TÜRKİYE CUMHURİYETİ
YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRİK - ELEKTRONİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

YOLCULUK ESNASINDA KULLANICI ULAŞIM TÜRÜ TESPİTİ

13011010 — Yunus Emre DEMİRBULUT

BİLGİSAYAR PROJESİ

Danışman
Yrd. Doç. Dr. M. Amaç GÜVENSAN

Ocak, 2017

TEŞEKKÜR

Projenin geliştirilme sürecinde bilgi ve tecrübeleriyle bana yol gösteren ve her türlü imkanı sağlayan bölümümüzün değerli hocalarından aynı zamanda proje danışmanım Sayın Yrd.Doç.Dr. M. Amaç Güvensan'a sonsuz teşekkürlerimi sunuyorum.
Çalışmalarımda bana moral ve destek veren aynı zamanda anlayış gösteren arkadaşlarıma ve aileme tüm kalbimle teşekkür ederim.

Yunus Emre DEMİRBULUT

İÇİNDEKİLER

SİMGE LİSTESİ

CO2e	Karbon Ayak İzi

KISALTMA LİSTESİ

CSV	Comma-Separated Values
GB	Gigabyte
GPS	Global Positioning System
HZ	Hertz
MB	Megabyte
RAM	Random Access Memory

ŞEKİL LİSTESİ

TABLO LİSTESİ

ÖZET

YOLCULUK ESNASINDA KULLANICI ULAŞIM TÜRÜ TESPİTİ

Yunus Emre DEMİRBULUT

Bilgisayar Mühendisliği Bölümü
Bilgisayar Projesi

Danışman: Yrd. Doç. Dr. M. Amaç GÜVENSAN

Yolculuk esnasında kullanıcı ulaşım türü tespiti projesinde, akıllı telefonlarda bulunan ivmeölçer ve jiroskop sensörleri aracılığı ile kullanıcının anlık ulaşım türünün belirlenmesi amaçlanmaktadır.

Kullanıcının ulaşım türünü belirlemek için kullanıcıdan yürürken, araba, otobüs, metrobüs, maramaray, metro, tramvay ve hafif raylı ile seyahat ederken ivmeölçer ve jiroskop algılayıcılarından 100 Hz ile elde edilen veriler toplanmıştır. Veriler özellik çıkarımı ve özellik seçimi işleminden geçirilmiştir. Elde edilen veriler üzerinde makine öğrenmesi teknikleri kullanılarak testler gerçekleştirilmiştir. Veriler 4 farklı makine öğrenmesi algoritması kullanılarak test edilmiştir. Bu algoritmalar Naive Bayes, KNN, J48 ve Random Forest algoritmalarıdır. En iyi sonucu %65 doğruluk oranı ile J48 algoritması vermiştir.

Sonuçlar incelendiğinde metro ile marmaray, otobüs ile metrobüs sınıflarının birbiri ile çok fazla karıştırıldığı görülmüştür. Metro - marmaray ikilisi tek bir sınıf, otobüs - metrobüs ikilisi tek bir sınıf haline getirilmiştir. Yeni veri seti makine öğrenmesi algoritmaları ile test edilmiştir. En iyi sonucu %85 doğruluk oranı ile J48 algoritması vermiştir.

Çalışma sonucunda araba, metro - marmaray, tramvay, hafif raylı ve otobüs - metrobüs olmak üzere 5 farklı ulaşım türü sınıflandırılabilmiştir.

Anahtar Kelimeler: Etkinlik Tanıma, Ulaşım Türü Tespiti, Akıllı Telefon

YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRİK - ELEKTRONİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

ABSTRACT

USER TRANSPORTATION MODE DETECTION DURING TRAVEL

Yunus Emre DEMİRBULUT

Department of Computer Engineering
Computer Project

Advisor: Assist. Prof. Dr. M. Amaç GÜVENSAN

It is aimed to determine the user's instant transport type through the accelerometer and gyroscope sensors found in smartphones in the user transportation mode detection during the journey project.

To determine the user transportation mode, accelerometer and gyroscope sensor data have been collected at 100 Hz while traveling by car, bus, metrobus, maramaray, metro, tram and light rail from the smartphone of the user. The data has been passed through feature extraction and feature selection processes. Machine learning methods were tested on the dataset. The data were tested in 4 different machine learning algorithms. These algorithms are Naive Bayes, KNN, J48 and Random Forest algorithms. The best result was obtained from J48 algorithm with an accuracy of 85%.

When the results were examined, it was seen that subway and marmaray, bus and metrobus classes were mixed with each other too much. Subway - marmaray classes and bus - metrobus classes were transformed into a single class. The new data set has been tested with the machine learning algorithm. The best result is the J48 algorithm with an accuracy of 85%.

As a result of the study, 5 different types of transportation could be classified as car, subway - marmaray, tramway, light rail and bus - metrobus.

Keywords: Activity Recognation, Transportation Mode Detection, Smartphone

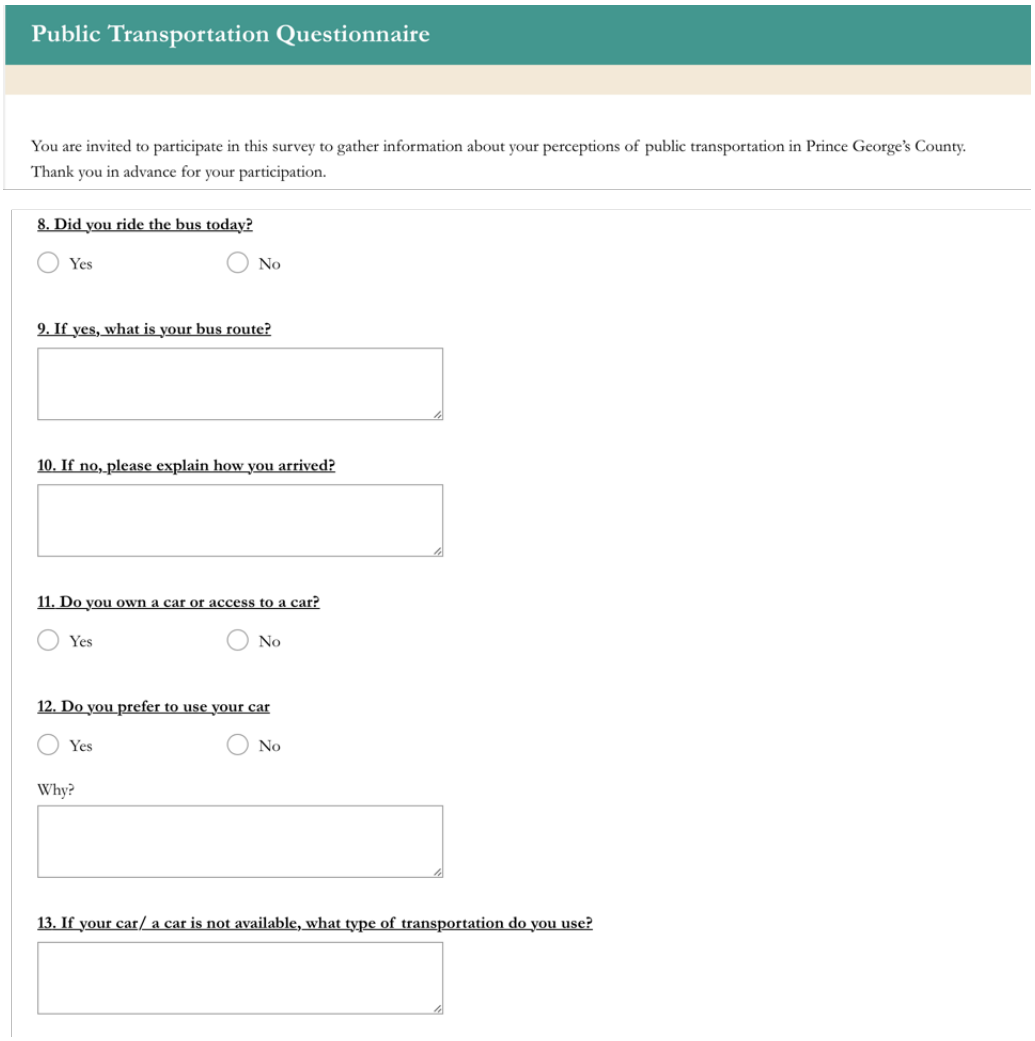
YILDIZ TECHNICAL UNIVERSITY
FACULTY OF ELECTRICAL AND ELECTRONICS
DEPARTMENT OF COMPUTER ENGINEERING

1
Giriş

Günümüz akıllı telefonlarının kolay programlanabilirliği, üçüncü parti uygulamalar için etkin dağıtım olanakları ve artan algılama yetenekleri sayesinde, telefonlar insanların davranışlarını takip etmek için etkili birer araç haline gelmiştir. Tüm bu etmenlerin yanı sıra gelişen donanım teknolojisi, telefonları, kullanıcısına telefon görüşmesinden daha fazlasını sunan cihazlar haline getirmiştir. Akıllı telefonların sahip olduğu GPS, mikrofon, kamera, ivmeölçer ve jiroskop gibi algılayıcıların sağlamış olduğu veriler sağlık, sosyal medya, taşıma, çevresel izleme ve güvenlik gibi çeşitli uygulamaların geliştirilebilmesini mümkün kılmaktadır. Bu çalışmanın konusu kullanıcıların ulaşım türlerinin akıllı telefonlardaki sensörler aracılığı ile tespit edilmesidir.

