesinde yolların şu anki durumu görsel olarak öğrenilebilmektedir. Şekil 3.1'de yeşil olan yollar trafiğin akıcı olduğu yerleri belirtmekte iken kırmızı tonlara yaklaştıkça trafik yoğunluğunun arttığı belirtilmektedir.

<http://tkm.ibb.gov.tr/data/newDM.aspx> adresinde yayınlanan anlık hız bilgilerinin yapısı aşağıdaki gibidir.

```
s2t=23:57&s2d1=105&s2d2=105&s2d3=101&ts2d1=&ts2d2=&ts2d3=&s3t=23:57&s3d1=86&s3d2=88&s3d3=84&ts3d1=82&ts3d2=80&ts3d3=78&s4t=23:56&s4d1=105&s4d2=100&s4d3=99&ts4d1=&ts4d2=&ts4d3=&s5t=23:56&s5d1=105&s5d2=76&s5d3=112&ts5d1=&ts5d2=&ts5d3=&s6t=23:56&s6d1=82&s6d2=89&s6d3=82&ts6d1=&ts6d2=
```

Bu sayfadan 327 RTMS'ten sağlıklı veri alınabilenlerin anlık hız bilgileri yayınlanmaktadır. Sadece bir RTMS için gönderilen veri ise aşağıdaki şekildedir.

```
(s2)t=23:57&(s2)d1=105&(s2)d2=105&(s2)d3=101
```



**Şekil 3.1:** IBB Trafik Kontrol Merkezi Sayfasında Trafiğin Anlık Durumunu Gösteren canlandırma (IBB-TKM, 2008).

Parantez içindeki kısım RTMS numarasını göstermektedir. Her bir RTMS “s” veya “ts” ile başlayan bir kod numarası ile kodlanmıştır. “sX” RTMS ile “tsX” RTMS konum olarak aynı yerde bulunmaktadır, fakat “tsX”, “sX” göre ters şeritin hız bilgilerini ölçmektedir. Buna göre “s2” nolu RTMS ile “ts2” nolu RTMS aynı yerde karşıt şeritlerin hız bilgilerini ölçmektedir.

RTMS numarasından sonra bulunabilecek işaretler ise t,d1,d2,d3’dür. RTMS numarası, “t” ile belirtilirse kendisinden sonra RTMS ölçüm zamanı verileceği, d1 ile verilirse RTMS’den t zamanındaki ölçülen hız bilgisinin verileceği, d2 ile verilirse (t-1) dakika önceki hız bilgisinin verileceği, d3 ile verilirse (t - 2) dakika önceki hız bilgisinin verileceği anlaşılır.

IBB-TKM web sitesindeki bilgiler dakikada bir güncellendiğinden her dakika başı çalıştırılan bir programla bu bilgiler çekilerek çalışmada kullanılan veri biçimin çevrilip, saklanmıştır. Çalışma için her bir RTMS’in dakikalık hız bilgileri toplam 45 gün boyunca çekilmiş ve veritabanında saklanmıştır.

RTMS’lerin alınan bilgileri aşağıdaki biçimde saklanmıştır.

ID:Kayıt numarası

RTMSNO:Verinin hangi RTMS’den alındığı

HIZ:İlgili RTMS için ilgili tarihte alınan hız bilgisi

TARİH:Hızın ölçüldüğü tarih

01.12.2007 tarihi ile 15.01.2008 tarihleri arasında alınan veriler yukarıda belirtilen biçimde saklanmış ve çalışma bu veriler üzerinden yürütülmüştür. Toplam 33351102 kayıtlık bir veritabanı oluşmuştur. Örnek bir kayıt bloğu Tablo 3.1’de gösterilmiştir.

**Tablo 3.1:** Örnek Kayıt Bloğu

Id	SensorNo	Hız	Tarih
2729	s102	112	2007-12-12 13:31:50.520
3036	s102	110	2007-12-12 13:32:51.457
3364	s102	112	2007-12-12 13:32:10.787
3682	s102	110	2007-12-12 13:33:50.520
4066	s102	110	2007-12-12 13:33:10.800
4520	s102	110	2007-12-12 13:34:10.940
4667	s102	107	2007-12-12 13:34:50.537
5236	s102	107	2007-12-12 13:35:10.787
5313	s102	107	2007-12-12 13:35:50.067
5620	s102	109	2007-12-12 13:36:52.520
5940	s102	107	2007-12-12 13:36:10.833
6266	s102	109	2007-12-12 13:37:50.223
6642	s102	109	2007-12-12 13:37:10.770

Veritabanında kullanılan veri saklama biçimi olabildiğince sade tutulmaya çalışılmıştır. Bunun nedeni çalışmanın herhangi bir veri kaynağından alınan verilere çevrilebilmesinin sağlanmasıdır. Diğer veri kaynakları bölümünde görülebileceği gibi İstanbul kenti alınabilen hız-konum bilgileri, dünyanın bir çok kenti içinde alınabilmektedir. Bu çalışmada kullanılan yöntemler, diğer kentlerden hız bilgileri alındığı takdirde kullanılabilir. Aynı zamanda veri hazırlama bölümünde anlatılacağı gibi bu alınan verilerin çeşitli yöntemlerle işlenebilmesi için her yöntem için

belirli bir biçime çevrilmesi gerekmektedir, verilerin bu temel şekilde saklanması sonucu herhangi bir başka biçime çevrilmesi için temel teşkil etmektedir.

### 3.1.3 Diğer Veri Kaynakları

Dünyanın çeşitli noktalarında, IBB-TKM sitesinden alınabildiği gibi, canlı trafik durumunu yayınlayan web siteleri bulunmaktadır. Bu çalışmada IBB-TKM dışında alınan bir veri kullanılmamıştır. Başka bir veri kaynağının bilgilerin kullanılmaması nedeni her yolun kendine has özellikleri olmasıdır. Aynı yolun bile çeşitli tarihlerde hız durumu farklılık göstermektedir. Bu nedenle farklı yollardan alınan veriler çalışmada kullanılsa bile bunların karşılaştırılmasının bir anlamı olmayacaktır. Fakat aşağıda belirtilen kaynaklardan da canlı trafik bilgileri alınarak trafik hız ve zaman öngörüsü yapılabilir. Aynı veri biçimine çevrilerek ve aynı örüntü tanıma yöntemleri kullanılarak geleceğe dair hız tahmini yapılabilecektir.

#### 3.1.3.1 BeatTheTraffic

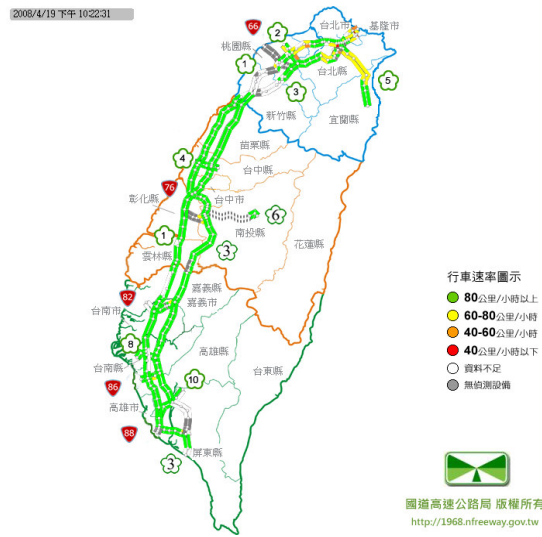
Amerika birleşik Devletleri'nin otuzü aşkın kentinin canlı trafik bilgisi bu sayfadan yayınlanmakta ve görsel olarak kullanıcıların erişimine sunulmuştur.



Şekil 3.2: Beat the traffic sayfasından bir görüntü (Beattetraffic, 2008).

### 3.1.3.2 Tayvan Uluslararası Yollar Yönetim Merkezi (TANFB)

Tayvan uluslararası yollar yönetim merkezi (Taiwan Area National Freeway Bureau - TANFB) sayfasında canlı yol bilgileri alınabilir. Yine yol bilgileri resim üzerinde çeşitli renk kodları ile yol durumunun ne durumda olduğu belirtilmiştir. **Wu ve diğ. (2004)** çalışması buradaki verileri kullanarak yapılmıştır. Fakat buradaki veriler internet üzerinden doğrudan erişilememekte, özel izin ile alınabilmektedir.



Şekil 3.3: TANFB'den alınan anlık yol durum bilgisi (TANFB, 2008).

### 3.2 Veri Hazırlama

Çalışmada kullanılan verilerin SVM ve KNN yöntemlerinin işleyebilecek hale gelmesi için veritabanına saklanan verilerin belirli bir biçime çevrilmesi gerekmektedir. Bu biçim tek bir sensör ve eğitim kümesi için aşağıdaki şekildedir.

$$V_x(t-s[n]) \ V_x(t-s[n-1]) \ \dots \ V_x(t-s[2]) \ V_x(t-s[1]) \ V_x(t-s[0])$$

$V_x(t)$  : x sensörü için t anında ölçülen hız

s : Özniteliklerin hangi zaman aralığından alınacağını belirleyen bir zaman serisi

s serisi'nin örnek bir serisi (0, -5, -10, -15, -20, -25, -30) şeklindedir. t anı ve seri halinde ondan beşer dakikalık önceki hızlar olmak üzere toplam 7 andaki hız eğitim kümesinin bir satırı olarak alınmaktadır. Diğer satırlarda t'nin belirli aralıklarla değiştirilerek oluşturulur. Eğer satırlar için t'nin değiştirilme aralığı 5 dakika verilirse, başlangıç t

zamanı ve örnek S serisi için oluşturulacak eğitim kümesinin genel yapısı aşağıdaki gibi oluşur.

$$V_x(t-30) V_x(t-25) V_x(t-20) V_x(t-15) V_x(t-10) V_x(t-5) V_x(t)$$

$$V_x(t-35) V_x(t-30) V_x(t-25) V_x(t-20) V_x(t-15) V_x(t-10) V_x(t-5)$$

$$V_x(t-40) V_x(t-35) V_x(t-30) V_x(t-25) V_x(t-20) V_x(t-15) V_x(t-10)$$

$$V_x(t-45) V_x(t-40) V_x(t-35) V_x(t-30) V_x(t-25) V_x(t-20) V_x(t-15)$$

s serisinin boyu aynı zamanda öznitelik sayısını vermektedir. Öznitelik sayısı n olarak verilirse trafik hızını tahmin için kullanılan çeşitli algoritmaların temel olarak yaptığı ilk n-1 öznitelik vektörünü kullanarak n'inci öznitelik vektörünü tahmin etmek için bir öğrenme kümesi oluşturmaktır.

Birden çok sensörün kullanıldığında öğrenme işlemlerinde kullanılan biçim ise aşağıdaki şekildedir.

$$V_{s[n,0]}(t-s[n,1]) V_{s[n-1,0]}(t-s[n-1,1]) \dots V_{s[2,0]}(t-s[2,1]) V_{s[1,0]}(t-s[1,1]) V_{s[1,0]}(t-s[0,1])$$

$V_{s[n,1]}(t)$  : ilgili sensör için t anında ölçülen hız. Sensör numarası S serisinden alınacaktır.

s : Özniteliklerin hangi sensörden ve hangi zaman aralığından alınacağını belirleyen bir zaman serisidir.

S serisi'nin örnek bir serisi ([s2:0], [s2:-5], [s2:-10], [s2:-15], [s2:-20], [s4:-5], [s4:-10], [s4:-15], [s4:-20]) şeklinde olabilir. Tek sensörün kullandığı seriden farklı olarak hangi sensörün verilerinin kullanılacağı bilgisi seri içinde yer alır. Yine satırlar için t başlangıç zamanı farkı 5 dakika verilirse örnek s serisi için oluşacak eğitim kümesinin yapısı aşağıdaki şekildedir.

$$V_{s4}(t-20) V_{s4}(t-15) V_{s4}(t-10) V_{s4}(t-5) V_{s2}(t-20) V_{s2}(t-15) V_{s2}(t-10) V_{s2}(t-5) V_{s2}(t)$$

$$V_{s4}(t-25) V_{s4}(t-20) V_{s4}(t-15) V_{s4}(t-10) V_{s2}(t-25) V_{s2}(t-20) V_{s2}(t-15) V_{s2}(t-10) V_{s2}(t-5)$$

$$V_{s4}(t-30) V_{s4}(t-25) V_{s4}(t-20) V_{s4}(t-15) V_{s2}(t-30) V_{s2}(t-25) V_{s2}(t-20) V_{s2}(t-15) V_{s2}(t-10)$$

$$V_{s4}(t-35) V_{s4}(t-30) V_{s4}(t-25) V_{s4}(t-20) V_{s2}(t-35) V_{s2}(t-30) V_{s2}(t-25) V_{s2}(t-20) V_{s2}(t-15)$$

15 özellikli örnek bir eğitim kümesinin görüntüsü tablo 3.2'deki gibi oluşmaktadır.

**Tablo 3.2:** Eğitim Kümesi Görüntüsü

v(t-10080)	v(t-1440)	v(t-60)	v(t-55)	v(t-50)	v(t-45)	v(t-40)	v(t-35)	v(t-30)	v(t-25)	v(t-20)	v(t-15)	v(t-10)	v(t-5)	V(t)
120	114	113	114	113	111	110	113	112	113	112	113	116	113	112
116	118	112	113	114	113	111	110	113	112	113	112	113	116	113
110	115	109	112	113	114	113	111	110	113	112	113	112	113	116
112	115	107	109	112	113	114	113	111	110	113	112	113	112	113
112	117	107	107	109	112	113	114	113	111	110	113	112	113	112
110	117	109	107	107	109	112	113	114	113	111	110	113	112	113
110	110	113	109	107	107	109	112	113	114	113	111	110	113	112
113	108	111	113	109	107	107	109	112	113	114	113	111	110	113
116	111	116	111	113	109	107	107	109	112	113	114	113	111	110

Eğitim yapıldıktan sonra hız tahmini yapılmasını istediğimiz andan önceki verileri vererek hız tahmini yapılmaktadır. Bu veri kümesi test veri kümesidir. Eğitim kümesiyle eğitilmiş hız öngörüsü yöntemi test verisini alarak sonra oluşacak değeri hesaplamaya çalışır. Test veri kümesinin yapısı aşağıdaki şekildedir.

$$V_x(t-s[n]) \ V_x(t-s[n-1]) \ \dots \ V_x(t-s[2]) \ V_x(t-s[1])$$

Test kümesinde kullanılan s serisi eğitim sırasında kullanılan s serisi ile aynı olmak zorundadır. Tek farkı ölçülmek istenen andaki hızı bilemediğimizden bunu vermemiş olmamızdır. (0, -5, -10, -15, -20, -25, -30) örnek s serisi için test veri kümesinin yapısı aşağıdaki şekildedir.

$$V_x(t-30) \ V_x(t-25) \ V_x(t-20) \ V_x(t-15) \ V_x(t-10) \ V_x(t-5)$$

s serisinin yapısı değiştirilerek tahmin yapılmak istenen zaman aralığı değiştirilebilir. Tahmin yapılmak istenen zaman s serisinin sıfıncı ile birinci elemanın farkından elde edilmektedir. Örnek s serimizde bu zaman farkı 5 olduğundan, bu s serisiyle 5 dakika

sonraki hızın tahmin edilmek istendiği anlaşılır. 60 dakika sonraki hızı ölçmek istediğimizde örnek s serisinin yapısı şu şekilde olabilir: (0, -65, -70, -75, -80, -85, -90).

s serisinin birinci indeksinden sonraki elemanlar ilgili sensör için en iyi sonuç vermiş zaman aralıklarından seçilir. Öznitelikleri oluşturan bu sayılar ilgili sensörün bulunduğu yolun durumuna değişiklik gösterebilmektedir. Sensörün bulunduğu konumdaki yol konumu görüldüğü s serisinin yapısını belirlemektedir. Öznitelik seçimi sırasında s serisinin yapısı da ortaya çıkmaktadır. Tabiki bunun her sensör için ayrı ayrı yapılması gerekmektedir. Ayrıca 2-3 sensörün ortak kullanıldığı durumlarda yine öznitelik seçimi bu sensörler için çalıştırılarak s serisi ortaya çıkartılmalıdır.