İnsanların ulaşım davranışlarının akıllı telefonlar aracılığıyla elde edilmesi birçok alanda kolaylıklar sağlamaktadır. Bu alanlardan bir tanesi de şehir planlamasıdır. Şehir planlaması yapılırken insanlardan toplanan veriler Şekil 1.1'deki gibi internet üzerinden veya telefon yoluyla yapılan anketlerden elde edilmektedir. Planlamacılar elde edilen bu anket verilerinden yeteri kadar memnun olmayabilirler. Bunun nedeni modern şehirler karmaşık ve dinamik bir yapı sergilerken anketlerin sadece tek bir noktada veri toplamasıdır. Ulaşım anketi ile ilgili olarak, insanlar ankette anlattıkları yolculuklarını her zaman tam olarak yapmazlar veya insanlar yaptıkları her yolculuğu ankette bildirmezler. Farklı zamanlarda iki veya daha fazla anket yapılmadıkça insanların yolculuklarındaki değişiklikleri ölçmek zordur. Fakat insanlar bu kadar sık yapılan anket taleplerinden kolayca bıkmaktadırlar. Dahası, tekrar tekrar yapılan anketler pahalı ve zaman alıcıdır.

Ulaşım bilgisinin akıllı telefonlar aracılığıyla otomatik olarak elde edilmesi anketlerin tam tersine çok sayıda insandan kolayca ve zahmetsiz bir şekilde veri toplanmasını sağlamaktadır. Bu yöntem birden fazla insanı gerçek zamanlı izleme fırsatı sunmaktadır. Şehirdeki insanların ulaşım davranışlarının gerçek zamanlı olarak elde edilmesini sağlamaktadır. Bunların yanında şehir planlaması yapan kişiler bir şehrin hareketlilik modelini çıkarabilmektedirler. Örneğin, bir şehirdeki insanların ortalama



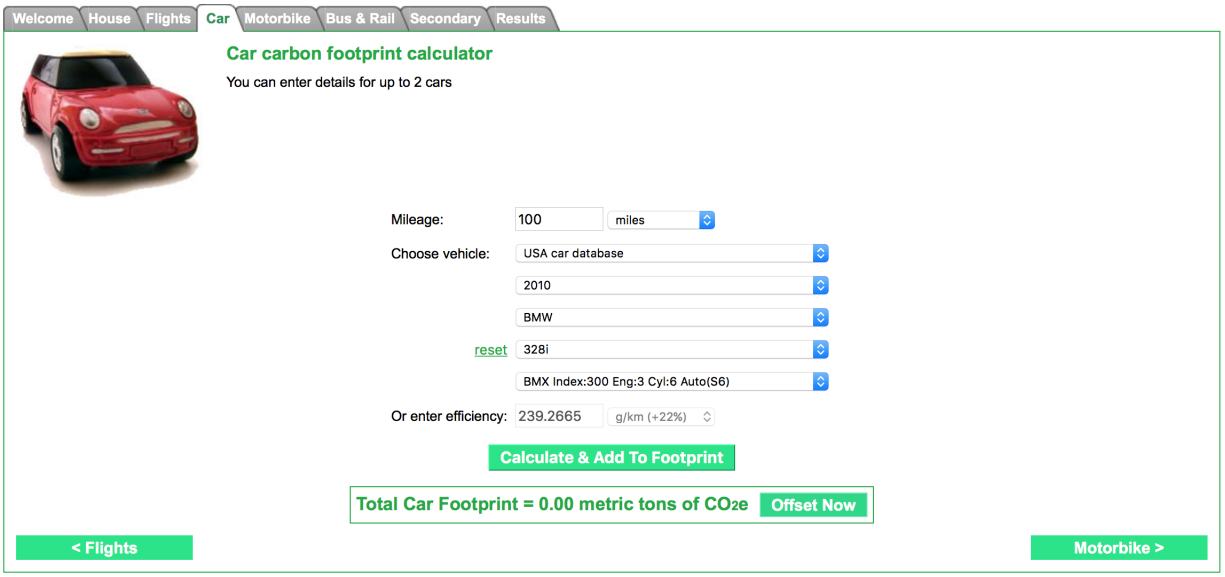
Şekil 1.1 Örnek ulaştırma anketi

hareket mesafeleri elde edilebilmekte, en çok sevilen ulaşım türü belirlenebilmekte ve bu veriler diğer şehirler ile karşılaştırılabilmekte. Şehirdeki insanların belirli saatler içerisinde en çok hangi yolları ve hangi ulaşım araçlarını kullandıkları tespit edilebilir. Bu ve buna benzer bilgiler şehir planlamasında önemli rol oynamaktadır.

Bir diğer önemli kolaylık ise, CO2e (Karbon Ayak İzi) hesaplanması. Endüstriyel Enerji Analizi'ne göre, ulaşım dünya sera gazı emisyonlarının dörtte birini oluşturmaktadir. Kişisel hareketlilik ise toplam ulaşım enerji kullanımının yaklaşık üçte ikisini temsil etmektedir Bu bağlamda, kentin emisyonlarına karşı bireyin kişisel katkısının değerlendirilmesi son derece önemli hale gelmektedir.

Günümüzde kişinin CO2e oranının hesaplanması için Şekil 1.2'de görüldüğü üzeri web uygulamaları kullanılmaktadır. Bu uygulamalar yeteri kadar verimli çalışmamaktadırlar. Bir kişi günlük ortalama CO2e oranını hesaplamak istiyorsa bu web uygulamasına el ile çok fazla sayıda bilgi girmesi gerekmektedir. Bu bilgiler

kullanıcı tarafından girildiği için doğru olmayabilir. Ayrıca sistem otomatik çalışmadığı için kullanıcı hesaplama işlemini kendisi düzenli olarak yapması gerekmektedir.



Şekil 1.2 Örnek karbon ayak izi hesaplama web uygulamsı

Ulaşım türünün otomatik tespiti sayesinde çok sayıda kullanıcının kullandığı vasıta tespit edilebilmektedir. Bu bilginin yanında kullanıcının o anki vasıta ile yapmış olduğu seyehat süresi ve hızı da hasaplanabilmektedir. Tüm bu bilgiler kullanıcının akıllı telefonu cebinde iken elde edilmekte ve otomatik olarak hesaplanmakatır. Bu sayede daha doğru ve etkili bir karbon ayak izi hesabı yapılabilmektedir. Günümüzde bu işlemi yapan CO2GO adında MIT tarafından geliştirilmiş bir uygulama bulunmaktadır. Uygulama Şekil 1.3'de görüldüğü üzere bir çok sensör kullanmaktadır.

Şekil 1.3 CO2Go uygulaması sensör bilgisi

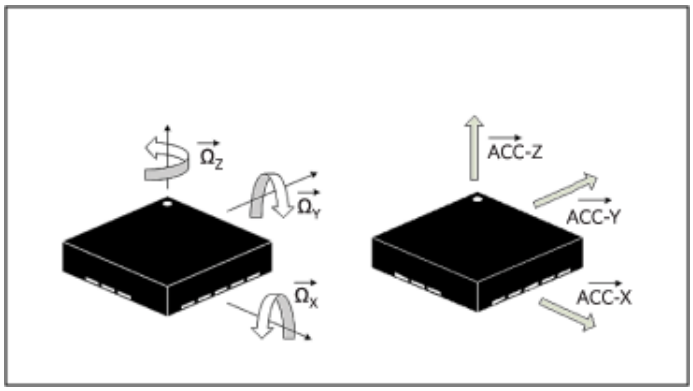
Uygulama, karbon emisyonunun hesaplanmasının yanında kullanıcıların kendilerine ait CO2e oranlarını diğer kullanıcılar ile paylaşmalarına olanak sağlamaktadır. Kullanıcılar bu sayede karbon emisyon oranlarını karşılaştırabilmektedir.

Bu proje kapsamında tespit edilmesi beklenen ulaşım türleri Şekil 1.4'de bulunan İstanbul halkının kullandığı toplu taşımalardır.

Şekil 1.4 Proje kapsamında tespit edilmesi beklenen vasıtalar

Bu toplu taşımalar sırasıyla, otobüs, marmaray, metrobüs, hafif raylı, metro, tramvay ve arabadır.

Proje kapsamında sensör olarak Şekil 1.5'de gösterilen jiroskop (Gyroscope) ve ivmeölçer (Accelerometer) kullanılmıştır. Soldaki jiroskopu, sağdaki ise ivmeölçeri göstermektedir.



Şekil 1.5 Jiroskop ve İvmeölçer

İvmeölçer Şekil 1.5'de gösterilen eksenler doğrultusunda akıllı telefona etki eden ivmeyi ölçer. Ham algılayıcı bilgisi ivmeölçerden üç eksende g cinsinden elde edilir. İvmelenme değerlerinin yanında ayrıca zaman bilgisi de elde edilir. Mevcut çoğu ivmeölçer kullanıcı arayüzünde örnekleme hızını ayarlamaya imkân sunmaktadır. Böylece kullanıcı en uygun örnekleme hızını seçebilmektedir. Proje kapsamında kullanılan örnekleme hızı 100Hz olarak belirlenmiştir. İvmeölçer, akıllı telefon tabanlı eylem tanıma uygulamalarında sıkça kullanılmaktadır. Bu algılayıcının popülerliği konumlandırıldığı cihazın veya taşıyan kullanıcının fiziksel hareketini direkt olarak hesaplayabilmesinden gelmektedir. Örneğin, kullanıcı yürür durumdan zıplar duruma geçerse ivmeölçer sinyallerinin şekli dikey eksende değişir.

Jiroskop ise akıllı telefonun x, y ve z ekseninde yapmış olduğu açısal hızı vermektedir. Jiroskop algılayıcısından elde edilen ham veriler akıllı telefonun üç fiziksel eksen etrafında dönüşünü rad/sn (radyan / saniye) cinsinden bildirmektedir. Karakter yönlendirme yapılan akıllı telefon oyunlarında jiroskop algılayıcısından yararlanılmaktadır. Aktivite tanıma araştırmalarında bu algılayıcı, yön tespitini gerçekleştirmede yardımcı olarak kullanılmaktadır.

Proje raporunun devam eden bölümlerinde sırasıyla ön inceleme, fizibilite ve sistem analizi incelenmektedir. Ön inceleme bölümü, projenin yapıldığı alanda daha önceden yapılmış çalışmaların incelenmesini içerir. Fizibilite bölümü, projenin olabilirliğinin araştırıldığı kısımdır. Projenin zaman planlaması, ekonomik öngörüleri vb. durumları içerir. Sistem Analizi bölümü ise sistemin öge ve işlevlerinin ele alınarak ayrıntılı tanımlanmasını içerir. Bu bölümde projenin hedefleri detaylandırılır.

2
Ön İnceleme

Teknolojinin ilerlemesi kullandığımız elektronik aletlerin donanımlarında da olumlu bir değişim sağlamıştır. 10 yıl öncesine ait bir telefondaki donanım, telefon üzerinde yapabileceğimiz işlemleri kısıtlamaktaydı. Günümüz telefonlarında bulunan donanımlar ise telefonları sadece karşı taraf ile iletişim kurmaktan öteye taşımaktadır. Bahsettiğimiz bu akıllı telefonların sahip olduğu gelişmiş işlemciler ve içerdiği ekstra donanımlar, dokunmatik ekran, sensörler, araştırmacılara yeni çalışma alanları sunmaktadır. Bu çalışma alanlarından bir tanesi de etkinlik tanımadır (Activity Recognition). Ulaşım türü tespiti ise etkinlik tanımanın alt çalışma alanlarından bir tanesidir. Literatür incelendiğinde ulaşım türü tespiti hakkında yapılmış birçok çalışma olduğu görülmektedir.

Yapılan çalışmalardan bir tanesinde [1] akıllı telefonların GPS ve ivmeölçer algılayıcılarından elde edilen veriler ile ulaşım türü tespiti gerçekleştirilmiştir. Çalışmada aşağıda belirtilen 5 farklı ulaşım türünün tespit edilmesi hedeflenmiştir.

- Hareketsiz
- Yürüme
- Koşma
- Bisiklet sürme
- Motorlu taşıtlar

Tablo 2.1'de çalışmanın 5 farklı makine öğrenmesi algoritmasına ait doğruluk değerleri bulunmaktadır. Çalışmada sınıflandırma algoritması olarak Decision Tree (Karar Ağacı) kullanılmıştır. Sensör verileri toplanırken mobil cihaz cepte taşınmıştır.

Tablo 2.1 Sınıflandırıcılara ait doğruluk sonuçları

	Still	Walk	Run	Bike	Motor	All
NB	96.0	87.1	98.4	61.2	93.6	87.2
DT	98.2	96.2	98.6	91.2	94.3	95.7
kNN	97.5	95.2	98.4	91.0	91.2	94.7
SVM	97.8	95.6	98.2	86.9	88.4	93.4
CHMM	96.2	96.1	98.4	89.4,	91.7	94.4

Wang S. , Chen C ve Ma J. tarafından yapılan çalışmada [2] akıllı telefonlarda bulunan algılayıcılardan sadece dâhili ivmeölçeri kullanarak ulaşım türü tespit edilmiştir. Çalışmada aşağıda belirtilen 6 farklı ulaşım türünün tespit edilmesi hedeflenmiştir.

- Hareketsiz
- Yürüme
- Bisiklet sürme
- Otobüs
- Araba
- Metro

GPS teknolojisinin çok fazla enerji tüketmesi nedeniyle çalışmada kullanılmamıştır. Ayrıca GPS sensörleri yer altında veri toplayamamaktadır. Böylece metro gibi yer altından giden araçlar tespit edilememektedir.

Sökün H. , Kalkan H. ve Cetişli B. [3], tarafından yapılan çalışmada üç eksenli ivmeölçerden alınan sinyaller kullanılarak temel fiziksel hareket sınıflandırılması yapılmıştır. Geliştirilen yöntem ile ivmeölçeri taşıyan insanın araç ile mi, yoksa yaya olarak mı seyahat ettiğinin tespit edilmesi amaçlanmıştır. Alınan veriler belirli zaman aralıklarında incelenmiş ve gerçek zamanlı olarak kNN (k En Yakın Komşu) algoritması kullanılarak sınıflandırılmıştır.

Feng ve Timmermans [4] GPS ve ivmeölçer verileri kullanarak ulaşım türü tespiti gerçekleştirmiştir. Yapılan çalışmada üç farklı yöntem incelenmiştir.

- Sadece GPS verisi kullanarak sınıflandırma
- Sadece İvmeölçer verisi kullanarak sınıflandırma
- GPS ve ivmeölçer verisi kullanarak sınıflandırma

Sadece ivmeöçler verisi kullanılarak yapılan tahminler sadece GPS verisi kullanarak yapılan tahminlerden daha iyi performans sergilemiştir. GPS ve ivmeölçer verilerini birleştiren yaklaşım ise en iyi performansı vermiştir. Çalışmada sınıflandırma yöntemi olarak Bayes Ağlarık (Bayesian Belief Networ) kullanılmıştır. Proje kapsamında aşağıda belirtilen 6 farklı ulaşım türünün tespit edilmesi hedeflenmiştir.

- Yürüme
- Koşma
- Bisiklet sürme
- Motor sürme
- Otobüs
- Araba
- Tramvay
- Metro

Stenneth ve Wolfson [5] telefon ve coğrafi bilgi sistemleri bilgileri ile ulaşım türü tespiti üzerinde çalışmıştır. Çalışma kapsamında aşağıda belirtilen 6 farklı ulaşım türünün tespit edilmesi hedeflenmiştir.

- Tren
- Hareketsiz
- Yürüme
- Bisiklet sürme
- Otobüs
- Araba

Toplanan veriler beş (Bayesian Net, Decision Tree, Random Forest, Naïve Bayesian, Multilayer Perceptron) farklı makine öğrenmesi algoritması ile test edilmiştir. En iyi sonucu %93 doğruluk oranıyla Random Forest algoritması vermiştir.

Bu proje kapsamında daha önceden yapılan çalışmalardan farklı olarak ivmeölçerin yanında jiroskop sensörü de kullanılmıştır. İki sensörden toplanan veriler dört farklı makine öğrenmesi algoritmasında test edilip en iyi sonucu veren algoritmaya ait model sınıflandırma işlemi için kullanılmıştır. Kullanılan Makine öğrenmesi algoritmaları aşağıdaki gibidir.

- Naive Bayes
- kNN
- J48
- Random Forest

GPS sensörü fazla enerji tüketmesi ve yer altında veri toplayamaması nedeniyle tercih edilmemiştir.

3
Fizibilite

Bu bölümde projenin olabilirlik etüdü yapılmaktadır. Projenin teknik, zaman, yasal ve ekonomik fizibilitesi incelenmektedir.

3.1 Teknik Fizibilite

Projenin teknik fizibilitesi, yazılım fizibilitesi ve donanım fizibilitesi olmak üzere iki ayrı başlık altında işlenmiştir.

3.1.1 Yazılım Fizibilitesi

Geliştirilen uygulamanın iPhone cihazlarda çalışacak olması sebebiyle iOS işletim sistemi tercih edilmiştir. Projenin geliştirilmesi için XCode geliştirici ortamı kullanılmıştır. iOS işletim sistemi için geliştirilen uygulamalar sadece XCode geliştirici ortamında gerçekleştirilebilmektedir. Apple tarafından yapılan bu kısıtlama sebebiyle XCode kullanılmıştır. Projeyi geliştirirken kullanılacak programlama dili ise iOS işletim sistemi tarafından desteklenen, yeni, esnek ve anlaşılabilir bir programlama dili olan Swift dilidir. Objective - C karmaşık ve eski bir programlama dili olması sebebiyle projenin geliştirilme aşamasında kullanılmamıştır. Veritabanı aracı olarak ise XCode ortamı ile beraber gelen CoreData sistemi kullanılmıştır. iOS işletim sistemleri üzerinde Swift programlama dili ile beraber oldukça verimli çalışan bir veritabanı sistemidir. Projenin makine öğrenmesi kısmında ise açık kaynak kodlu bir araç olan Weka kullanılmıştır. Açık kaynak kodlu ve basit ara yüzü sebebiyle tercih edilmiştir.