### 3.3 Veri Temizleme

Hız ölçüm sensörleri coğrafik olarak çok büyük bir alana yayıldığı ve açık alanda olduğundan dolayı dış etkilere mahruz kalabildiğinden çeşitli veri kayıpları oluşabilmektedir. Sorunun oluş biçimine göre etkisi de farklı şekillerde oluşmaktadır. Sıkça yaşanan sorunlardan biri sensör ile sensör verilerinin tutulduğu merkez arasında iletişimin kesilmesidir. Bu durumda IBB-TKM ilgili sensör ile ilgili verileri yayınlamamaktadır. Değerler kayıt edilirken ilgili sensör için sorun oluşan tarihler arasında bir bilgi kaydı oluşmamaktadır. Veri hazırlama esnasında eğitim kümesi oluşturulurken eğer sensör değerinin olmadığı veya 0 olduğu bir alana rastlanırsa bu alan ile ilgili bilgi eğitim sürecini yanıltabileceğinden alınmamaktadır. Eğer ilgili sensörün yakın yerlerinde benzer hız değerleri üreten başka bir sensör bulunuyorsa onun hız değerleri kullanılabilir.

Veri üzerinde oluşabilecek başka bir sorun ise sensörün hatalı ölçüm yapıyor olmasıdır. Sensörün açtığı değişimleri veya sensörün önüne onun ölçüm yapmasını engelleyici çeşitli cisimlerin gelmesi sonucu sensörün mevcut hızı doğru şekilde ölçmesini engellenecektir (EIS Electronic, 2003). Bu durumda sensör sürekli olarak aynı hız değerini üretebileceği gibi, değişik hız verilerini üretmeye devam edebilir. Şayet sensör sürekli aynı hız değerini çıktı olarak veriyorsa bunun gerçekte fiziksel olarak gerçekleşme olasılığının çok düşük olduğundan bu üretilen sonucu yanlış olarak değerlendirip işlem yapılmaktadır. Eğer sensör yanlış değerler ölçtüğü halde sürekli değişen değerler

veriyorsa bu durumun anlaşılması daha zordur. Öncelikle IBB-TKM’de kullanılan sensörler belirli aralıklarla bulunduğu coğrafi alan gidilerek ölçtüğü değerler ile gerçek değerler karşılaştırmak sureti ile verdiği değerlerin olabildiğince doğru değerler olması sağlanmaktadır. Fakat yine de sensör değerlerinin ne kadar doğru sonuç verdiği ondan önce ve sonra bağlanan yollardaki hız değerlerine bakılarak ölçülebilir.

### **3.4 Veri İşleme**

Eğitim ve test veri kümesi oluşturulurken alınan değerler beşer dakika aralıklarla alınarak öznitelikler oluşturulmaktadır. Fakat IBB-TKM’den alınan veriler dakika başı güncellenmektedir. Bu durumda 5 dakikalık verinin sadece 1 dakikası kullanılmakta diğer anlardaki hızlar eğitim ve test sürecinde kullanılamamaktadır. Bunu önlemek için t anındaki hız alınırken direk t anındaki hızı çıktı olarak vermek yerine t-2, t-1, t, t+1, t+2 anındaki hızlar alınarak bunların aritmetik ortalaması çıktı olarak verilmektedir. Böylece anlık hız artışlarının hatalı sonuç üretmesi önlenmektedir.

### **3.5 Birbirine Bağımlı Sensörlerin Bulunması**

Birbirlerine bağımlı sensörlerin korelasyon yöntemiyle bulunması için sensörlerden alınan hızlar aşağıdaki biçimde hazırlanmıştır.

s1\_0 s1\_5 s1\_10 s1\_15 s1\_20 s1\_25 s1\_30 s1\_35 s1\_40 s1\_45 s1\_50 s1\_55 s1\_60 s2\_0  
s2\_5 s2\_10 s2\_15 s2\_20 s2\_25 s2\_30 s2\_35 s2\_40 s2\_45 s2\_50.....

327 sensörün verileri yukarıdaki biçim sütunları oluşturacak şekilde hazırlanmıştır. Her sensörün 5 dakikadan 60 dakikaya kadar 13 farklı zaman farkındaki hız değerleri sütunlara yazılmıştır. Böylece 327 sensör \* 13 zaman bölgesi toplam 4251 sütundan oluşturulmuştur. Satırlar ise toplam 30 günlük ilgili sensörün ilgili zamanına karşılık gelen hız bilgisinden oluşmaktadır. 5 dakika aralıklarla oluşturan satırlar toplamda 8640 olmaktadır.

4251\*8640 boyutunda hazırlanan ilgili sensörlerin çeşitli zaman aralığında hız bilgisini içeren bu matriksin hazırlanmasındaki amaç bu sensörlerin birbirlerine ne kadar bağımlı olduklarını bulmak veya hangi sensörlerinin birbirlerine benzer hız özelliklerini verdiklerini görmektir.

Matlab'ta "corr" komutuyla ilgili sensörlerden birbirleriyle ilişkili olanlar bulacak ilişki matrisi elde edilmiştir.

Matlab'ta "corr" komutuyla bulunan 2 sensör arasında bağıntıyı bulan matematiksel formül (3.1)'deki gibidir.

$$ss_{xy} = \sum (x_i - \bar{x})(y_i - \bar{y}) \quad (3.1)$$

$ss_{xy}$ : x ile y'nin bağıntısı

$x_i$ : x'in i. elemanı

$y_i$ : y'in i. Elemanı

$\bar{x}$ : x'nin ortalaması

$\bar{y}$ : y'nin ortalaması

Bağıntısı bulunmak istenen x ve y dizilerinin her bir elemanı aynı oranda ortalamadan aynı anda yüksek olması veya aynı anda düşük olması ilişkiyi yükseltmektedir. Sensörlerde ölçülen hız değerleri için düşünürsek iki sensörün aynı anda kendi ortalamasından yüksek olması bunlar arasında bağıntıyı arttırırken, biri kendi ortalamasından düşük değerde iken diğer sensörün kendi ortalamasından yüksek olması aralarındaki bağıntıyı düşürmektedir. Aynı anda hız değerleri yükselen ve aynı anda hız değerleri düşen iki sensörün hız bakımından yüksek bağıntıya sahip olduğu söylenebilir. Bu fiziksel anlamda aynı yol üzerinde bulunduğunu anlamına geldiği testlerde görülmüştür. Yüksek bağıntıya sahip sensörlere bakıldığında bunların aynı yol üzerinde bulunan sensörler olduğu görülmektedir.

Matlab'ta işlem sonucunda 4251\*4251 boyutunda sensörlerin bağıntılarını belirten bir matris oluşur. Bu matris kullanılarak ilgili 2 sensöre karşılık gelen bağıntıya bakılarak bu 2 sensörün ne kadar birbirlerini etkiledikleri bulunabilir.

Bağıntıları en yüksek çıkan sensör değerleri tablo 3.3'de gösterilmiştir. Gösterim olarak altçizgiden önceki değer sensör numarasını, sonraki değer ise ne kadar bir zaman sapması ile hız değerini verdiğini göstermektedir.

**Tablo 3.3:** Yüksek bağıntıya sahip sensörler

s95_-5	s73_-0
s95_-40	s73_-35
s95_-45	s73_-40
s95_-10	s73_-5
s95_-35	s73_-30
s95_-15	s73_-10
s95_-20	s73_-15
s95_-30	s73_-25
s95_-25	s73_-20
s95_-0	s73_-0
s95_-50	s73_-45
s95_-40	s73_-40
s266_-5	ts266_-10
s266_-30	ts266_-30
s266_-10	ts266_-15
s266_-15	ts266_-20
s266_-20	ts266_-25
s266_-0	ts266_-5

Tablo 3.3 görülebileceği gibi s95 ile s73 birbirlerine 5 dakika zaman farkı ile bağımlıdır. Bunun anlamı s95'te oluşan bir trafik yoğunluğu veya trafik açılmasının 5 dakika sonra s73'ün hızını da aynı şekilde etkilediğidir. s73'ün şu anki hız değeri s95'in 5 dakika önceki hız değeri ile bağıntısı yüksektir. s95'in 40 dakika önceki hız değeri s73'ün 35 dakika önceki değeriyle bağıntısı yüksektir. Buradan aynı s95 ile s73'ün yol üzerindeki aralarında 5 dakikalık yol mesafesindeki iki sensör olduğu söylenebilir. Aynı şekilde

tablo 3.3'ye bakılarak s266 – ts266'nın ve s156 – ts156'nın yüksek korelasyona sahip sensörler olduğu söylenebilir.

## 4. ÖRÜNTÜ TANIMA YÖNTEMLERİ

### 4.1 Regrasyon Yöntemleri

#### 4.1.1 Karar Destek Yöntemi

##### 4.1.1.1 Karar Destek Makinası (SVM)

SVM **Burges (1998)**, yaklaşık son 10 yıldır kullanılan, değişik problemler için iyi sonuçlar vermiş, bir örüntü tanıma yöntemidir. Vapnik tarafından 1995 yılında geliştirilmiş ve bir çok etkileyici özelliği ve pratikte çok iyi sonuç vermesi sonucu bir çok çalışmada kullanılmıştır. SVM temel olarak yapısal risk küçültme üzerine dayanır. Bu yöntem ANN'de kullanılan deneysel risk küçültme yöntemine göre, yerel sınırlara yakalanmaması ve daha genel değerler vermesi nedeniyle daha başarılıdır. SVM, ilk olarak sınıflandırma problemlerinde kullanılmak üzere kullanılmış ve geliştirilmiştir. Fakat tahmin problemlerinde de başarılı olacak şekilde geliştirilmiş ve bu çalışmada kullanılan SVR (Support Vector regression) yöntemi doğmuştur.

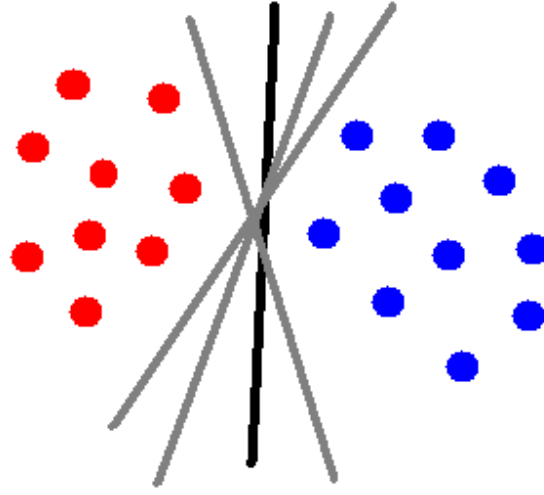
Sınıflandırma problemlerinin en basiti iki sınıfı birbirinden ayırma problemleri olarak düşünülebilir. Sınıflandırma problemlerinde amaç mevcut örnekleri kullanarak iki sınıfı birbirinden ayıracak bir sınıflandırıcı oluşturmaktır. Şekil 4.1'de bir çok doğrusal sınıflandırıcı örnekleri birbirinden ayırabilmektedir. Fakat yalnız bir tanesi en yakın veri noktalarına uzaklığın en fazla olmasını sağlamaktadır. Bu oluşturulan sınır, sınıflandırıcı olarak en iyi sonuç verileceği tahmin edilir ve en iyi ayıran düzlem (optimal separating hyperplane) olarak isimlendirilir.

Eğer problemi eğitim vektörlerini iki ayrı sınıfa ayırmak olarak düşünürsek.

$$(y_1, x_1), \dots, (y_l, x_l), x \in \mathbb{R}^n, y \in \{-1, +1\}$$

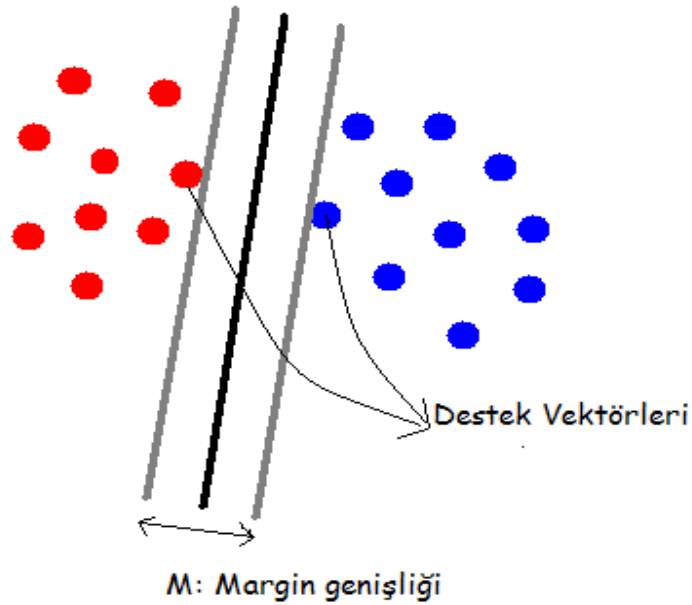
(4.1)'deki gibi formula olan düzlemi bulmak amacımızdır.

$$(w \cdot x) + b = 0 \tag{4.1}$$



**Şekil 4.1:** İki sınıfı ayıran örnek sınıflandırılar

Eğer eğitim vektörleri hatasız şekilde ayrılırsa ve düzleme en yakın veri noktalarının uzaklığı maksimal olursa bu vektörler en iyi şekilde ayrılmıştır. (4.1) SVM'in en basit hali olan doğrusal SVM'i de tanımlamaktadır. Düzleme en yakın noktalar ise şekilde gösterildiği karar destek noktalarıdır. Aradaki uzaklığı maksimize ederek hata yapılması olasılığını en aza indirilmektedir.



**Şekil 4.2:** Karar Destek Makenizmasında Destek Vektörleri

Eğer kırmızı noktalara yakın gri çizgiyi +1 ayırıcı(artı düzlem), mavi noktalara yakın çizgiyi -1 ayırıcı(eksi düzlem) olarak isimlendirsek, kırmızı noktalar için aşağıdaki denklemin sağlanması gerekir. Burada +1 ve -1 uzaklıkları gösterim için sabit uzaklıklar olarak alınmıştır. Tabiki gerçek bir problemde sınırlar arasındaki uzaklık sabit bir sayı olamaz. Bu durumda sınırlar arasındaki uzaklığında bilinmeyen bir katsayı olarak denklemlere girmesi gerekir. Fakat anlatımı basitleştirmek için burada sabit bir katsayı olarak kullanılmıştır.

$$w \cdot x + b \geq 1 \quad (4.2)$$

Aynı şekilde maviler içinde aşağıdaki denklemin sağlanması gerekir.