3.1.2 Donanım Fizibilitesi

Proje, iPhone cihazlar üzerinde çalışmak üzere tasarlanmıştır. Verilerin elde edildiği iki tür sensör bulunmaktadır. İvmeölçer (Accelerometer) ve jiroskop (Gyroscope). Projede verileri elde etmek için kullanılan cihazın teknik özellikleri

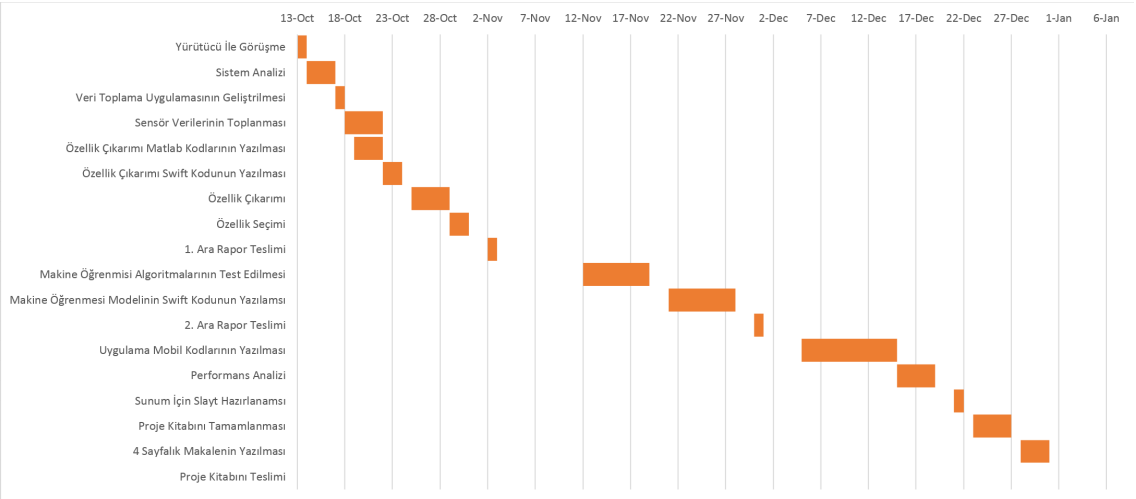
- 64 bit A8 Mikroişlemci
- M8 Hareket İşlemcisi
- 1GB RAM
- 16GB Disk Kapasitesi

Projenin gerçekleştirildiği bilgisayara ait teknik özellikler

- 2.4 GHz Intel Core i5 İşlemci
- 8GB RAM
- Intel Iris 1536 HD Grafik Kartı
- 256GB SSD Disk Kapasitesi

Proje iOS 10 yüklü , ivmeölçer ve jiroskop bulunan tüm iPhone cihazlarda çalışabilecek şekilde tasarlandı.

3.2 Zaman Fizibilitesi



Şekil 3.1 Zaman fizibilitesine ait Gantt diyagramı

3.3 Yasal Fizibilite

Geliştirilmekte olunan proje mevcut kanun ve yükümlülüklere uygun olup, herhangi bir patent vb. korunmuş hakkı ihlal etmemektedir

3.4 Ekonomik Fizibilite

Proje kapsamında kullanılan yazılım teknolojileri ücretsiz olup sisteme ek bir maliyet getirmemektedir. Bu sistemi geliştirmek için gerekli olan donanım maliyeti Tablo 3.1 'deki gibidir. Çalışan maliyeti Tablo 3.2 'deki gibidir. Sensör verilerinin toplanması için gerekli olan Ulaşım Giderleri Tablo 3.3 ' deki gibidir.

Tablo 3.1 Donanım Giderleri

Cihaz	**Fiyat (TL)**
Kişisel Bilgisayar	3.500
Test Telefonu	2.000
Toplam	***5.500***

Tablo 3.2 Çalışan Giderleri

Görev	**Süre (Ay)**	**Aylık Ücret**	**Toplam Gider (TL)**
Makine Öğrenmesi Modeli Oluşturma	3	1.500	4.500
iOS Mobil Uygulama Geliştirme	1	2.500	2.500

Tablo 3.3 Ulaşım Giderleri

Ulaşım Türü	**Fiyat (TL)**	**Süre (Dk)**
Metro	10	40
Hafif Raylı	10	40
Metrobüs	12	40
Marmaray	10	40
Otobüs	5	45
Tramvay	10	40
Araba	60	45
Toplam	***117***	***290***

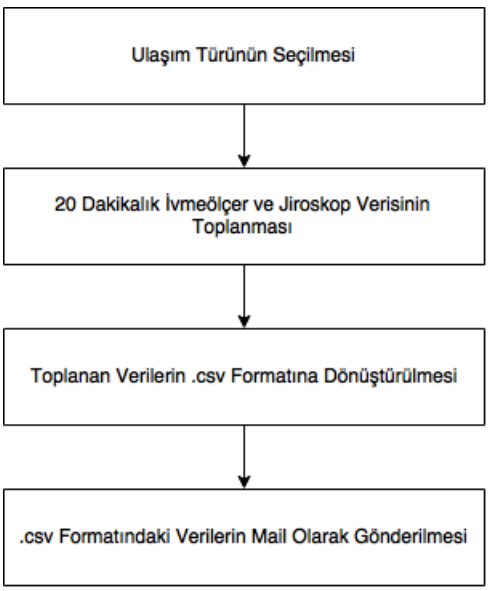
Tablo 3.4 Toplam Gider

Gider	**Fiyat (TL)**
Donanım Giderleri	5.500
Yazılım Giderleri	0
Çalışan Giderleri	7.000
Ulaşım Giderleri	112
Toplam	***12.612***

4
Sistem Analizi

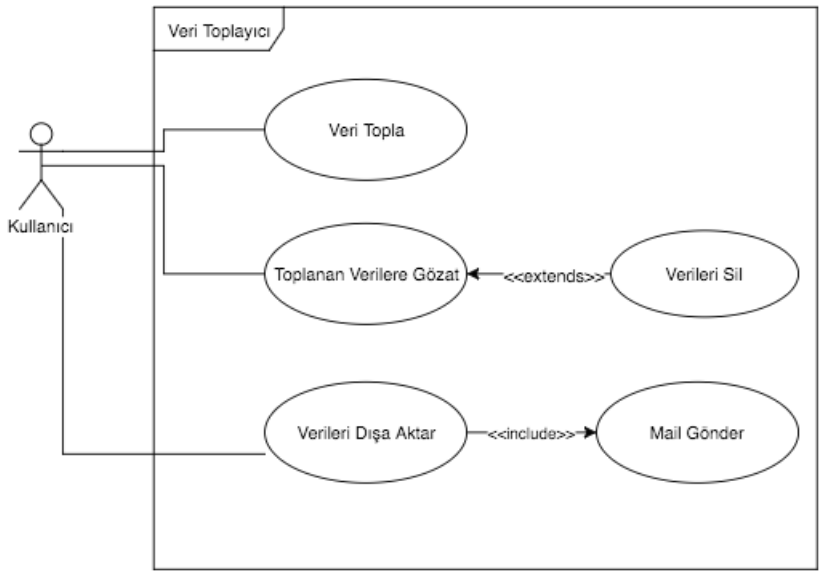
Projenin hedefi, eldeki sensör verilerinden çıkarımda bulunarak kullanıcının o anki ulaşım türünün tespit edilmesidir. Sensör verilerinin işlenmesi ve çıkarımda bulunulabilmesi için ise çıkarım mekanizmasına ihtiyaç duyulmaktadır. Bu mekanizmayı oluşturmak için makine öğrenmesi teknikleri kullanılmıştır. Makine Öğrenmesi (Machine Learning), matematiksel ve istatistiksel yöntemler kullanarak mevcut verilerden öğrenim modeli veya başka bir deyişle çıkarım mekanizması oluşturur. Bu model aracılığı ile bilinmeyene dair tahminlerde bulunur.

Projede makine öğrenmesi tekniklerinin kullanılması bazı ihtiyaçları ortaya koymaktadır. Yukarıda belirtilen çıkarım mekanizmasının oluşturulabilmesi için her bir vasıta ile yapılmış yolculuklara ait ivmeölçer ve jiroskop sensör verilerine ihtiyaç duyulmaktadır. İhtiyaç duyulan bu veri seti Şekil 4.1'de genel yapısı gösterilen sistem aracılığı ile elde edilmiştir.



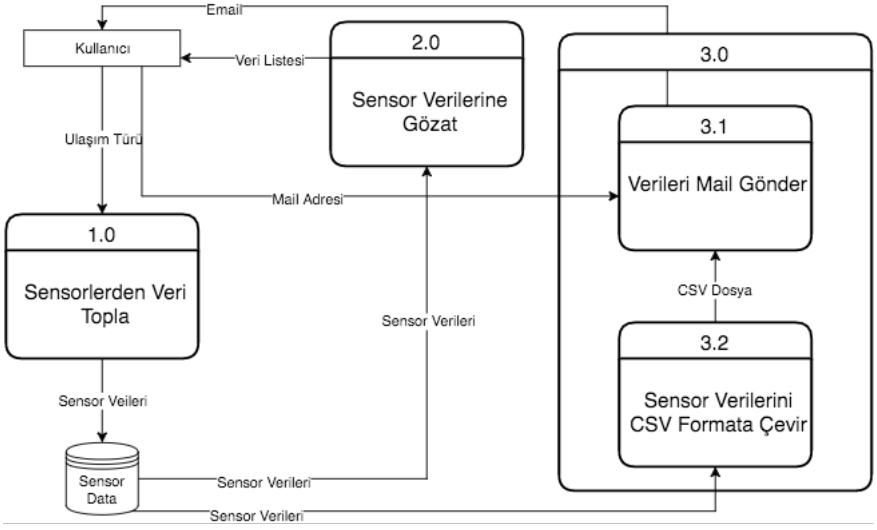
Şekil 4.1 Sensör veri seti oluşturmak için gerekli sistemin genel diyagramı

Tasarlanan sisteme ait kullanım senaryosu Şekil 4.2'de ve veri akış diyagramı Şekil 4.3'de görülmektedir.