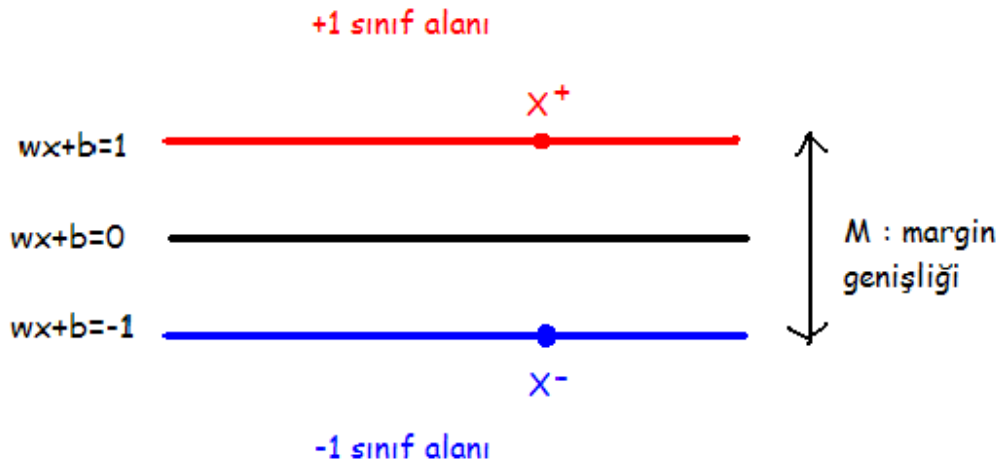
$$w \cdot x + b \leq -1 \quad (4.3)$$

Eğer denklemleri genelleştirirsek (4.4)'deki denkleme ulaşmış oluruz.

$$-1 < w \cdot x + b < 1 \quad (4.4)$$

Ayrıca  $w$  vektörü artı ve eksi düzleme dik olduğu görülebilir. Bu durumda eksi düzlem üzerinde herhangi bir  $x^-$  noktası alırsak ve artı düzlemde ona en yakın olacak şekilde  $x^+$  noktası seçersek aşağıdaki eşitliğin olduğunu varsayabiliriz. Çünkü  $w$  bu düzlemlere dik olduğundan  $w$ 'nin bir katsayı ile çarpılması sonucu  $x^+$  ya ulaşabileceği görülür.

$$x^+ = x^- + \lambda w \quad (4.5)$$



**Şekil 4.3:** Sınıflandırıcı Gösterimi

Eğer  $\lambda$ 'yı  $M$  cinsinden ifade etmek istersek (4.6) eşitliği karşımıza çıkar.

$$w.(x^- + \lambda w) + b = 1$$

$$(w.x^- + b) + \lambda w.w = 1 \quad (4.6)$$

$$-1 + \lambda w.w = 1$$

$$\lambda = \frac{2}{w.w}$$

M'de bu durumda (4.7)'deki gibi ifade edilebilir.

$$M = |x^+ - x^-| = |\lambda w|$$

$$= \lambda |w| = \lambda \sqrt{w.w} \quad (4.7)$$

$$= \frac{2\sqrt{w.w}}{w.w} = \frac{2}{\sqrt{w.w}}$$

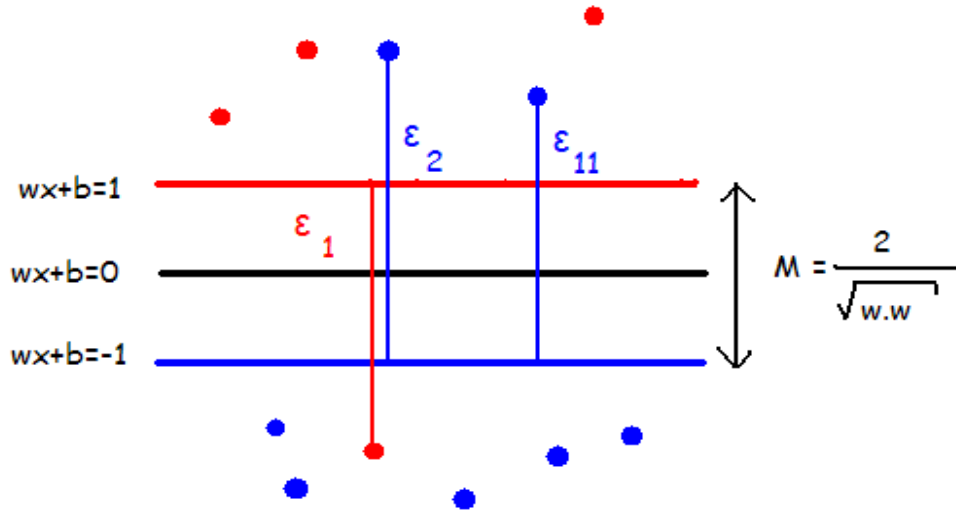
Artık tahmin edilen  $w$  ve  $b$  parametreleri ile her noktanın doğru bölgede olup olmadığı hesaplanabilir ve marginin genişliği bulunabilir. Bize verilen eğitim noktalarından  $w$  ve  $b$  parametrelerini tahmin etmek için belirli bir doğrusal koşul altında karesel fonksiyonları çözen karesel programlama (quadratic programming) kullanılmaktadır. En düşük hale getirmek istediğimiz değer burada sınır uzunluğunun en fazla olması için  $w.w$ 'dur. Aynı zamanda aşağıdaki eşitliği sağlaması gerekmektedir. Bu durumda  $R$  adet örnek noktamız olduğu durumda toplam  $R$  adet koşula sahip olmaktadır.

$$w.x_k + b \geq 1 \text{ eğer } y_k = 1 \quad (4.8)$$

$$w.x_k + b \leq -1 \text{ eğer } y_k = -1$$

(4.8)'deki denklem tamamıyla doğrusal bir şekilde ayrılan düzlemler için doğru olabilmektedir. Fakat pratikte bu her zaman mümkün olmamaktadır, belirli bir hata payını baştan kabul ederek veya ne kadar hata payını kabul edebileceğimizi belirten bir  $C$  parametresiyle hem sınır en fazla olduğu değere çekilmeye çalışılmakta aynı zamanda hata oranı verilen  $C$  hata cezası katkısı göz önüne alınarak azaltılmaya çalışılmaktadır.  $R$  veri noktası için oluşabilecek  $R$  tane  $\epsilon$  hata payı kabul edilmiştir. Daha sonra çalışmada kullanılacağı gibi bu  $C$  parametresi dışarıdan verilerek ayırma işleminin ne kadar hassas olabileceği kararlaştırılabilmektedir. Küçük  $C$  değerleri küçük cezaya denk gelmektedir,

bu durumda hata olma oranı yüksek olabilmektedir, bunun yerine sınırın geniş olması daha büyük bir öncelik teşkil etmektedir. Buna karşın yüksek C değerlerinde sınırlar çok daha katı kurullarla çizilmektedir.



Şekil 4.4: Karar Destek Makinasında Hata Kabulü

Bu durumda minimize edilmeye çalışılan değer (4.9) daki gibidir.

$$\lambda \frac{1}{2} w \cdot w + C \sum_{k=1}^R \epsilon_k \quad (4.9)$$

Koşul değerleride formül (4.10) daki gibi oluşmaktadır.

$$w \cdot x_k + b \geq 1 - \epsilon_k \text{ eğer } y_k = 1$$

$$w \cdot x_k + b \leq -1 + \epsilon_k \text{ eğer } y_k = -1 \quad (4.10)$$

$\epsilon_k \geq 0$  bütün k değerleri için

Bunları karasel programda yerine koyarsak maksimize edeceğimiz değer (4.11) olur.

$$\sum_{k=1}^R a_k - \frac{1}{2} \sum_{k=1}^R \sum_{l=1}^R a_k a_l Q_{kl} \quad (4.11)$$

$$Q_{kl} = y_k y_l (x_k \cdot x_l)$$

Yukarda maksimize edilmesi gereken deęerin sınır kořulları (4.12) de bulunmaktadır.

$$0 \leq a_k \leq C \quad \forall k \quad (4.12)$$

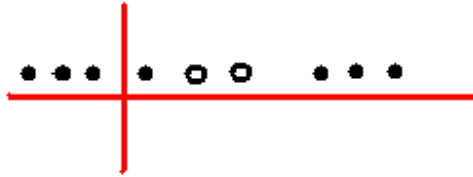
$$\sum_{k=1}^R a_k y_k = 0$$

Bu durumda  $w$  ve  $b$  řu řekilde ifade edilebilir.

$$w = \sum_{k=1}^R a_k y_k x_k \quad (4.13)$$

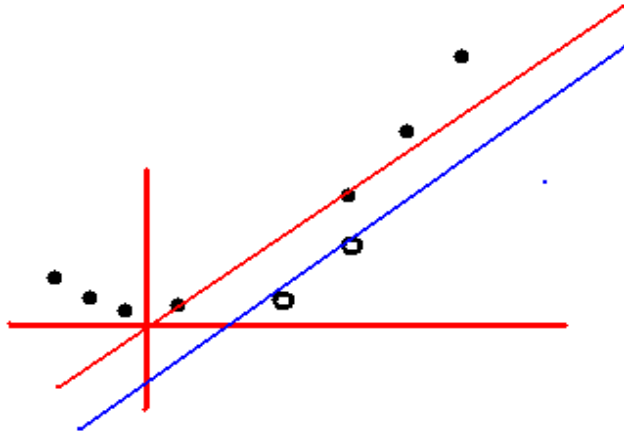
$$b = y_k (1 - \epsilon_k) - x_k \cdot w_k \quad K = \arg_k \max a_k$$

Ne yazık ki her zaman  $n$  adet öznitelik vektörlü bir veri kümesi için  $n$  boyutta doğrusal bir çözüm oluşmamaktadır. Örnek olarak Şekil 4.5'de görüleceęi gibi 2 boyutlu düzlemdeki içi dolu noktalar ile içi boş noktalar için 2 boyutta doğrusal bir çözüm üretilememektedir. Fakat Şekil 4.6'da görüldüęü gibi  $z_k = (x_k, x_k^2)$  işlemini mevcut noktalara uygulandıęında denklem çok boyutlu uzaya taşınmakta ve mevcut iki sınıfı ayırmak için bir düzlem bulunabilmektedir. Bu çok boyutlu uzaya çıkarma işlemi yapan fonksiyonlara kernel fonksiyonu denilmektedir.



Şekil 4.5: Doğrusal Düzlemde Sınıflandırma

Çekirdek fonksiyonlarının uygulanması sonucu karesel denklem aşağıdaki řekilde olmaktadır. Çekirdek fonksiyonları  $\Phi(x)$  ifadesiyle gösterilmektedir.



**Şekil 4.6:** Çok Boyutlu Uzayda Sınıflandırma

$$\sum_{k=1}^R a_k - \frac{1}{2} \sum_{k=1}^R \sum_{l=1}^R a_k a_l Q_{kl} \quad (4.14)$$

$$Q_{kl} = y_k y_l (\Phi(x_k) \cdot \Phi(x_l))$$

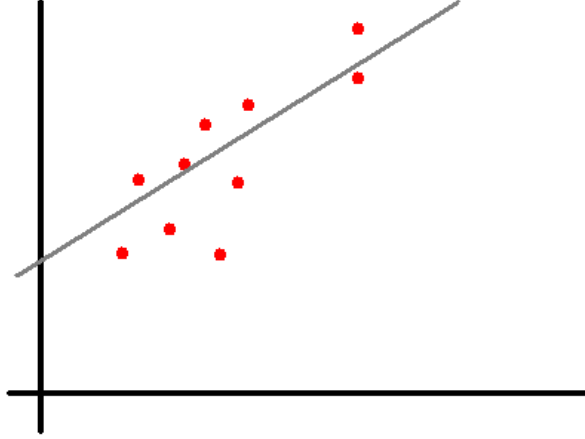
$$w = \sum_{k=1}^R a_k y_k x_k$$

#### 4.1.1.2 Karar Destek Regresyon (SVR)

SVR, SVM'in ilereye dönük tahmin yapmak için kullanılan özel bir versiyonudur. SVR'de SVM'den farklı olarak yeni verilen örneğin bulunduğu sınıf değil, y eksenindeki değeri tahmin edilmek istenmektedir. Aşağıdaki şekilde tek bir sınıfa ait elemanlar bulunmaktadır. Öğrenilmek istenen yeni elemanın y eksenindeki değeridir. SVR burada noktaların en olması gereken konumları bularak hataları minimize eder ve marginini en geniş tutar. Tabiki şekil 4.7'deki noktalar 2 boyutta olduğundan doğrusal bir çizgi ile sınır ifade edilebilmiştir. Daha karmaşık durumlarda çok boyutlu ve daha karmaşık bir sınır düzlemi ortaya çıkabilmektedir.

SVR'in SVM'den temel farkı hata oranını hesaplayan kayıp fonksiyonunun hesaplanmasının farklı olmasıdır. SVR'da minimize edilmek istenen değer formülü (4.15)'deki gibidir.

$$R_{\text{reg}}(f) = C \sum_{i=0}^{\ell} \Gamma(f(x_i) - y_i) + \frac{1}{2} \|w\|^2 \quad (4.15)$$



**Şekil 4.7:** SVR ile regrasyon

SVR'da kayıp fonksiyonun en çok kullanılan formülü (4.16)'daki gibidir.

$$\Gamma(f(x) - y) = \begin{cases} |f(x) - y| - \varepsilon, & \text{eger } |f(x) - y| \geq \varepsilon \\ 0 & \text{di sin da} \end{cases} \quad (4.16)$$

**Tablo 4.1:** Çekirdek Fonksiyonları

Çekirdek	Fonksiyon
Doğrusal	$x \cdot y$
Polinom	$[(x * x_i) + 1]^d$
RBF(Radial Basis Function)	$\exp\{-\gamma x - x_i ^2\}$
Neural	$\tanh(ax * y + b)$
anova	$(\sum_i \exp\{-\gamma x - x_i ^2\})^d$

SVM ve SVR'da en çok kullanılan çekirdek fonksiyonları tabloda bulunanlardır.

Çalışmada en iyi sonucu veren doğrusal çekirdek kullanılmıştır.

#### 4.1.3 k-Nearest Neighbor(KNN)

kNN **Alpaydın (2004)**, yöntemi en basit örüntü tanıma yöntemlerinden birisidir. Sınıflandırma problemlerinde kullanılan KNN, bu çalışmada regresyon için kullanılmıştır. KNN kendisine verilen bir test noktasının sınıfını bulmaya çalışırken kendisine daha önce verilen eğitim noktalarına bakar ve bunlardan en yakın özellikli k taneyi seçer. En yakın k tane eğitim noktasının mevcut bulunduğu sınıflara bakar ve bunların en fazla bulunduğu sınıfı örnek test noktasını uygun görür. Örnek olarak Şekil 4.8'de k=5 olarak ayarlanmıştır. Sınıfı anlaşılmaya çalışılan X noktasından başlayarak 5 adet eğitim noktasına rastlayıncaya kadar eğitim noktalarını kapsayan daire büyütülür. 5 adet nokta bulunduğu anda araştırma sonlandırılır ve en çok hangi sınıfa ait eğitim noktalarının bulunduğu bakılır. Şekil 4.8'de yuvarlak kırmızılarının sayısı daha fazla olduğundan X örnek noktamız yuvarlak kırmızı olarak değerlendirilecektir. Daire içinde kalan k nokta için yalnız sayısal üstünlüğün önemli olmadığı X noktasına uzaklığında değerlendirildiği bir hesaplamada yapılabilir. Bu durumda her noktaya uzaklığının tersi oranında bir katsayı verilebilir. Çalışmamızda olduğu gibi regresyon işleminin yapılacağı durumda sınıfa değil kendisine en yakın k tane noktanın y koordinatındaki değerlerin ortalaması alınır.

Çalışmada öznitelik vektörü sayısı birden fazla olduğundan noktaların birbirinden uzaklığını ölçmek için aşağıdaki formül kullanılmıştır.

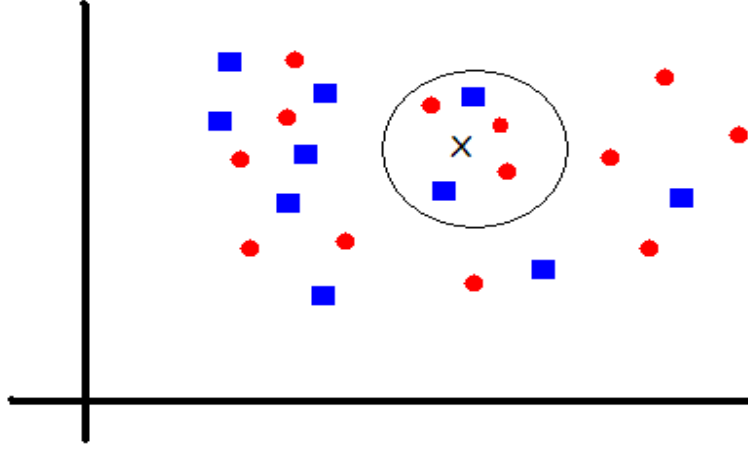
$$D_r(a,b) = \left( \sum_{k=0}^r (a_k - b_k)^2 \right)^{1/2} \quad (4.17)$$

$D_r(a,b)$ : a ve b noktaları arasındaki uzaklık

$a_k$ : a'nın k. Özelliği

$b_k$ : b'nin k. Özelliği

K adet en yakın hız noktası belirlendikten sonra bu hızların aritmetik ortalaması alınarak tamini hız elde edilir. Çalışmada k değeri için en uygun değer yapılan testler sonucu 25 olarak bulunmuştur.