Şekil 4.2 Sensör veri toplama uygulamasına ait kullanım senaryosu

Kullanıcı veri toplayacağı araç tipini belirledikten sonra başlama komutunu verir ve jiroskop ile ivmeölçere ait veriler toplanmaya başlanır. Bu veriler 20 dakika toplandıktan sonra veritabanına kaydedilir. Veritabanında her bir sensörün 3 eksenine ait veriler ve verilerin toplandığı araç tipi bulunmaktadır. Kullanıcı toplanan verileri listeleyebilmektedir. Yanlış veri toplanmış ise kullanıcı bu verileri silebilmektedir. Toplanan verilerin bilgisayar ortamına aktarılması gerekmektedir. Verilerin mail yoluyla gönderilmesi tercih edilmiştir. İlk olarak sensör verileri CSV formatına dönüştürülür. Daha sonrasında bu dosya alıcıya mail olarak gönderilir.

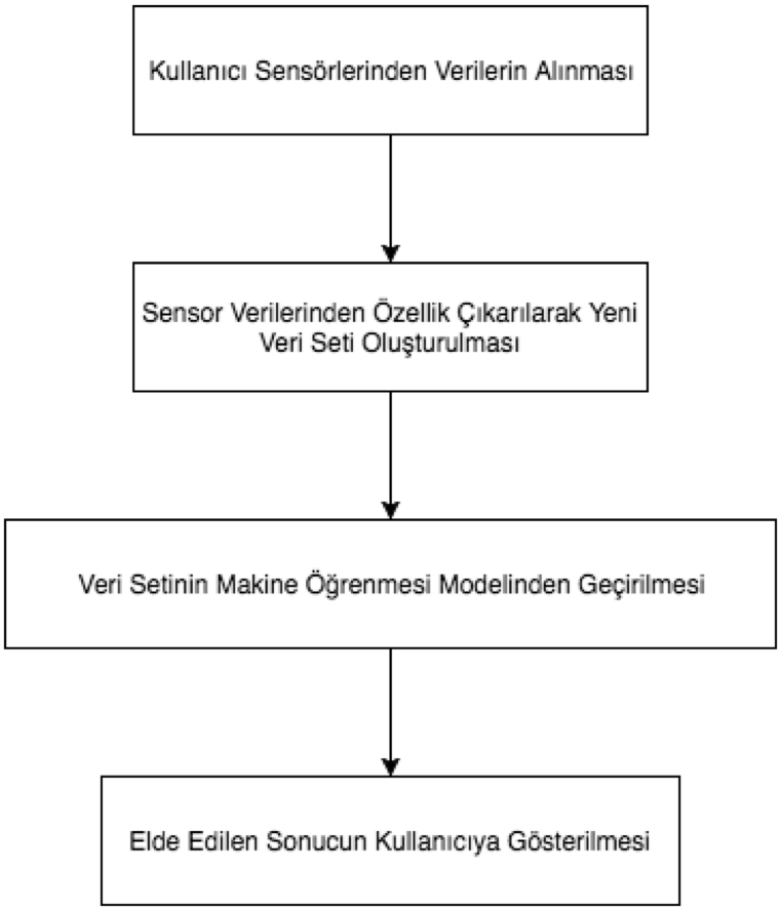


Şekil 4.3 Sensor Veri Toplayıcısına ait veri akış diyagramı

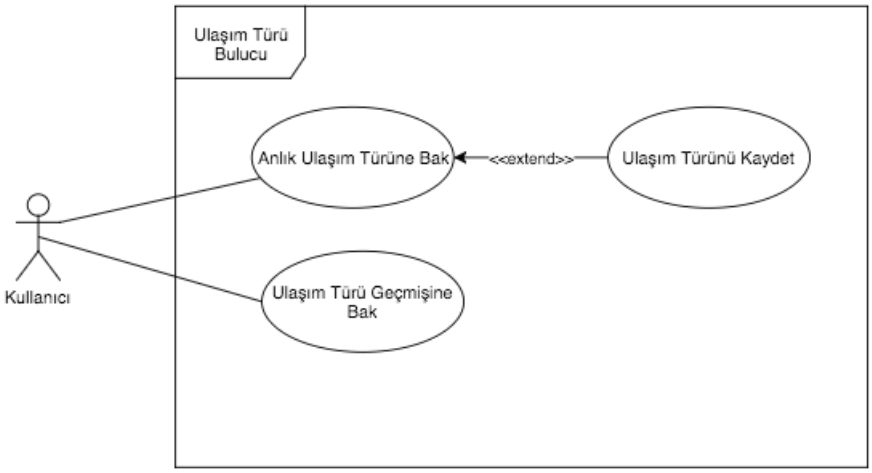
Sensör verilerinden oluşan ham veri setinin makine öğrenmesi algoritmalarına girdi olarak verilebilmesi için hazır olması gerekmektedir. Ham veri setine ait veriler ilk olarak özellik çıkarımı işleminden geçirilir. Veri setinden çıkarılacak özellikler

daha önceden eğitim verisi üzerinde belirlenmektedir. Özellik çıkarımı işlemi ile beraber oluşturulan yeni veri seti makine öğrenmesi modelinden geçirilecek hale gelmiş olur. Uygulama içerisinde kullanılacak olan modelin belirlenmesi işlemi Weka üzerinde daha önceden yapılır. Toplanan eğitim verileri özellik çıkarımı ve özellik seçimi işleminden geirildikten sonra elde edilen yeni veri seti makine öğrenmesi algoritmalarından geçirilir. En yüksek doğruluk değerini veren algoritmanın oluşturduğu makine öğrenmesi modeli uygulama içerisinde kullanılır.

Kullanıcının ulaşım türünün tespit eden sistemin genel akış diyagramı Şekil 4.5'de görülmektedir.

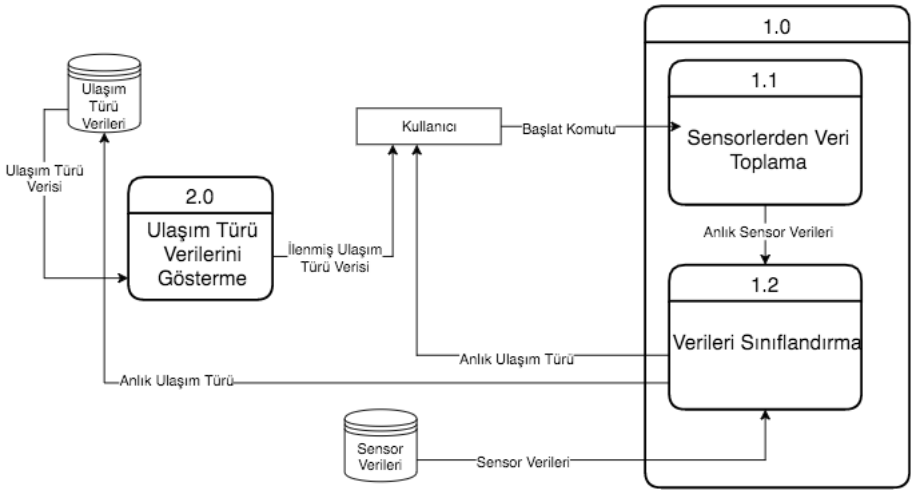


Şekil 4.4 Sistem yapısına ait blok diyagram



Şekil 4.5 Ulaşım türünü tespit eden uygulamaya ait kullanım senaryosu

Veri seti makine öğrenmesi modelinden geçirildikten sonra elde edilen sonuç kullanıcıya gösterilir. Sonuç bilgisi, kullanıcının anlık ulaşım türünün hangi vasıta olduğunu içerir. Bu bilgi daha sonrasında veritabanına kaydedilir. Bu sayede kullanıcı ulaşım türü geçmişine bakabilmektedir. Aşağıda kullanıcının anlık ulaşım türünün tespit edilmesini sağlayan uygulamanın kullanım senaryosu ve veri akış diyagramı bulunmaktadır.



Şekil 4.6 Ulaşım türünü tespit eden uygulamaya ait veri akış diyagramı

5
Sistem Tasarımı

Sistem analizinden sonra sistem için gerekli modüller tasarlanmıştır. Tasarlanan sistemin ana işlem adımları temel olarak aşağıda listelenmiştir.

- Kullanıcının anlık sensör verilerinin toplanması.
- Elde edilen verilerin işlenmesi.
- İşlenen verilerin makine öğrenmesi modelinden geçirilmesi.
- Sonucun kullanıcıya gösterilmesi.