**Şekil 4.8:** KNN metodu gösterimi

#### 4.2 Öznitelik Seçme Yöntemleri

Öznitelik seçme yöntemleri (feature selection), veri boyutu azaltma (dimensionality reduction) yöntemlerinin bir alt grubudur. PCA, ICA gibi öznitelik projeksiyon yöntemleri, bütün ölçülen özniteliklerin doğrusal birleşimi olan ama daha az sayıda boyut kullanırken, ileri ya da geriye doğru öznitelik seçimi (forward/backward feature selection) sadece belirli öznitelikleri seçip kullanmaktadır. Özellikle ölçümlerin alınmasının ya da saklanmasının zaman ya da para açısından masraflı olduğu durumlarda öznitelik seçme yöntemleri, öznitelik projeksiyon yöntemlerine tercih edilir.

Örnek problemimizde öznitelik seçme yöntemlerinin kullanılması zorunlu olmuştur çünkü öznitelik sayısı arttıkça hesaplama süresi karesi oranında artmaktadır. Ayrıca sonuçlar göstermektedir ki ortalama 7-8 özniteliğin kullanılması en iyi sonucu vermektedir. En iyi özniteliklerin bulunması için ileriye yönelik özellik seçim yöntemi kullanılmıştır. Başlangıçta n adet öznitelik vektörü ile başlanmış ve her döngü sonucu en kötü özellik gösteren öznitelik atılarak algoritmaların performansı ölçülmüştür.

#### 4.3 Örüntü Tanıma Algoritmalarının Performanslarının Ölçümü

Öngörülmesi istenen zamanları, gerçek hız değerini, ise öngörülen hızı gösterebilir. Performans ölçütü olarak gerçek ve öngörülen hızların farklarının mutlak değerinin, gerçek hıza oranının ortalaması alınmıştır. Formül (4.18)'de bu gösterilmiştir.

$$\text{RME} = \frac{1}{n} \sum_{i=1}^n \left| \frac{V(t)_i - V(t)_i^*}{V(t)_i} \right| \quad (4.18)$$

## 4.4 mySVM Programı

### 4.4.1 Programın Kullanılması

mySVM Stefan Rüping tarafından örüntü tanıma ve regresyon problemlerinde kullanılmak için C++ ile yazılmış açık kaynak kodlu bir yazılımdır (**Rüping, 2004**). Çalışmada SVR tahminleri yapmak için bu program kullanılmıştır. mySVM kendisine parametre olarak 3 adet dosya almaktadır. Bu dosyaların biri SVR parametrelerini içermekte, diğeri eğitim kümesini içermekte, üçüncüsü ise test kümesini içermektedir.

3 çeşit giriş tipi alarak kendisine verilen dosyanın ne çeşit bilgiler içerdiği mySVM tarafından anlaşılır. Bu giriş tipleri şunlardır.

@parameter: mySVM'in kullanılacağı parametrelerin giriş yapılacağını belirtir.

@kernel : kernel tipinin giriş yapılacağını belirtilir.

@example: eğitim veya test kümesinin giriş yapılacağını belirtilir.

Parametreleri içeren dosyada şu özellikleri belirterek SVR'in çalışma şekli değiştirilebilmektedir.

**Pattern:** Eğer örüntü tanıma özellikleri kullanılacaksa oluşabilecek sınıfların yapısı belirtilir.

**Regression:** SVR'in kullanılacağı belirtilir. Varsayılan olarak bu seçilidir.

**Format:** Alınacak eğitim ve test verilerinin biçimi bu parametre ile belirtilir. Eğer belirtilmezse varsayılan biçimde kabul eder.

**C:** Hata ceza oranını belirtmektedir. Yüksek hata ceza oranları olabildiğince doğru bir sınır eğrisi araştırılmasına neden olmaktadır. Bu durumda tekrar eden döngü sayısı çok fazla olmaktadır. Hatta 1 000 000 döngüde bile kesin sonucun bulunamadığı durumlar oluşabilmektedir, bu aynı zamanda programın çalışma zamanını oldukça arttırmaktadır. Hız öngürüsü sistemleri gibi gerçek zamanda işlem yapması gereken sistemlerde C düşük tutularak çalışma zamanı azaltılabilir. Böylece daha fazla hataya olasılık versede

çalışma zamanından kar edilecektir. Çalışmada yapılan testler sonucunda C'nin 40 verilmesinin en iyi sonucu verdiği görülmüştür.

**Epsilon:** Duyarsızlık katsayısı olarak değerlendirilir. Hata değeri bu değerden düşük olan noktalar cezalandırılmazlar. Çalışmada bu katsayı 0.1 olarak verilmiştir.

**Working\_set\_size:** Eğitim ve test veri kümesindeki eleman sayısını belirtmektedir. Eğer verilmezse bu parametreler ilgili dosyalardan alınır.

**Max\_iterations:** En fazla kaç döngüde sonuca ulaşılmazsa programın durdurulacağını belirtir. Elde edilen en iyi sonuç geri döndürülür. Varsayılan değeri 100 000'dir. Fakat bu rakam çalışma zamanını oldukça arttırmaktadır bu yüzden çalışmada bu değer 10 000 olarak verilmiştir. Yapılan testler sonucu döngü sayısının düşürülmesinin %0.1 den fazla hata oranını arttırmadığı görülmüştür.

Programda kullanılacak çekirdek ise @kernel satırından sonra seçilebilmektedir. Bu kernel SVR'ı çok boyutlu hale getirecek fonksiyonu belirtmektedir ve daha önce belirtilen çekirdek fonksiyonları bölümüne karşılık gelmektedir. Kullanılabilecek çekirdekler ise şunlardır.

**Dot:** Parametre almamaktadır. Varsayılan olarak bu çekirdek kullanılmaktadır.

**Polynomial:** Polinom çekirdeğinin kullanılmasını sağlar. Parametre olarak polinomun derecesini almaktadır.

**Radial:** Radial çekirdeğinin kullanılmasını sağlar. Parametre olarak gamanın değerini alır.

**Neural:** Neural çekirdeğinin kullanılmasını sağlar. Parametre olarak a ve b değerini alır.

**Anova:** Anova çekirdeğinin kullanılmasını sağlar. Parametre olarak gamma ve polinomun derece değerini alır.

**User kernel:** Kullanıcı derlenmiş kendi yazdığı kernelide \*.cpp dosyasını parametre vererek kullanabilir.

Çalışmada kullanılan kernel dosyasının içeriği aşağıdaki gibidir.

```
@kernel
```

```
type dot
```

```
@parameters
C 40
epsilon 0.1
max_iterations 10000
```

Eğitim veri dosyasının yapısının içeriği ise şu şekildedir.

```
@examples
format xy
dim 2
1 1 2
-1 -1 4
2 0 0
```

Burada kullanılan özellikler şunlardır.

**format :** Kendisine verilen biçimin ne şekilde olacağı belirtilir. Örnek gösterimde xy biçiminde veri verileceği belirtilmiştir. Bu durumda kendisine dim komutu ile verilen öznitelik sayısı kadar x özelliği verilir. Son eleman ise bu x elemanlarına karşılık gelen çıktıdır. mySVM kendisine verilen her bir satırı alarak bir örnek nokta olarak değerlendirir ve kendisini x değerlerine karşılık gelecek y değeri verecek şekilde eğitir.

dim: Eğitim kümesindeki elemanların kaç boyutlu olduğunu belirtir. Bu aynı zamanda öznitelik sayısına karşılık gelir.

Çalışmada kullanılan örnek bir eğitim veri kümesi dosyası aşağıdaki şekildedir. 14 boyutlu bir öznitelik vektörüne karşılık gelmektedir.

```
@examples
dim 14
format xy
120 114 113 114 113 111 110 113 112 113 112 113 116 113 112
116 118 112 113 114 113 111 110 113 112 113 112 113 116 113
110 115 109 112 113 114 113 111 110 113 112 113 112 113 116
112 115 107 109 112 113 114 113 111 110 113 112 113 112 113
112 117 107 107 109 112 113 114 113 111 110 113 112 113 112
```

110 117 109 107 107 109 112 113 114 113 111 110 113 112 113  
110 110 113 109 107 107 109 112 113 114 113 111 110 113 112

**Test veri kümesi :** Eğitim veri kümesi ile eğitilmiş mySVM, sınır eğrisini kullanarak test dosyasındaki noktaların y eksenine karşılık gelen çıktısını üretmektedir. Test kümesinin y elemanları tahmin edilmeye çalışıldığından bu değerler verilmez. Format parametresi de bunun için sadece x şeklinde belirtilmiştir. Format parametresinde y olup olmaması aynı zamanda verilen dosyanın eğitim mi, test veri kümesi mi olduğunun anlaşılmasında kullanılmaktadır.

@examples

format x

dim 2

1 1

-1 -1

2 0

Çalışmada kullanılan örnek bir test dosyasının ise şu şekildedir.

@examples

dim 14

format x

102 112 105 113 111 110 110 110 109 109 112 102 103 107

104 111 108 105 113 111 110 110 110 109 109 112 102 103

109 110 114 108 105 113 111 110 110 110 109 109 112 102

107 112 114 114 108 105 113 111 110 110 110 109 109 112

110 117 112 114 114 108 105 113 111 110 110 110 109 109

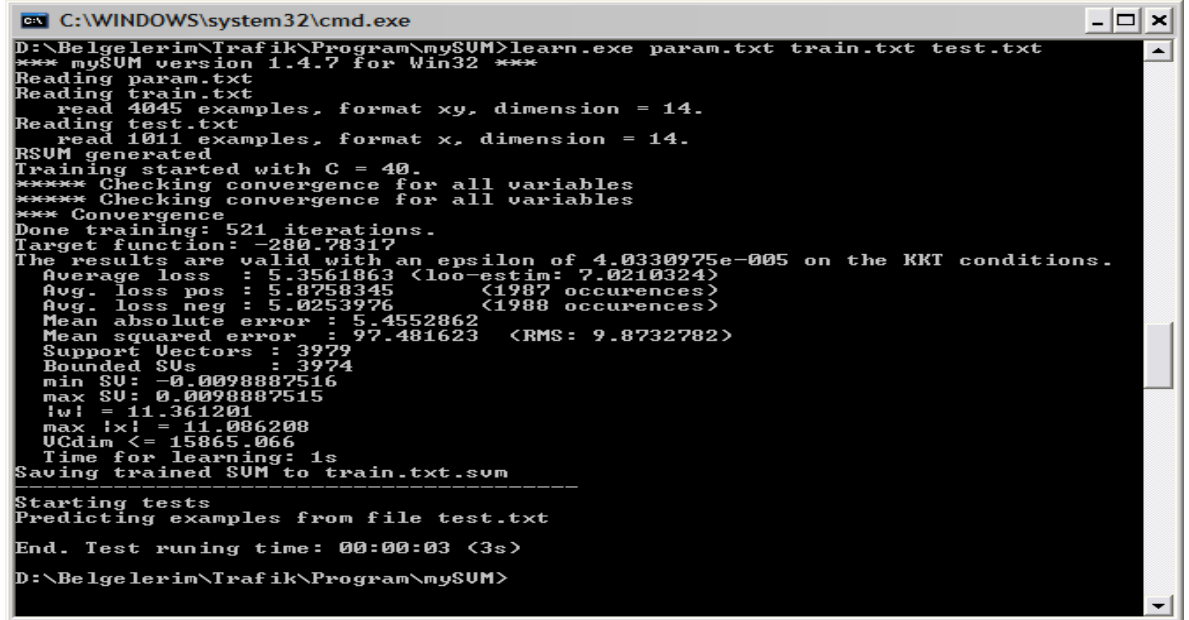
110 118 115 112 114 114 108 105 113 111 110 110 110 109

#### 4.4.2 Programın Çalıştırılması

Programın çalıştırılması için learn.exe'ye yukarıda belirtilen 3 adet kaynak dosyasının verilmesi gerekmektedir. Örnek bir çalıştırma komutu aşağıdaki gibidir.

learn.exe param.txt train.txt test.txt

Programın çalıştırılması sonucu oluşan ekran görüntüsü aşağıdaki gibi olmaktadır. Görüldüğü gibi eğitim sürecinin kaç iterasyonda yapıldığı, destek vektörlerinin sayısı gibi eğitim süreci ile ilgili bilgiler ekranda verilmektedir.



Şekil 4.9: mySVM Programının Çıktısı

Programın çalışması sonucu testdosyasismi.pred uzantılı bir dosya oluşur. Bu dosyada verilen test noktaları için tahmin edilen y değerleri bulunmaktadır. Örnek dosyamız için üretilen test dosyasının içeriği aşağıdaki gibidir. Her satırdaki son değer tahmin kendisinden önce verilen x noktası için tahmin edilen y değerini vermektedir.

@examples

# svm example set

dimension 14

number 1011

b 98.3951

format xy

102 112 105 113 111 110 110 110 109 109 112 102 103 107 106.604

104 111 108 105 113 111 110 110 110 109 109 112 102 103 106.491

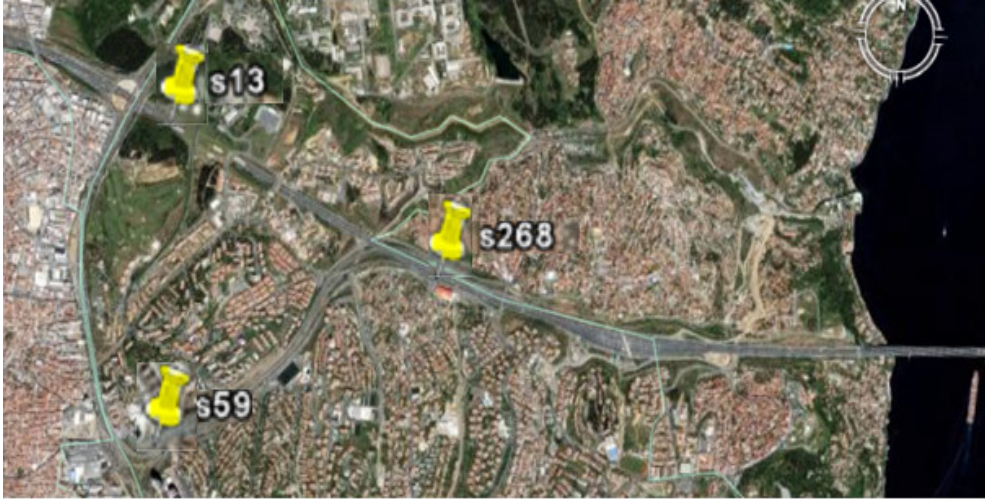
109 110 114 108 105 113 111 110 110 110 109 109 112 102 107.846

107 112 114 114 108 105 113 111 110 110 110 109 109 112 110.241

## 5 GELECEĞE YÖNELİK HIZ TAHMİNİ

### 5.1 Sadece Sensördeki Verileri Kullanarak

Bu bölümde sensörlerden şu an ve geçmişte ölçülen hız değerleriyle, ilgili sensörün bulunduğu konumda gelecekte oluşabilecek hızların öngörüsü yapılacaktır. Veri olarak veri kümesi bölümünde anlatıldığı gibi İstanbul'da bulunan 327 RTMS'ten alınan veriler kullanılacak ve örüntü tanıma yöntemleri kısmında metotları anlatılan SVM ve KNN yöntemleri kullanılarak tek bir sensörün verisi işlenerek ilgili sensörün kısa zaman sonraki hız değeri tahmin edilmeye çalışılacaktır.