5.1 Yazılım Tasarımı

Sistemin ilk aşaması çıkarım mekanizmasının yani makine öğrenmesi modelinin oluşturulmasıdır. Modelin oluşturulabilmesi için Şekil 5.1'de görülen ham sensör verilerinin işlenmesi gerekmektedir. Şekil 5.1'deki ham veriler her bir vasıta için iki farklı zaman diliminde yirmişer dakika toplanmıştır. İlk toplanan 20 dakikalık veriler makine öğrenmesi modelini eğitmek için kullanılmıştır. İkinci toplanan veriler ise makine öğrenmesi modelinin doğrulunu test etmek için kullanılmıştır. Sonuç olarak her vasıtaya ait 20 dakikalık test, 20 dakikalık eğitim veri seti oluşturulmuşur. Sensör verileri 100 Hz ile toplanmıştır.

ax	ay	az	gx	gy	gz	mode
0.012343	0.10188	-0.04593	-0.0062	0.058268	0.02522	ARABA
-0.05377	0.064119	-0.07626	0.030272	-0.01351	-0.07595	ARABA
-0.01173	-0.00976	0.015882	0.017654	0.049388	0.005145	ARABA
0.008937	0.056711	-0.03748	-0.08014	-0.05987	-0.19368	ARABA
-0.08241	-0.02372	-0.04541	-0.02387	-0.03956	-0.27168	ARABA

Şekil 5.1 Araba sınıfına ait ham sensör verisi

İvmeölçer ve jiroskop sensörlerden toplanan bilgiler ise aşağıdaki gibidir.

- AX: İvmeölçer X-Koordinatı (Accelerometer X-Coordinate)
- AY: İvmeölçer Y-Koordinatı (Accelerometer Y-Coordinate)
- AZ: İvmeölçer Z-Koordinatı (Accelerometer Z-Coordinate)

- GX: Jiroskop X-Koordinatı (Gyroscope X-Coordinate)
- GY: Jiroskop Y-Koordinatı (Gyroscope Y-Coordinate)
- GZ: Jiroskop Z-Koordinatı (Gyroscope Z-Coordinate)

Elde edilen eğitim veri seti ve test veri seti özellik çıkarımı işleminden geçirilmiştir. Tablo 5.1'de görülen toplan 19 istatistiksel özellik her bir eksen için çıkarılmıştır. Altı farklı eksen için toplam 114 özellik çıkarılmıştır. Özellik çıkarımı 20 saniye, 40 saniye ve 60 saniye olmak üzere 3 farklı pencere aralığı için gerçekleştirilmiştir. Özellik çıkarımı sonucunda

- 20 saniye pencere aralığına sahip test ve eğitim veri seti
- 40 saniye pencere aralığına sahip test ve eğitim veri seti
- 60 saniye pencere aralığına sahip test ve eğitim veri seti

olmak üzere toplamda 6 farklı veri seti elde edilmiştir.

Özellik çıkarımı işlemi yapıldıktan sonra özellik seçimi işlemi gerçekleştirilmiştir. Özellik seçimi, sınıflandırma işlemi için kullanılacak özelliklerin belirlenmesi aşamasında, tüm özellik kümesi sütunlarından bağımlı değişkenle olan ilişkiyi açıklamada, ilgisiz sütunların elenmesi ve açıklayıcı gücü yüksek sütun alt kümelerinin belirlenmesi işlemidir. Toplam 114 olan özellik sayısı bu aşamada düşürülmüştür. Özellik seçimi alogirtmaları eğitim veri seti üzerinde çalıştırılır. Algoritmalar sonuç olarak kullanılması gereken özelliklerin listesini sunar. Listelenen özellikler dışındaki özellikler hem eğitim hem de test veri setinden çıkarılarak yeni veri setleri elde edilir.

- 20 saniye pencere aralığı için 23 özellik seçilmiştir.
- 40 saniye pencere aralığı için 20 özellik seçilmiştir.
- 60 saniye pencere aralığı için 15 özellik seçilmiştir.

Tablo 5.1 Özellik çıkarımı için kullanılan özellikler

Özellik	Açıklama
Minimum Reduction	Ardışık veriler arasındaki minimum azalma miktarı
Maximum Reduction	Ardışık veriler arasındaki maksimum azalma miktarı
Minimum Increase	Ardışık veriler arasındaki minimum artma miktarı
Maximum Increase	Ardışık veriler arasındaki maksimum artma miktarı
Minimum Value	Veriler içerisindeki en küçük değer
Maximum Value	Veriler içerisindeki en büyük değer
Range	Veriler içerisindeki maksimum ile minimum değer farkı
Arithmetic Mean	Verilerin aritmetik ortalaması
Square mean	Verilerin karelerinin ortalaması
Harmonic Mean	Verilerin harmonik ortalaması
Geometric mean	Verilerin geometrik ortalaması
Quadratic mean	Verilerin kuadratik ortalaması
Median	Veri içerisindeki orta değer
Standard Deviation	Verilerin standart sapması
Variance	Verilerin varyansı
Coefficient of Variation	Verilerin varyasyon katsayısı
Kurtosis	Verilerin dağılımını gösteren çan eğrisinin basıklık derecesi
Skewness	Verilerin dağılımın ortalamaya göre simetrisizliği
Interquartile Range	Verilerin çeyrekler açıklığı

Özellik seçimi yapıldıktan sonra elde edilen yeni eğitim veri setleri makine öğrenmesi algoritmalarından geçirilmiştir. Aşağıda belirtilen 4 farklı makine öğrenmesi algortiması kullanılmıştır.

- Naive-Bayes
- KNN
- J48
- Random Forest

Naive-Bayes algoritması eğitim verisi üzerinden öğrenme işlemini gerçekleştirir ve en yüksek orandaki örneğini sınıfa dahil eder. Algoritma verinin hangi sınıfa ait olduğunu belirlemek için Bayes Teoremi olasılığını hesaplar.
KNN algoritmasında (Şekil 5.2) sınıflandırılacak olan yeni örneğe eğitim setinden en yakın mesafedeki k tane örneğe bakılır ve bu k örnek çoğunluk olarak hangi sınıfa dâhil edilmiş ise yeni örnek de o sınıfa dâhil edilir. Mesafe hesabı olarak ise yaygın bilinen Manhattan uzaklık ölçütü, Öklid uzaklık ölçütü ve Minkowski uzaklık ölçütü formüllerinden bir tanesi kullanılır.

```
k-Nearest Neighbor
Classify (X, Y, x) // X: training data, Y: class labels of X, x: unknown sample
for i = 1 to m do
  Compute distance d(X_i, x)
end for
Compute set I containing indices for the k smallest distances d(X_i, x).
return majority label for {Y_i where i ∈ I}
```

Şekil 5.2 KNN algoritmasına ait sözde kod

J48 algoritması (Şekil 5.3) bilgi entropi kavramı kullanarak bir eğitim setinden karar ağacı inşa eder. Ağacın her bir düğümünde J48, alt kümelerdeki zenginleştirilmiş verinin niteliğini seçer. Bölünme ölçütü normalize bilgi kazancıdır. En yüksek normalize bilgi kazancına sahip nitelik karar için seçilir. Sonrasında J48 algoritması daha küçük alt listelerden çekilir.

```
Input: an attribute-valued dataset D
  if D is "pure" OR other stopping criteria met then
     terminate
  end if
  for all attribute a ∈ D do
     Compute information-theoretic criteria if we split on a
  end for
  a_best = Best attribute according to above computed criteria
  Tree_v = Create a decision node that tests a_best in the root
  D_v = Induced sub-datasets from D based on a_best
  for all D_v do
     Tree_v = C4.5(D_v)
     Attach Tree_v to the corresponding branch of Tree
  end for return Tree
```

Şekil 5.3 J48 algoritmasına ait sözde kod

Random Forest algoritmasında (Şekil 5.4) amaç tek bir karar ağacı üretmek yerine her biri farklı eğitim kümelerinde eğitilmiş olan çok sayıda çok değişkenli ağacın kararlarını birleştirmektir. Farklı eğitim kümeleri oluştururken ön yükleme ve rastgele özellik seçimi kullanılır. Çok değişkenli karar ağaçları oluşturulurken CART algoritması kullanılır. Her seviyedeki özniteliği belirlerken önce bütün ağaçlarda hesaplamalar yapılarak nitelik belirlenir, ardından bütün ağaçlardaki nitelikler birleştirilerek en fazla kullanılan öznitelik seçilir. Seçilen nitelik ağaca dahil edilerek diğer seviyelerde aynı işlemler tekrarlanır.

Precondition: A training set $S := (x_1, y_1), \ldots, (x_n, y_n)$, features F, and number of trees in forest B.

```
function RandomForest(S, F)
    H ← ∅
    for i ∈ 1, ..., B do
        S^(i) ← A bootstrap sample from S
        h_i ← RandomizedTreeLearn(S^(i), F)
        H ← H ∪ {h_i}
    end for
    return H
end function
function RandomizedTreeLearn(S, F)
    At each node:
        f ← very small subset of F
        Split on best feature in f
    return The learned tree
end function
```

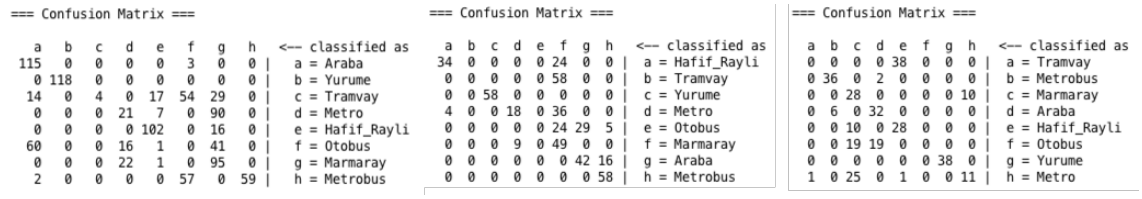
Şekil 5.4 Random Forest algoritmasına ait sözde kod

20 saniye, 40 saniye ve 60 saniye pencere aralığına sahip eğitim veri setlerinin yukarıda belirtilen makine öğrenmesi algoritmalarından geçirilmesi sonucu elde edilen doğruluk değeleri Tablo 5.2'deki gibidir. Tablo 5.2 incelendiğinde 60 saniye pencere aralığı oldukça düşük doğruluk oranına sahiptir. 20 saniye ve 40 saniye pencere aralığı daha iyi sonuç üretmektedir. En iyi sonuç 20 saniye pencere aralığında J48 algoritmasından alınmaktadır.