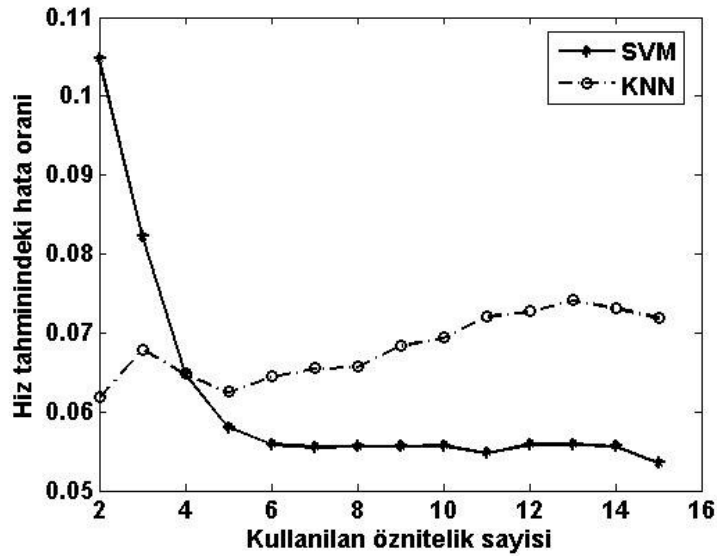
**Şekil 5.1:** s13, s59 ve s268 sensörlerinin konumları

Örnek olarak kullanılan sensörler **Yıldırım ve Çataltepe (2008)** çalışmamızda kullanılan 3 sensör noktasıdır. Bu 3 sensör Fatih Sultan Memhmet (FSM) köprüsüne Avrupa yakasında bağlantı yolları üzerinde bulunmaktadır. Bu 3 noktanın diğer bir özelliği ise trafik sıkışıklığının özellikle işe gidiş saatleri olan sabah 7-9 arası ve işten geliş saatleri olan 18-20 saatleri arasında trafik yoğunluğunun fazla olmasıdır.

SVM ve KNN yöntemlerinin başarımları ölçülürken başlangıçta öznitelik sayısı 15 olarak tutulmuştur. Bu öznitelikler veri kümesi bölümünde açıklanan s vektöründe

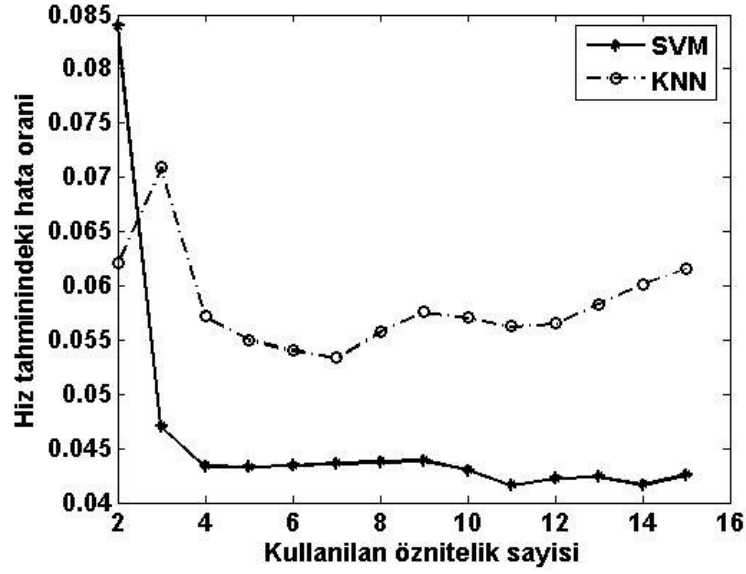
belirtilmek üzere [0,-5,-10,-15,-20,-25,-30,-35,-40,-45,-50,-55,-60,-1440,-10080] şeklindedir. Yani şu anki andan itibaren 5'şer dakika aralıklarla 60 dakika öncesine kadar hızları vermektedir. Ayrıca ölçülmek istenen andan tam 1 gün önceki ve 1 hafta önceki hızlarda öznitelik olarak vermektedir. Bu öznitelikler verilerek 15.12.2007 ile 15.01.2008 tarihleri arasında ilgili sensör için 5 dakika aralıklarla hızlar alınmaktadır. Bu alınan veri kümesi %80 eğitim ve %20 test verisi olarak ayrılarak aşağıdaki testler yapılmaktadır. Öznitelik sayısı başlangıçta 15 olarak verildikten sonra öznitelik seçme yöntemlerinde anlatılan geriye doğru seçme yöntemi kullanılarak her seferinde en kötü özelliği gösteren bir öznitelik atılması sonucu öznitelikler giderek azaltılmış ve örüntü tanıma yöntemlerinin başarımına bakılmıştır.

s268, s13, s59 sensörlerinin ile 5 dakika sonrası için yapılan hız öngörüsünün sonuçları şekil 5.2, şekil 5.3 ve şekil5.4'de gösterilmiştir.

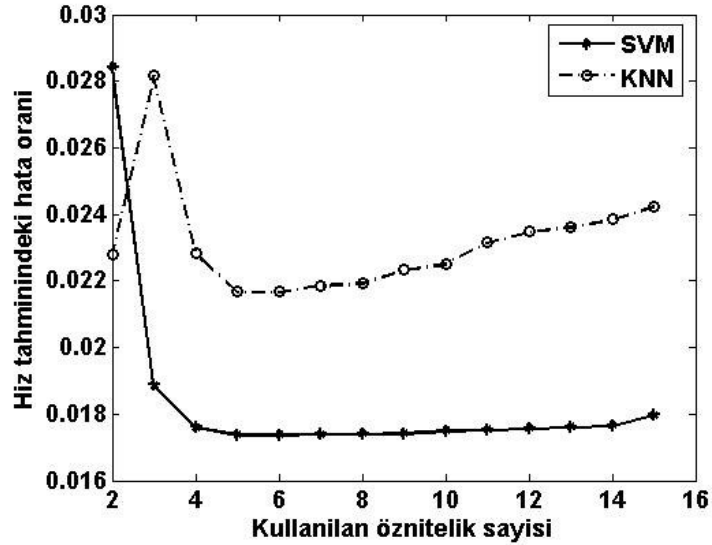


Şekil 5.2: s268 sensörü için 5 dakika sonraki hız öngörüsü

Bu 3 sensör için 5 dakika sonrası hız öngörüsü testleri sonucu SVM yönteminin genel olarak KNN'den daha iyi sonuç verdiği görülmüştür. Fakat alınan hata oranları her 3 sensör için farklı olması yolun özelliklerinin yöntemlerin verdiği hata oranlarını değiştirdiğini de göstermiştir.



Şekil 5.3: s13 sensörü için 5 dakika sonraki hız öngörüsü



Şekil 5.4: s59 sensörü için 5 dakika sonraki hız öngörüsü

SVM için öznitelik sayısı en fazla iken hız hata tahmin oranı en düşük olduğu gözlenmektedir. Fakat öznitelik sayısı 6 olana kadar nerdeyse hata oranı sabit olarak gitmektedir. Bu SVM için ilgili sensör için en iyi 6 özneliğin bulunup sadece bu 6 özneliğin hız öngörüsünde kullanılmasının daha uygun olacağını göstermektedir.

Çünkü öznelik sayısı arttıkça hesaplanma süresi artmaktadır. Öznelik sayısı 6'dan daha az olduğu durumda ise hata oranı giderek artmaktadır.

KNN için ise öznelik sayısı azaldıkça başarımın arttığı görülmektedir. Öznelik sayısı azaldıkça başarımı SVM'e yaklaşmaktadır. Ayrıca s268 ve s13'e göre nispeten daha akıcı seyir eden s59 yolunda hata oranının daha az olduğu da gözlenmektedir.

Tablo 5.1 ve Tablo 5.2'de SVM ve KNN için yapılan testler sonucu öznelikler önem sırasına göre sıralanmıştır.

Tablo 5.1 ve Tablo 5.2'de SVM ve KNN için ortak olarak görülen 1 gün önceki ve 1 hafta önceki hız değerlerinin tahminde oldukça önem derecesinin yüksek olduğudur.

**Tablo 5.1:** Azalan önem sırası ile, üç değişik noktada SVM kullanılarak seçilen öznelikler

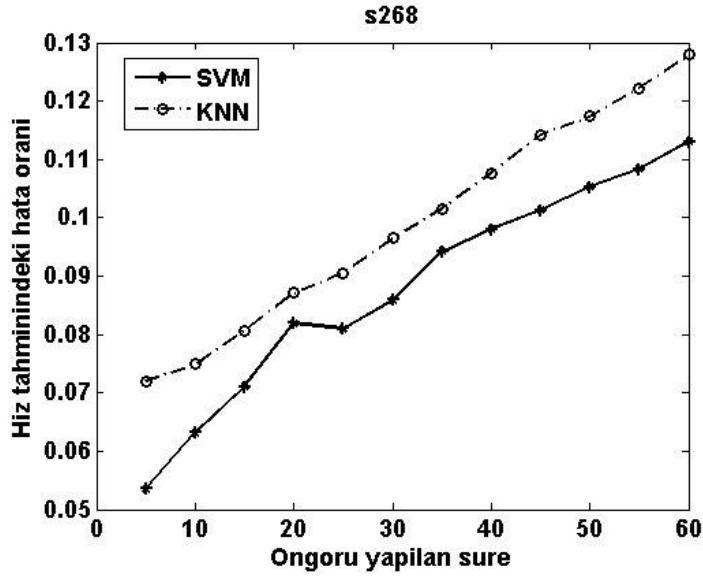
Öznelik	S13	S59	S268
1	<b>-1440</b>	<b>-10080</b>	<b>-5</b>
2	<b>-10080</b>	<b>-1440</b>	<b>-1440</b>
3	<b>-30</b>	<b>-5</b>	<b>-10080</b>
4	<b>-10</b>	<b>-60</b>	<b>-10</b>
5	<b>-15</b>	<b>-35</b>	<b>-50</b>
6	<b>-40</b>	<b>-30</b>	<b>-35</b>
7	-20	-20	-60
8	-45	-50	-25
9	-25	-15	-30
10	-50	-45	-45
11	-55	-55	-40
12	-35	-40	-55
13	-60	-25	-15
14	-5	-10	-20

**Tablo 5.2:** Azalan önem sırası ile, üç değişik noktada KNN kullanılarak seçilen öznelikler

Öznelik	S13	S59	S268
1	<b>-5</b>	<b>-5</b>	<b>-5</b>
2	<b>-10</b>	<b>-10080</b>	<b>-10</b>
3	<b>-10080</b>	<b>-1440</b>	<b>-10080</b>
4	<b>-15</b>	<b>-10</b>	<b>-1440</b>
5	<b>-20</b>	<b>-25</b>	<b>-15</b>
6	<b>-30</b>	<b>-40</b>	<b>-55</b>
7	-55	-15	-20
8	-50	-45	-60
9	-1440	-20	-30
10	-25	-55	-40
11	-35	-35	-50
12	-40	-30	-45
13	-45	-60	-25
14	-60	-50	-35

Ayrıca şu anki ana yakın hızların kullanılması da başarıyı arttırmaktadır.

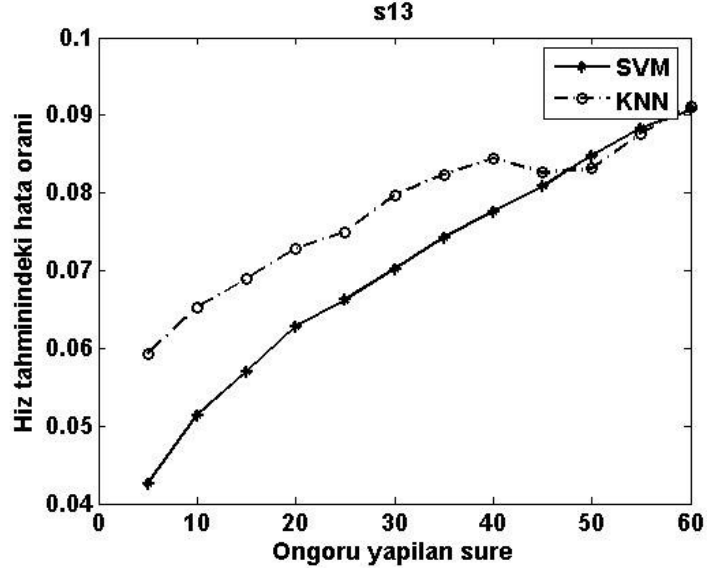
5 dakikadan 60 dakikaya kadar olan sürelerde hız tahmini yapılmak istendiğinde şekil 5.5, Şekil 5.6 ve Şekil 5.7’de gösterilen sonuçlar elde edilmiştir. SVM için en iyi özellik gösteren öznitelik vektörü olan [0,-5,-10,-15,-20,-25,-30,-35,-40,-45,-50,-55,-60,-1440,-10080] alınmıştır. Bu vektör 5 dakika sonraki hızı tahmin için kullanılan öznitelik vektörüdür. 10 dakika sonrayı tahmin etmek için kullanılan vektör ise [0,-10,-15,-20,-25,-30,-35,-40,-45,-50,-55,-60,-65,-1440,-10080] şeklindedir. Burada 5 dakika sonraki hız öngörüsünde kullanılan vektör ile farkı -5 dakika önceki değeri bilemediğimizden bunu kullanmamız ve bunun yerine -65 dakika önceki hız değerini kullanmamızdır. Yine ölçülmek istenen zamandan tam 1 gün ve 1 hafta önceki hız değerleri kullanılmaktadır. KNN için yine onun en iyi sonuç gösterdiği [0,-5,-10,-15,-20,-25,-30,-1440,-10080] vektörü başlangıç olarak kullanılacaktır.



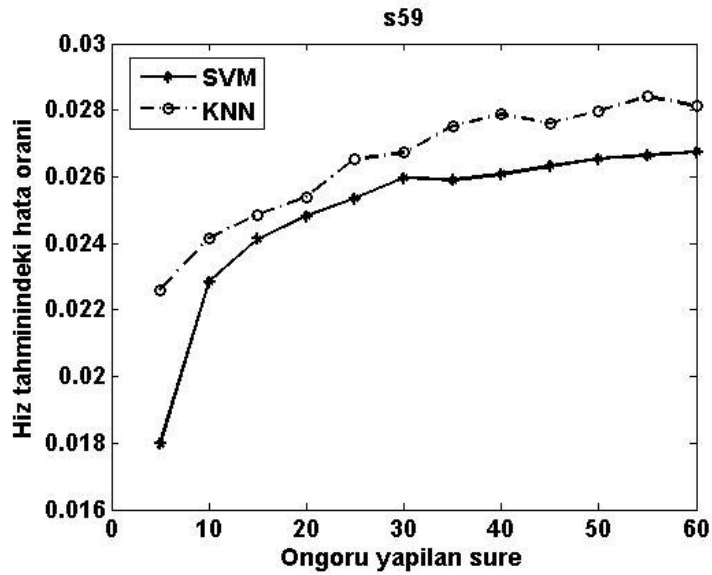
Şekil 5.5: s268 sensörü için artan sürelerde hız öngörüsü

Artan sürelerde yapılan hız öngörüsü ölçümlerinde yine SVM’in KNN’den daha iyi performans gösterdiği görülmüştür. Fakat öngörü yapılan süre arttıkça SVM ile KNN arasında hata oranı farkı azalmakta ve KNN performansı SVM’e yaklaşmaktadır. Bu **Kwon ve diğ. (2000), Hobeika ve diğ., (1994)** çalışmalarında belirtilen öngörü süresi arttıkça tarihsel ortalamanın başarımının arttığı çıkarımıyla örtüşmektedir. Çünkü KNN yöntemi de bir nevi tarihsel ortalama işlevi görmektedir. Benzer şekilde tahmin süresi

uzadıkça tahmin işleminin giderek daha rastgele bir işlem haline gelmesi kullanılan yöntemlerin başarımlarını yaklaştırmaktadır.