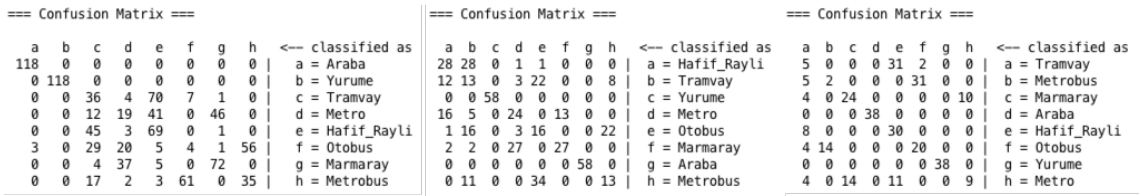
Tablo 5.2 Eğitim veri setlerinin doğruluk değerleri

	20 Saniye	40 Saniye	60 Saniye
Naive Bayes	% 54.4492	% 55.819	% 56.9079
KNN	% 49.8941	% 51.0776	% 54.6053
J48	% 65.1483	% 61.2069	% 54.2763
Random Forest	% 60.5932	% 60.9914	% 58.2237

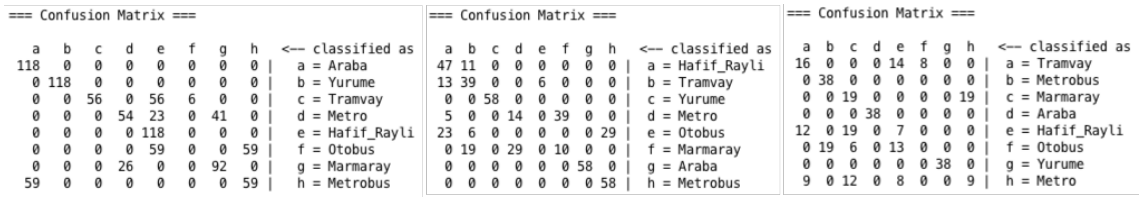
Tablo 5.2'de belirtilen doğruluk değerlerine ait karmaşıklık matrisleri; Naive Bayes algoritması için Şekil 5.5'de, KNN algoritması için Şekil 5.6'da, J48 algoritması için Şekil 5.7'de ve son olarak Random Forest algortması için Şekil 5.8'de gösterilmiştir.



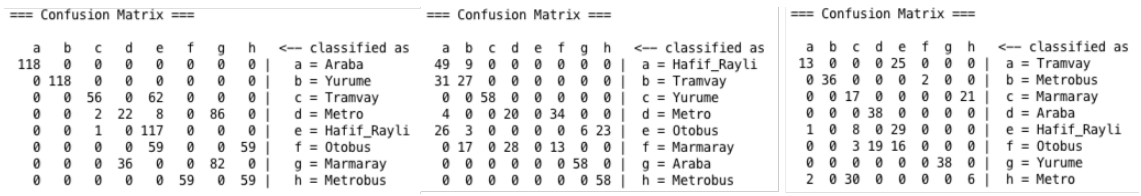
Şekil 5.5 Naive Bayes algoritmasının sırasıyla 20 - 40 - 60 saniye için karmaşıklık matrisi



Şekil 5.6 KNN algoritmasının sırasıyla 20 - 40 - 60 saniye için karmaşıklık matrisi



Şekil 5.7 J48 algoritmasının sırasıyla 20 - 40 - 60 saniye için karmaşıklık matrisi



Şekil 5.8 Random Forest algoritmasının sırasıyla 20 - 40 - 60 saniye için karmaşıklık matrisi

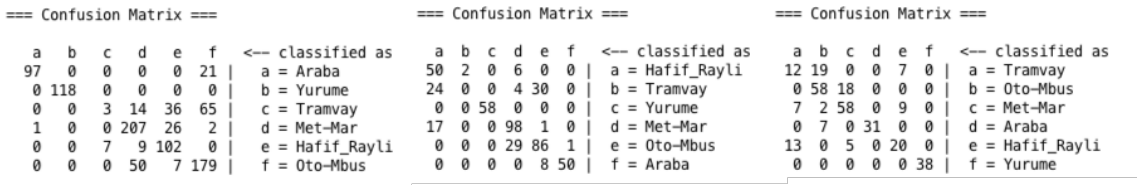
Karmaşıklık matrisleri incelendiğinde Metro - Marmaray ve Otobüs - Metrobüs ikililerinin birbiri ile çok fazla karıştırıldığı görülmektedir. Bu ikililer tek bir sınıf altında birleştirildiğinde doğruluk değerleri Tablo 5.3'deki gibi olmaktadır.

Tablo 5.3'de belirtilen doğruluk değerlerinin karmaşıklık matrisleri; Naive Bayes algoritması için Şekil 5.9'de, KNN algoritması için Şekil 5.10'da, J48 algoritması için Şekil 5.11'de ve son olarak Random Forest algortması için Şekil 5.12'de gösterilmiştir.

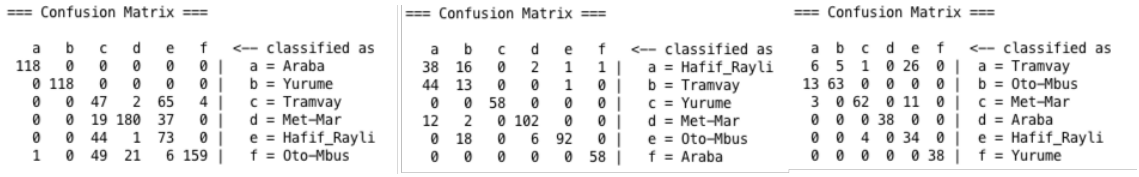
Metro-Marmaray ikilisi Met-Mar şeklinde, Otobüs-Metrobüs ikilisi ise Oto-Mbus şeklinde temsil edilmiştir.

Tablo 5.3 Sınıfların birleştirilmesiyle elde edilen doğruluk değerleri

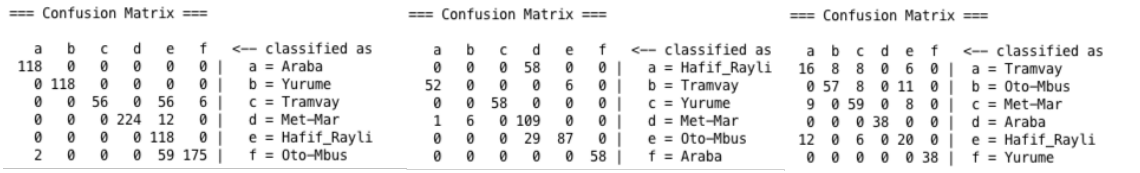
	20 Saniye	40 Saniye	60 Saniye
Naive Bayes	% 74.7881	% 73.7069	% 78.3826
KNN	% 73.6229	% 77.8017	% 79.2763
J48	% 85.6992	% 67.2414	% 75
Random Forest	% 85.5127	% 85.3448	% 81.9079



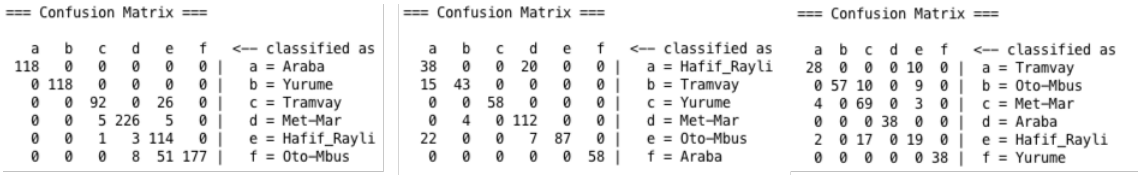
Şekil 5.9 Naive Bayes algoritmasının sırasıyla 20 - 40 - 60 saniye için karmaşıklık matrisi



Şekil 5.10 KNN algoritmasının sırasıyla 20 - 40 - 60 saniye için karmaşıklık matrisi



Şekil 5.11 J48 algoritmasının sırasıyla 20 - 40 - 60 saniye için karmaşıklık matrisi



Şekil 5.12 Random Forest algoritmasının sırasıyla 20 - 40 - 60 saniye için karmaşıklık matrisi

Sonuçlar incelendiğinde en uygun model J48 algoritması tarafından oluşturulan ve 20 saniye pencere aralığı içeren makine öğrenmesi modeli olduğu görülmüştür. Sistemin çıkarım mekanizması olarak Şekil 5.13'deki model tercih edilmiştir.

```
AX-SqMean <= 0.000546
|   AZ-StdDev <= 0.013789: Met-Mar (203.0)
|   AZ-StdDev > 0.013789
|   |   AY-InqRan <= 0.01386
|   |   |   AX-SqMean <= 0.000227: Met-Mar (33.0)
|   |   |   AX-SqMean > 0.000227: Tramvay (118.0)
|   |   AY-InqRan > 0.01386: Hafif_Rayli (118.0)
AX-SqMean > 0.000546
|   AX-SqMean <= 0.072719
|   |   GZ-SqMean <= 0.008267
|   |   |   AX-Kurtosis <= 4.910941: Araba (26.0)
|   |   |   AX-Kurtosis > 4.910941: Oto-Mbus (236.0)
|   |   GZ-SqMean > 0.008267: Araba (92.0)
|   AX-SqMean > 0.072719: Yurume (118.0)
```