Şekil 5.6: s13 sensörü için artan sürelerde hız öngörüsü

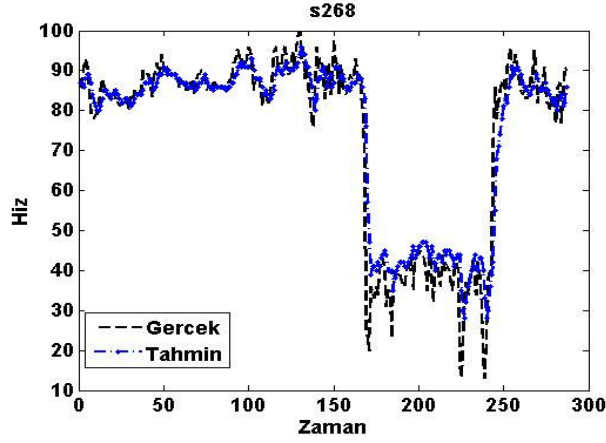


Şekil 5.7: s59 sensörü için artan sürelerde hız öngörüsü

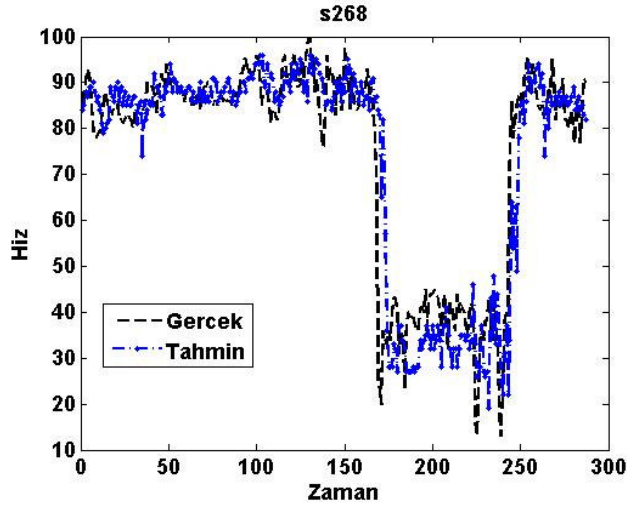
Ayrıca beklendiği gibi öngörü yapılan süre arttıkça oluşan hata oranı artmaktadır. Fakat dikkat edileceği gibi 5 dakikalık öngörü süresinde diğerlerine göre daha az hatası olan s59 sensörünün hata oranı 60 dakikalık hız öngörüsünde hata oranı yine diğerlerine göre

azdır. Yani kısa sürelerde az hatası olan sensörün uzun öngörü sürelerinde de hatası az olmaktadır.

24/12/2007 Pazartesi gününde 5 dakika sonrası hız öngörüsü ve ölçülen zaman karşılaştırmalı grafiği Şekil 5.8 ve Şekil 5.9’da gösterilmektedir.



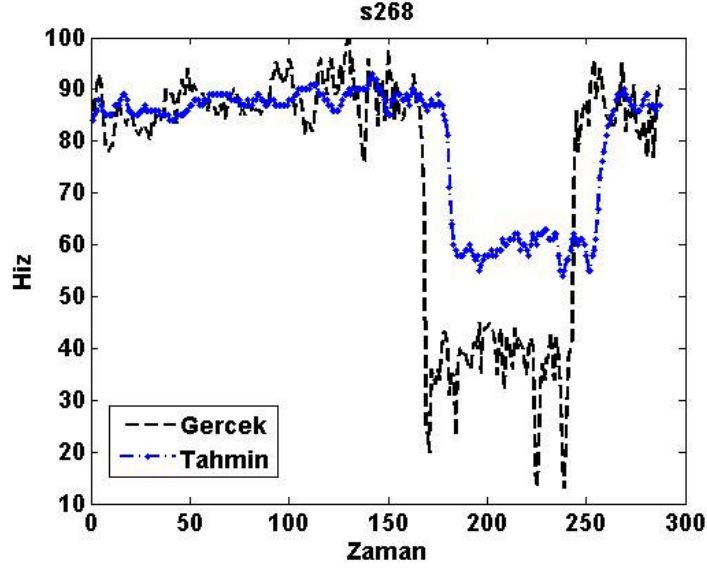
**Şekil 5.8:** s268 için 24/12/2007 tarihinde 00:00 ile 23:55 saatleri arasında SVM metodu için 5 dakikalık hız öngörüsü-gerçek hız



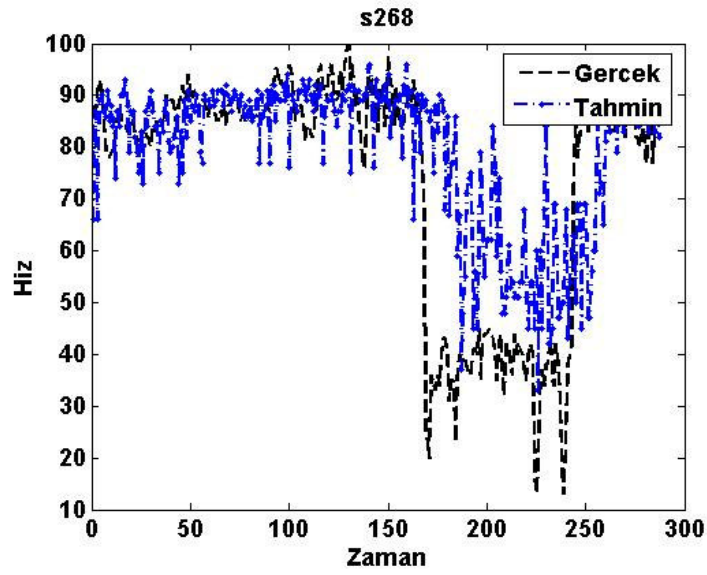
**Şekil 5.9:** s268 için 24/12/2007 tarihinde 00:00 ile 23:55 saatleri arasında KNN için 5 dakikalık hız öngörüsü-gerçek hız

Şekil 5.8 ve Şekil 5.9’da görüldüğü gibi SVM, KNN’ye göre gerçek hızı daha yakın tahmin etmektedir. Hız değişimlerinin birden olduğu durumlarda ise öngörü sapması daha fazla olmaktadır.

24/12/2007 Pazartesi gününde 60 dakika sonrası hız öngörüsü ve ölçülen zaman karşılaştırmalı grafiği ise aşağıdaki şekillerde gösterilmektedir.



**Şekil 5.10:** s268 için 24/12/2007 tarihinde tarihinde 00:00 ile 23:55 saatleri arasında SVM metodu için 60 dakikalık hız öngörüsü-gerçek hız



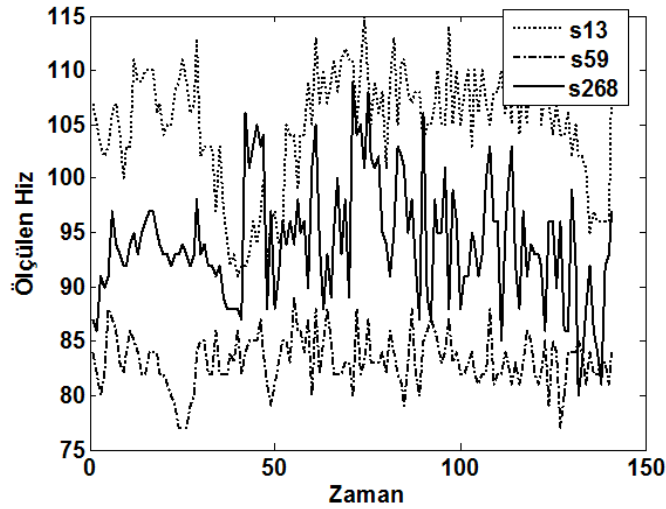
**Şekil 5.11:** s268 için 24/12/2007 tarihinde tarihinde 00:00 ile 23:55 saatleri arasında KNN metodu için 60 dakikalık hız öngörüsü-gerçek hız

5 dakikalık öngörü grafiğinde olduğu gibi SVM, KNN'ye göre daha iyi tahmin yaptığı gözükmemektedir. Fakat hızın normalden düşük olduğu bölümlerde öngörünün tahmin

oranı düşmektedir. Çünkü eğitim sırasında hızın düşük olduğu bölümden 1 saat sonra hızın aynı oranda arttığı veya düştüğü örnekler bulunmaktadır.

## 5.2 Başka Sensörlerdeki Verilerin Kullanılması

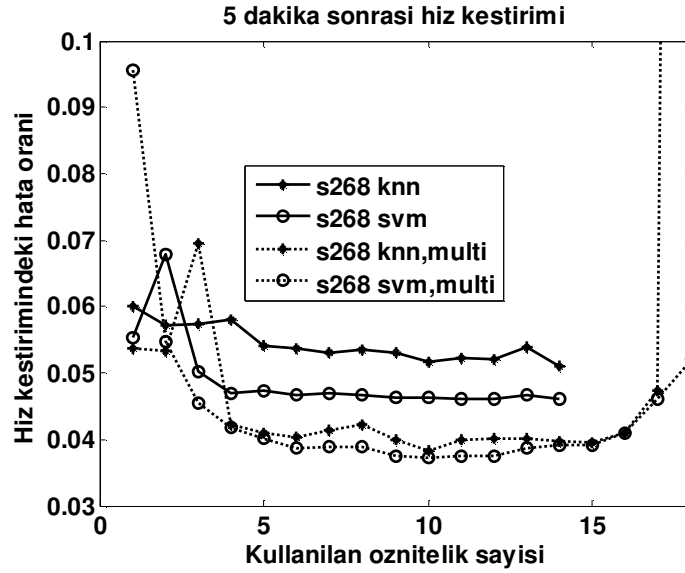
Birbirlerine yakın konumlarda bulunan sensörlerin benzer hız değerleri ürettikleri görülmektedir. Örnek olarak haritada konumları görünen s268, s59, s13 sensörlerinin 3/12/2007 tarihinde alınan hız-zaman değerleri Şekil 5.12’de gösterilmiştir.



**Şekil 5.12:** s13, s59, s268 noktalarından 3/12/2007 Pazartesi günü 00:00 ile 23:55 saatleri arasında alınan hız ölçümleri

Şekil 5.12’de görüldüğü gibi bu 3 sensör konumunun birbirlerinden oldukça fazla oranda etkiledikleri gözükmemektedir. Özellikle hızlarındaki artış ve azalış oranları birbirlerine paralellik göstermektedir. Özellikle bunların birbirine bağlı yollarda olması birindeki trafik artışının diğer yollarında belirli bir süre sonra etkileyeceği sonucunu çıkartmaktadır. Şekil 5.13 ve Şekil 5.14 deki gösterimlerde s268 sensörü yalnız kendisinin değil s13 ve s59’un verileri kullanılarak hesaplanmış ve sadece s268 ile hesaplanan hata oranları ile karşılaştırılmıştır. s13 ve s59’un ölçüm yaptığı yolların birleşerek s268’i oluşturması sonucu bu 2 sensörün kullanılması uygun görülmüştür. Kullanılan öznitelikler içinde artık s13 ve s59’unda bilgileri olacaktır. Kullanılan s vektörü veri kümesinde bölümünde birden fazla sensörün kullanılması durumunda kullanılan biçimde olacak şekilde belirtildiği gibi ([s268:0], [s268:-5], [s268:-10], [s268:-15], [s268:-20], [s268:-25],

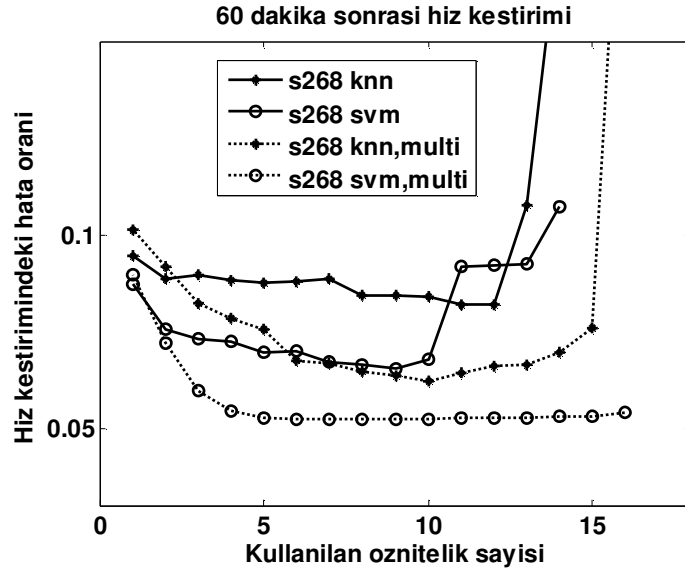
[s268:-30], [s268:-35], [s268:-40] , [s268:-45], [s268:-50], [s268:-55], [s268:-60] , [s268:-1440], [s268:-10080], [s59:-5], [s59:-10], [s59:-15], [s59:-20], [s59:-25], [s59:-30], [s59:-35], [s59:-40] , [s59:-45], [s59:-50], [s59:-55], [s59:-60] , [s59:-1440], [s59:-10080] ], [s13:-5], [s13:-10], [s13:-15], [s13:-20], [s268:-25], [s268:-30], [s13:-35], [s13:-40] , [s13:-45], [s13:-50], [s13:-55], [s13:-60] , [s13:-1440], [s13:-10080]) şeklinde tanımlanmıştır. Bu tanımlamada s13 ve s59'un tahmin edilmek istenen 5 dakika sonraki değerlerini bilmediğimiz için s59:0 ve s13:0 değerleri kullanılmamıştır. Şekil 5.13 ve Şekil5.14'deki gösterimlerde birden fazla sensörün kullanıldığı durumlarda 15 ve altı özneliğin kullanıldığı sonuçlar verilmiştir.



**Şekil 5.13:** s268 noktasında 5 dakika sonrasi hız tahmini için geriye doğru öznelik seçimi

5 dakika sonrasi hız tahminlerinde birden fazla sensör kullanıldığı durumda genel olarak tek sensör kullanıldığı duruma göre daha iyi sonuç verdiği görülmektedir.

60 dakika sonrasi hız tahmininde birden fazla sensör kullanıldığında S vektörü ([s268:0], [s268:-60], [s268:-65], [s268:-70], [s268:-75], [s268:-80], [s268:-85] , [s268:-90], [s268:-95], [s268:-100], [s268:-105], [s268:-110], [s268:-115], [s268:-1440], [s268:-10080], [s59:-60], [s59:-65], [s59:-70], [s59:-75], [s59:-80], [s59:-85] , [s59:-90], [s59:-95], [s59:-100], [s59:-105] , [s59:-110], [s59:-115], [s59:-1440], [s59:-10080] ], [s13:-60], [s13:-65], [s13:-70], [s13:-75], [s13:-80], [s13:-85] , [s13:-90], [s13:-95], [s13:-100], [s13:-105] , [s13:-110], [s13:-115], [s13:-1440], [s13:-10080]) şeklinde oluşturulmuştur.



**Şekil 5.14:** s268 noktasında 60 dakika sonraki hız tahmini için geriye doğru öz nitelik seçimi.

5 dakikalık hız öngörüsünde olduğu gibi 60 dakikalık hız öngörüsünde de birden fazla sensör kullanıldığında daha başarılı sonuç verildiği s268 sensörü için görülmüştür.

### 5.3 Birbirlerine bağımlı sensörlerin bulunması

s268 hız hesaplamasında yardım için kullanılan s13 ve s59 sensörleri ona yakın konumda olduğu görülerek seçilmiştir. Fakat 327 sensör için kendisini etkileyen sensörleri bulmak için 3.5 bölümünde açıklanan bağıntı yöntemi kullanılacaktır. Her sensör için bağıntısı yüksek sensörler bulunarak onun hız öngörüsünde kullanılarak başarımın artıp artmadığı gözlenecektir. Veri kümeleri bölümünde açıklanan bağıntı yöntemi kullanarak her sensörün bağımlı olduğu sensörler bulunabilir. Tablo 5.3’de birbirlerine en yüksek bağıntıyla bağlı sensörler listelenmiştir.

Bu sensörlerin fiziksel konumlarına bakıldığında gerçekten bunların birbirleriyle aynı yol üzerinde veya birbirlerine bağlanan yollar üzerinde olduğu görülmektedir. Örnek olarak birbirlerine sırasıyla bağımlı s95,s73 ve s4 sensörlerinin aynı yol üzerinde peşpeşe geldikleri Şekil 5.15’te görülmektedir.

**Tablo 5.3:** Yüksek bağımlı sensörler

Sensör 1	Sensör 2
s95	s73
s266	ts266
s156	ts156
s73	s4
s95	s4
s174	s171



**Şekil 5.15:** s95, s73 ve s4'ün harita üzerindeki konumları

Benzer şekilde s266 ve ts266'da harita üzerinde hemen hemen aynı noktadaki trafik hızını ölçmektedir, fakat yolun farklı şeritlerini ölçmektedir. s266 Sarıyer'den – Maslak'a giden yolun hızını ölçerken, ts266 Maslak'tan Sarıyer'e giden yol üzerindeki hızı ölçmektedir.

s156 ve ts156'da karşı şeritlerde olmak üzere E5- Beylikdüzündeki konumlarıdır.

Şekil 5.15, Şekil 5.16 ve Şekil 5.17'den görülebileceği gibi birbirleriyle yüksek korelasyona sahip sensörler birbirlerine yakın konumlarda bulunmakta ve birbirlerine hız bakımından bağımlı olduğu aralarındaki korelasyonun yüksek olduğundan anlaşılmaktadır.



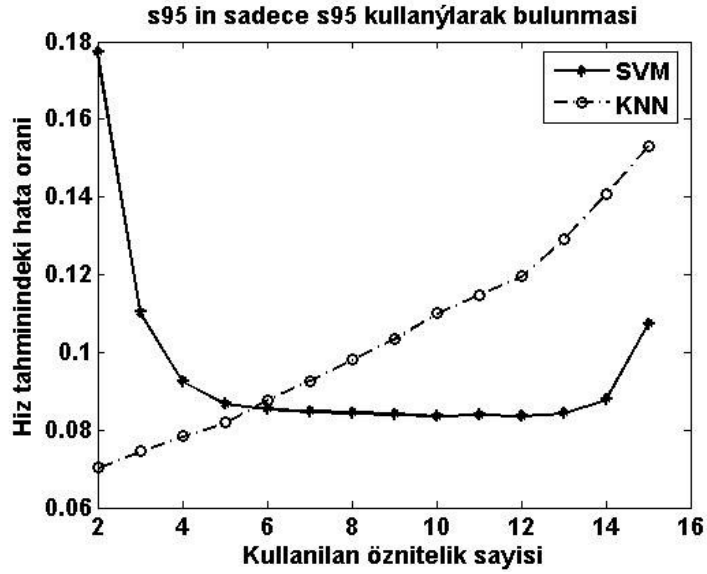
Şekil 5.16: s266 ve ts266'nın harita üzerinde konumları



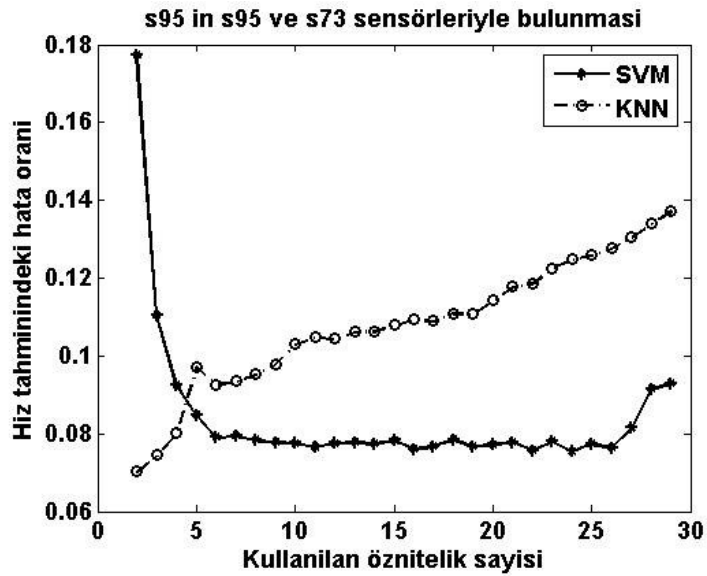
Şekil 5.17: s156 ve ts156'nın harita üzerinde konumları

Şekil 5.18'de s95'in sadece kendisinin kullanılarak hesaplanan hata oranları görünmektedir. Şekil 5.19'da ise s95'in hem s95 hem de s73 kullanılarak hesaplanan hata oranları görünmektedir. Şekil 5.19'a baktığımızda 2 sensör kullanılarak hesaplanan hata oranlarında SVM yönteminin daha önceki örneklere benzer şekilde daha başarılı

olduğu görünmektedir.Şekil 5.19'un şekil 5.18'in hata oranlarıyla karşılaştırdığımızda ise 2 sensör kullandığımızda başarımın tek sensör kullanmaya göre %10 daha iyi olduğu görülmektedir.

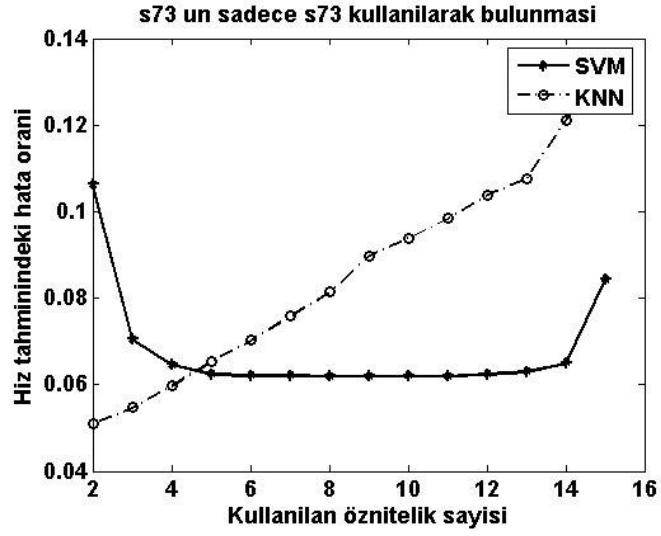


Şekil 5.18: s95 in sadece s95 kullanılarak bulunması

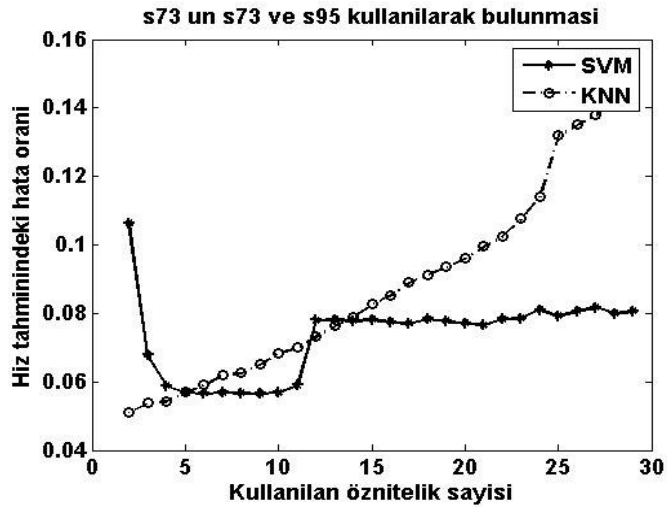


Şekil 5.19: s95 in s95 ve s73 sensörleriyle bulunması

s95 ile s73 yüksek korelasyonla bağılı olduğundan s73 ile s95 de birbirlerine yüksek korelasyonla bağılı olduğundan bahsedilebilir ve s73'ün hız hesaplamalarında s95'in verileri de kullanılabilir. Şekil 5.20 ve şekil 5.21'de s73 sadece kendi verileri kullanılarak ve s95'in verileri de kullanılarak hesaplanan hata oranları görünmektedir. Yine yüksek korelasyonlu 2 sensör kullanıldığında başarımın daha yüksek olduğu görülmektedir.

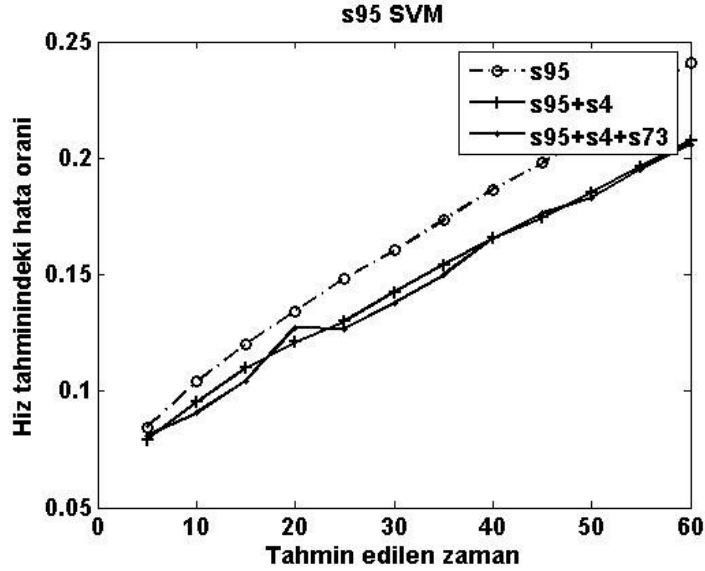


Şekil 5.20: s73 ün sadece s73 kullanılarak bulunması



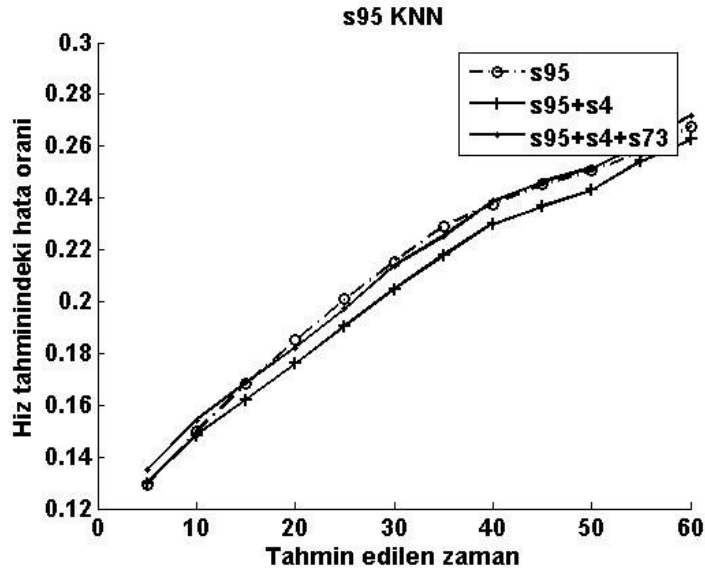
Şekil 5.21: s73'ün s95 ve s73 sensörleriyle bulunması

Şekil 22’de ise 5 dakikadan 60 dakikaya kadar artan zaman aralıklarında sadece s95’in, s95+s4’ün ve s95+s4+s73’ün karşılaştırmalı SVM başarımları verilmiştir. Birden fazla sensör kullanıldığında başarımın daha yüksek olduğu görülmektedir.



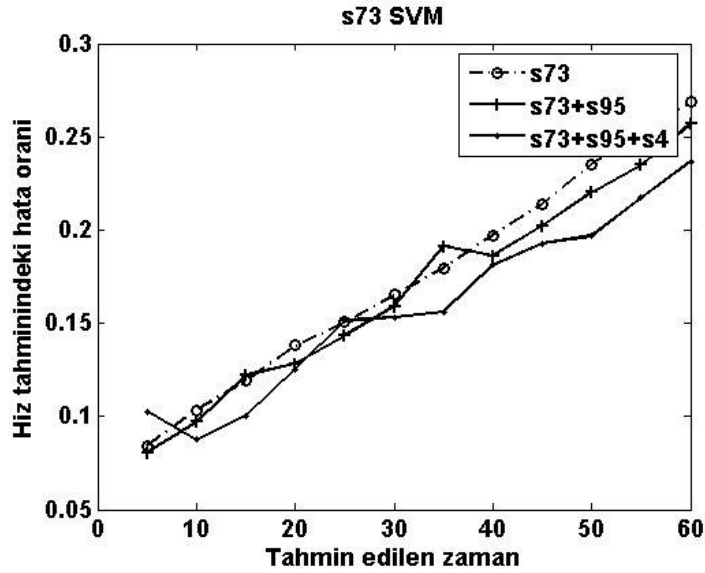
Şekil 5.22: Tahmin edilen zamanın arttırılmasıyla s95 SVM metodunun kullanılması

Şekil 5.23’te ise sadece s95’in, s95+s4’ün ve s95+s4+s73’ün karşılaştırmalı KNN başarımları verilmiştir. Yine birden fazla sensör verisi kullanıldığında başarımın yüksek olduğu görülmektedir.



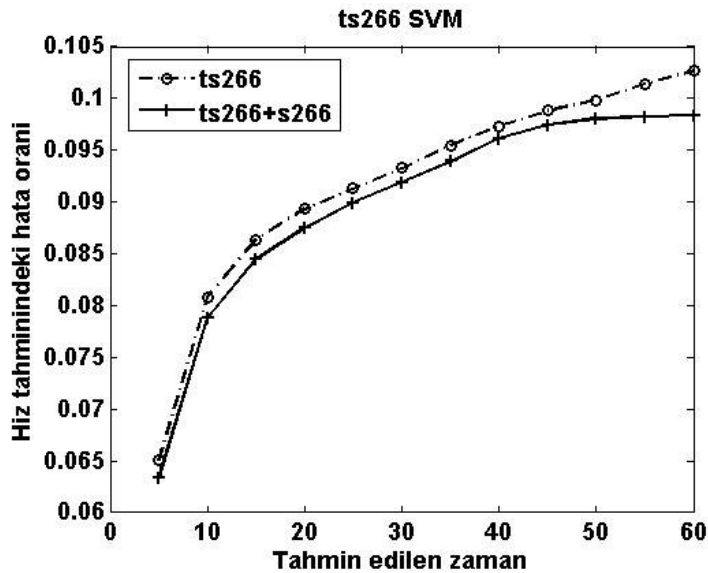
Şekil 5.23: Tahmin edilen zamanın arttırılmasıyla s95 KNN metodunun kullanılması

Şekil 5.24'de s73 sensörünün kendisine yüksek korelasyonla bağlı sensörler ile artan zaman aralıklarında başarımları görülmektedir. Yine birden fazla sensör kullanıldığında başarımlar artmaktadır.

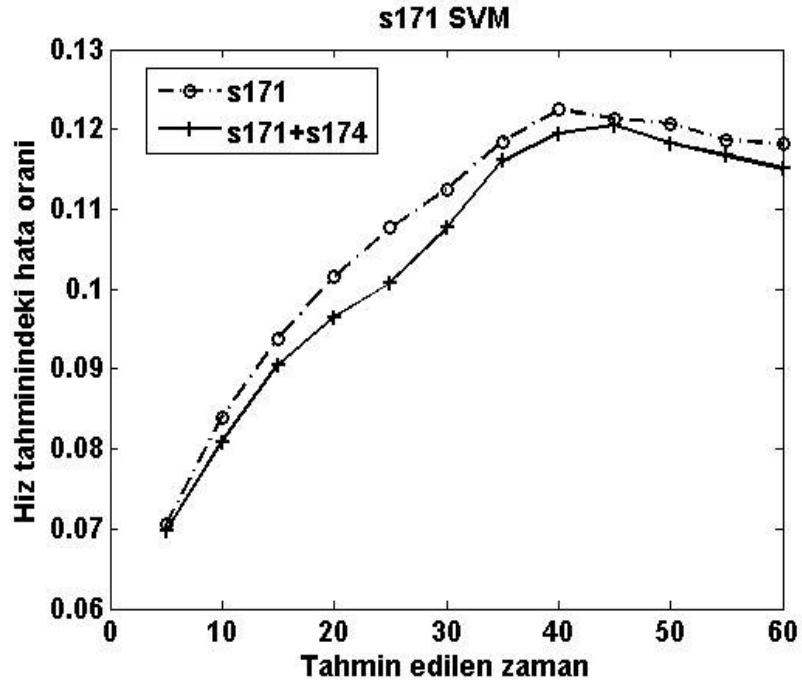


Şekil 5.24: Tahmin edilen zamanın arttırılmasıyla s73 SVM metodunun kullanılması

Şekil 5.25 ve şekil 5.26'da yine yüksek korelasyonlu sensörlerin birbirlkte kullanıldığında başarımların daha yüksek olduğu görülmektedir.



Şekil 5.25: Tahmin edilen zamanın arttırılmasıyla ts266 için SVM metodu

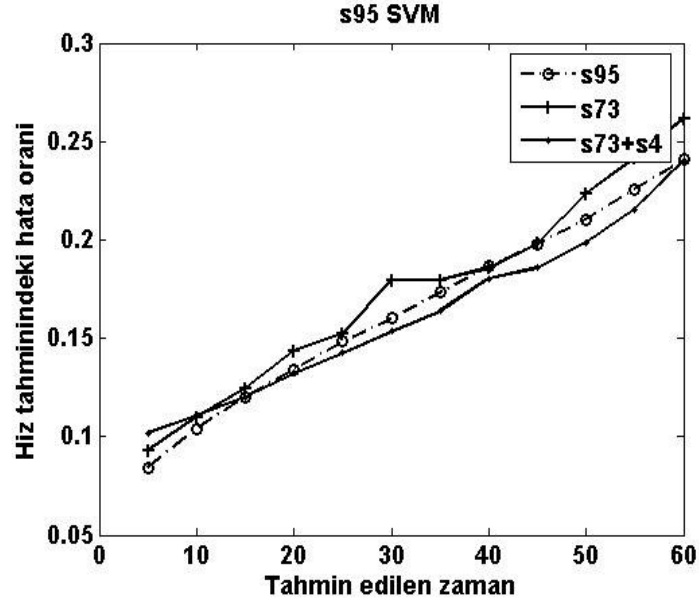


Şekil 5.26: Tahmin edilen zamanın artırılmasıyla s171 için SVM metodu

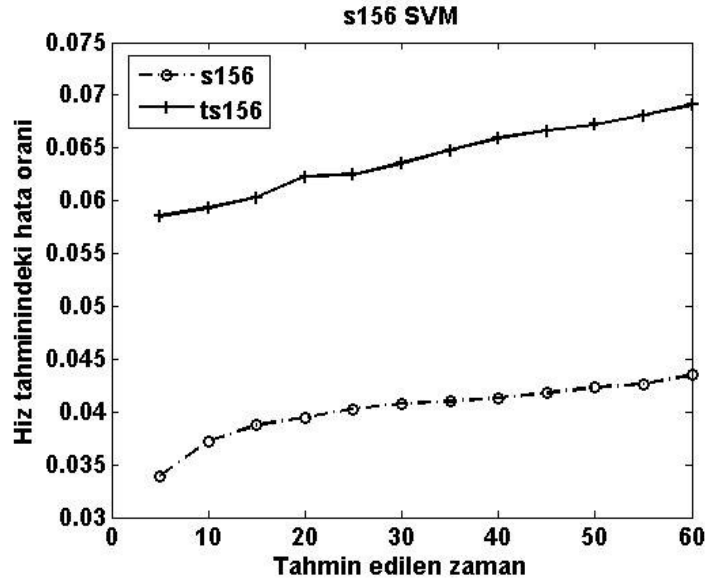
#### 5.4 Yüksek Korelasyonlu Sensörlerin Birbirleri Yerine Kullanılması

Yüksek korelasyonlu sensörlerin birbirlerinin hızlarına benzer hız eğrileri verdiği görülmüştür. Sensörlerin bozulma veya iletişim kopukluğu olduğu durumlarda yüksek korelasyonlu sensörlerin birbirleri yerlerine kullanılabilceği düşünülebilir. Bu bölümde sensörlerin kendi verileri kullanılmadan sadece diğer sensörlerin verileri kullanılarak başarımları hesaplanacaktır. Şekil 5.27’de birbirlerini etkileyen 3 yüksek korelasyonlu sensör kullanılmıştır. Bu sensörlerden s95 yerine s73 veya s73 ile s4’ün birlikte kullanıldığı durumda en az s95 verileri kullanıldığı kadar başarılı sonuçlar alındığı görülmektedir.

Şekil 5.28’de yüksek korelasyonlu s156 sensörünün yerine ts156 sensörü kullanıldığında yapılan hata oranları görülmektedir. %3 civarında bir hata payı ile s156 ile yapılan ölçümlere benzerlik göstermektedir.

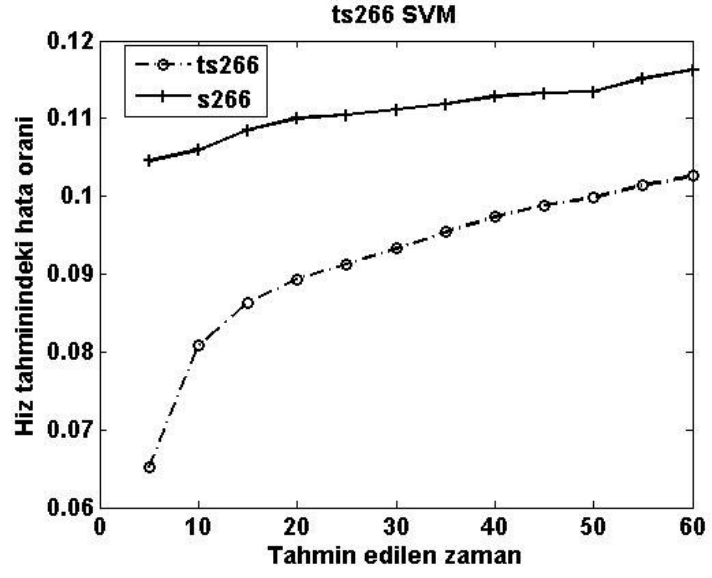


Şekil 5.27: Tahmin edilen zamanın arttırılmasıyla s95 için SVM metodu



Şekil 5.28: Tahmin edilen zamanın arttırılmasıyla s156 için SVM metodu

Şekil 5.29'da yüksek korelasyonlu ts266 sensörünün yerine s266 sensörü kullanıldığında yapılan hata oranları görülmektedir. Yine %3 civarında bir hata payı ile ts266 ile yapılan ölçümlere benzerlik göstermektedir.



Şekil 5.29: Tahmin edilen zamanın arttırılmasıyla ts266 için SVM metodu

## 6. SONUÇLAR VE TARTIŞMA

Bu tezde, hem SVM hem de kNN yöntemleri kullanılarak kısa zamanlı hız tahmini yapılmıştır.

Yapılan testlerde genel olarak bu problem için SVM yönteminin KNN yöntemine göre daha iyi sonuçlar ürettiği görülmüştür. Öznitelik seçme vektörleriyle en iyi öznitelikler karşılaştırıldığında SVM'in öznitelik sayısının 6 ve üzeri olduğu durumlarda başarımının değişmediği görülmüştür. KNN'de ise öznitelik sayısı arttıkça başarım oranı düşmektedir. Ayrıca trafik öngörüsünde 1 gün ve 1 hafta önceki hızların öznitelik olarak kullanıldığında başarımın arttığı görülmüştür.

Öngörü süresi uzadıkça başarım oranları hem SVM hem de KNN için beklenildiği gibi düşmektedir, fakat SVM ve KNN'nin başarımları uzun süreli öngürülerde birbirlerine yaklaşmaktadır. Benzer şekilde trafik sıklığının yoğun olduğu durumlarda başarım oranı düştüğü gözlenmiştir. Bir noktadaki hızı bulurken tek bir sensör verisi kullanmak yerine yakın konumlardaki ve ilgili noktaya bağlantı yollarındaki sensör bilgilerinin kullanılmasının hız tahmini başarımını arttırdığı tespit edilmiştir. Ayrıca birbirlerine yüksek bağıntıyla bağlı sensörler bulunduğu bunların fiziksel olarak birbiriyle yakın konumda bulunan sensörlere karşılık geldiği ve birlikte hız tahmininde kullanılmasının başarımı arttırdığı gözlenmiştir. Sensör verilerinin alınmadığı durumlarda yakın veya yüksek bağıntıya sahip sensör verilerinin kullanılmasının da yüksek başarım sağladığı görülmüştür.

Burada yapılan çalışma ile kısa zaman sonraki hız tespit edilmesi amaçlanmıştır. Daha ileri bir çalışma olarak uzun süreli gözlem ile mevcut yollardaki trafik yoğunluğunun 1-2 yıl sonrası için öngörülmesi düşünülebilir.

## KAYNAKLAR

**Alpaydın, E.**, 2004. Introduction to Machine Learning, MIT Press.

**Balcılar, M. ve Sönmez, C.**, 2008. Trafik Akış Hızının Hareket Vektörleriyle Belirlenmesi, SIU 2008, Didim, Türkiye.

**Beatthetraffic**, 2008. <http://www.beatthetraffic.com>

**Bin, Y., Zhongzhen, Y. Baozhen, Y.**, 2006. Bus Arrival Time Prediction Using Support Vector Machines, *Journal of Intelligent Transportation Systems*, Volume 10, Issue 4 October 2006 , pp. 151 – 158.

**Burges, C.J.C.**, 1998. A Tutorial on Support Vector Machines for Pattern Recognition, *Data Mining and Knowledge Discovery*, Vol. 2, Number 2, p. 121-167.

**Chen M. and S. Chien**, 2001. Dynamic freeway travel time prediction using probe vehicle data: link-based vs path-based. In Transportation Research Board.

**Chrobok, R., Kaumann, O., Wahle, J., Schreckenberg, M.**, 2000. Three Categories of Traffic Data: Historical, Current, and Predictive, the *9th IFAC Symposium Control in Transportation Systems*, pp. 250-25.

**Chrobok, R., Wahle, J. And Schreckenberg, M.**, 2001. Traffic Forecast Using Simulations of large Scale Networks, IEEE Intelligent Transportation System Conference Proceedings –Oakland(CA), USA – August 25-29.

**EIS Electronic**, 2003. ITS Designer's Guide to RTMS Products.

**Hobeika, A.G. and Kim, C.K.**, 1994. Traffic-flow-prediction systems based on upstream traffic, *Proceedings of Vehicle Navigation and Information Systems Conference*, 31 Aug-2 Sep 1994, 345 – 350.

**IBB-TKM**, 2008., <http://tkm.ibb.gov.tr>

**Kwon, J., Coifman, B., Bickel, P.**, 2000. Day-to-day travel time trends and travel time prediction from loop detector data', *Transportation Research Record*, 1554.

**Lingras, P., and Mountford, P.**, 2001. Time Delay Neural Networks Designed Using Genetic Algorithms for Short Term Inter-City Traffic Forecasting Engineering of Intelligent Systems, *14th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA/AIE 2001*, Budapest, Hungary, June 4-7, 2001, Proceedings, page 390.

**Mark, C.D., Sadek, A.W., Rizzo, D.**, 2004. Predicting experienced travel time with neural networks: a PARAMICS simulation study, *Intelligent Transportation Systems*,

2004. Proceedings. The 7th International IEEE Conference on Volume , Issue , 3-6 Oct. 2004 Page(s): 906 – 911.

**Okutani, I. and Y. Stephanedes**, 1984. Dynamic Prediction of Traffic Volume through Kalman Filtering Theory. *Transportation Research, Part B*, Vol.18B, No.1, pp.1-11.

**Park, D., and Ritett, L. R.**, 1998. Forecasting Multiple-Period Freeway Link Travel Times Using Modular Neural Networks, *77th Annual Meeting of the Transportation Research Board*, Washington, D.C., January 1998.

**Park, D. and Rilett, L. R.**, 1999. Forecasting freeway link travel times with a multilayer feedforward neural network, *Computer-Aided Civil and Infrastructure Engineering*, 14(5), 357–367.

**Petty, K., Bickel, P., Jiang, J., Ostland, M., Rice, J., Ritov, Y., and Schoenberg, F.**, 1998. Accurate Estimation of Travel Times from Single-Loop Detectors. *Transportation Research, Part A*, Vol. 32A, No. 1, 1998, pp. 1-17.

**Rice, J., van Zwet, E.**, 2001. A simple and effective method for predicting travel times on freeway, *Intelligent Transportation Systems, IEEE Proceedings*, pp. 227 -232.

**Ruping, S.**, 2004, "mySVM software", Available: <http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM>

**Sheng L.**, 2002., Nonlinear combination of travel-time prediction model based on wavelet network, *IEEE 5th International Conference on Intelligent Transportation Systems, Proceedings*, pp. 741 -746.

**Sun, H., Liu, H., and Ran, B.**, 2003. Short Term Traffic Forecasting Using the Local Linear Regression Model, *Transportation Research Record*.

**TANFB**, 2008. <http://freeway.gov.tw>

**U.S. Highway Administration**, 1999. Intelligent Transportation System Benefit.

**Van Lint, W.C., W.C., Hoogendoorn, S.P., and van Zuylen, H.J.**, 2000. Robust and adaptive travel time prediction with neural networks, *Proceedings of the 6th annual TRAIL Congress (part 2)*, December 2000.

**Wahle, J., Annen, O., Schuster, C., Neubert, L. and Schreckenberg, M.**, 2000. A Dynamic Route Guidance System Based On Real Traffic Data, *Euro. J. Op. Res.*

**Wang, Y. And Nihan, N.L.**, 2003. Can Single-Loop Detectors Do the Work of Dual-Loop Detectors, *ASCE Journal of Transportation Engineering*, Vol. 129, No. 2, pp. 169 – 176.

**Wu, C.H., Ho, J.M., D.T., Lee**, 2004. Travel-time prediction with support vector regression, *Intelligent Transportation Systems, IEEE Transactions on*, Volume 5, Issue 4, 276 – 28, Dec. 2004.

**Yıldırım, Ü. ve Çataltepe, Z.**, 2008. Örüntü Tanıma ve Öznitelik Seçme Yöntemleri Kullanarak Kısa Zaman Sonraki Yol Trafik Hız Öngörüsü, *SIU 2008*, Didim, Türkiye.

**Zhang, L. And Levinson, D.**, 2004. Agent-Based Approach to Travel Demand Modeling. *Journal of the Transportation Research Board*, No. 1898, pp. 28-38.

**Zhang, Z., Rice, J., and Bickel, P.,** 1999. *Empirical Comparison of Travel Time Estimation Methods*, Report for MOU 353, UCB-ITS-PRR-99-43, ISSN1055-1425, December 1999.

**Zhang, X., and Rice, J. A.,** 2001. *Short-Term Travel Time Prediction Using A Time-Varying Coefficient Linear Model*, Technical Report, Dept. Statistics, U.C. Berkeley. March, 2001.

**Zou, N., Wang, J. And Chang, G.,** 2007. A Hybrid Model for Reliable Travel Time Estimation on a Freeway with Sparsely Distributed Detectors, World Congress on ITS, Beijing, China.

## **ÖZGEÇMİŞ**

Ümit Ülkem YILDIRIM, 06/08/1982 tarihinde Ordu'da doğmuştur. Orta eğitimini Ordu Anadolu Lisesi'nde, lise eğitimini Özel Ordu Seçkin Kolej'de tamamladıktan sonra İstanbul Teknik Üniversitesi Bilgisayar Mühendisliği'ne girmeye hak kazanmıştır. 2005'te lisans diplomasını aldıktan sonra 2007 yılına kadar İstanbul Teknik Üniversitesi Bilgi İşlem Merkezi'nde Uzman kadrosunda görev almıştır. 2007 yılından beri İTÜ Teknokent'te bulunan TİKLE firmasında Bilgisayar Mühendisi olarak çalışmaktadır.