Şekil 5.13 Sistemin J48 algoritması tarafından oluşturulan makine öğrenmesi modeli

Kullanıcının akıllı telefonuna ait 20 saniyelik sensör verisi Şekil 5.13'de belirtilen makine öğrenmesi modelinden geçirilmeden önce sözde kodu Şekil 5.14'de görülen gürültü önleyici bir filtreden geçirilerek aykırı verilerin temizlenmesi sağlanır. Sisteme ait gürültü önleyici filtredeki alfa değeri 0.85 olarak belirlenmiştir.

```
for i from 1 to n

y[i] := y[i-1] + α * (x[i] - y[i-1])
```

Şekil 5.14 Alçak geçirgen filtre sözde kodu

Her 20 saniye de bir toplanan filtrelenmiş sensör verileri makine öğrenmesi modelinden geçirilir. Elde edilen zaman ve ulaşım türü bilgileri geçici olarak saklanır. Veri toplama işlemi durdurulduğunda depolanan veriler incelenerek farklı zamanlardaki ardışık iki yürüme işlemi arasında yer alan taşıt bilgilerinden en fazla olan bulunur; aradaki tüm taşıt bilgileri bulunan taşıt bilgisi ile etiketlenir. Elde edilen veriler işlenerek herbir zaman aralığında hangi vasıta ile ne kadar süre seyehat edildiği veritabanında saklanır.

5.2 Veritabanı Tasarımı

Projede toplanan verilerin saklanması için veritabanı yapısı kullanılmıştır. Kullanılan veritabanı ise CoreData yapısıdır. Veritabanı iki tablodan oluşmaktadır.

Tablo 5.4 Sensor verileri tablosu

AX (Double)	AY (Double)	AZ (Double)	GX (Double)	GY (Double)	GZ (Double)	Mode (String)

Tablo 5.4'de Ax-Ay-Az sütunları ivmeölçerin o anki x-y-z koordinatlarındaki değerlerini tutmaktadır. Gx-Gy-Gz sütunları jiroskopun o anki x-y-z koordinatlarındaki

değerlerini tutmaktadır. Mode sütununda ise verilerin kaydının yapıldığı ulaşım türü bilgisi tutulmaktadır.

Tablo 5.5 Ulaşım türü tablosu

Tarih (Date)	Saat (Time)	Mode (String)

Tablo 5.5'de uygulama tarafından tahmin edilen ulaşım türü aşağıdaki tabloya kayıt edilir. Tarih ve saat sütunlarında ulaşım türünün belirlendiği tarih ve saat bilgisi tutulur. Mode sütununda kullanıcının seyehat ettiği vasıta ismi yazmaktadır.

Tablo 5.6 Ulaşım türü - süre tablosu

Süre (Double)	Mode (String)

Tablo 5.6'da görülen veritabanı tablosunda belirli zaman aralıklarında hangi ulaşım türü ile ne kadar süre seyehat edildiği bilgisi tutulmaktadır. Süre sütununda ulaşım türünün süresi, mode sütununda ise ulaşım türü bilgisi bulunmaktadır.

5.3 Girdi Çıktı Tasarımı

Sistemde girdi bilgisi olarak sensör verileri alınmaktadır. Kullanıcı bireysel olarak herhangi bir bilgi girmemektedir.

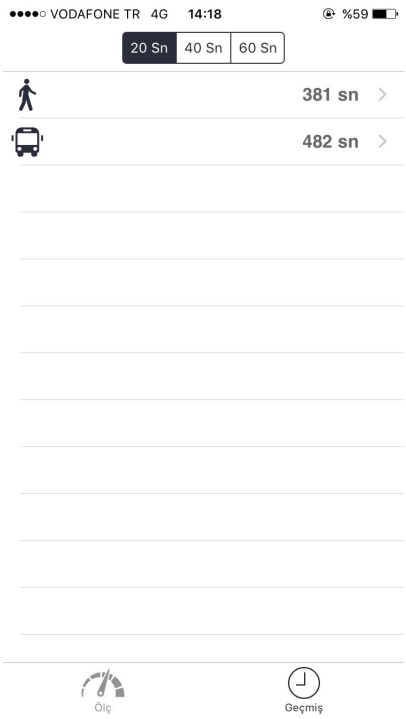
Sistemde çıktı bilgisi olarak kullanıcının o anki ulaşım türü verilmektedir. Çıktı araba, otobüs, metrobüs, metro, marmaray, hafif raylı ve tramvay vasıtalarından herhangi bir tanesidir.

6
Uygulama

Proje kapsamında geliştirilen mobil uygulamaya ait ekran görüntüleri bu bölüm içerisinde tanıtılmaktadır.

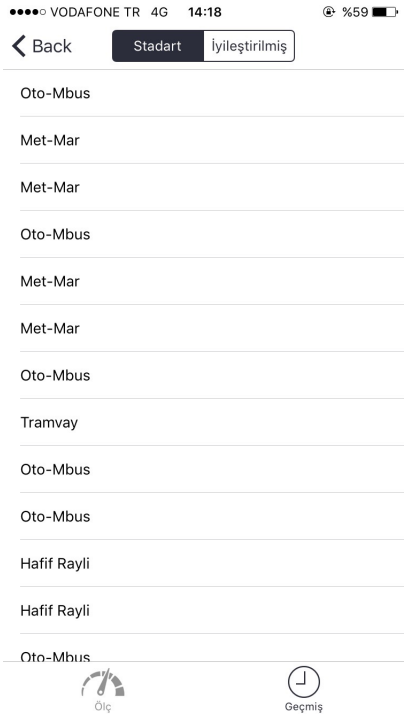
Şekil 6.1 Gezgin Veri Toplama Ekranı - İlk Açılma

Uygulama ilk açıldığında Şekil 6.1'deki ekran kullanıcıyı karşılamaktadır. Kullanıcı ekranın üzerinde bulunan butonlar aracılığı ile pencere aralığını belirleyebilmektedir. Play butonu belirlenen pencere boyutunda sensör verisi toplamaktadır.



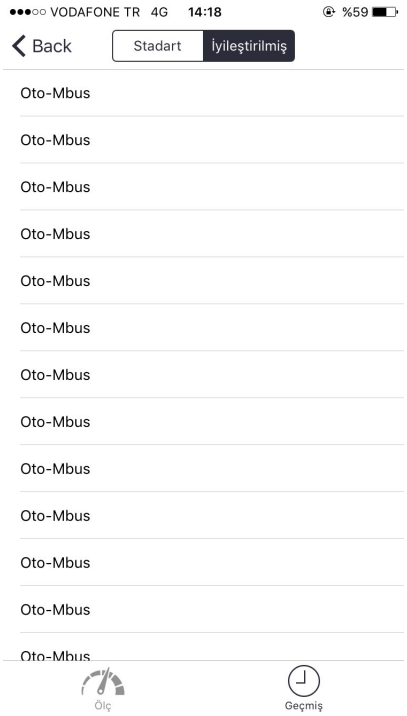
Şekil 6.2 Gezgin Geçmiş Ekranı

Şekil 6.2'de ekranın üst kısmında bulunan pencere boyutu seçilerek, seçilen pencere boyutuna ait kullanıcının ulaşım türü geçmişi listelenmektedir.



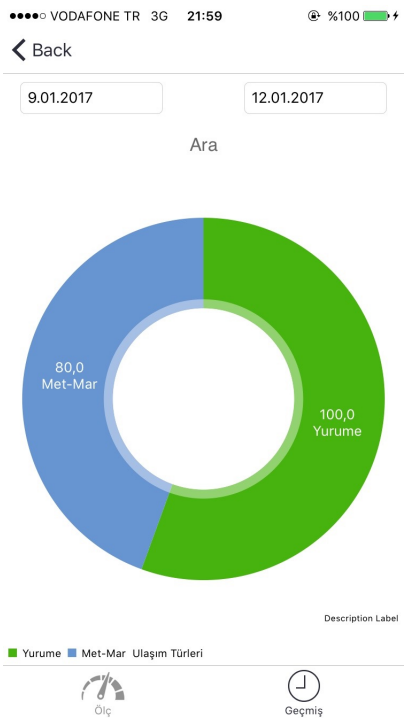
Şekil 6.3 Gezgin Geçmiş Detay Ekranı - İyileştirme Uygulanmamış Sonuçlar

Şekil 6.3'de ise her bir ulaşım türü hesaplanırken makine öğrenmesi modelinin oluşturduğu etiketler listelenmektedir.



Şekil 6.4 Gezgin Geçmiş Detay Ekranı - İyileştirme Uygulanmış Sonuçlar

Şekil 6.3'de listelenen etiketler iyileştirme algoritmasından geçirilince elde edilen yeni sonuçlar Şekil 6.4'deki ekranda listelenmektedir.



Şekil 6.5 Gezgin Grafik Ekranı

Şekil 6.5'de ise kullanıcı tarafından girilen tarihler arsında hangi ulaşım türü ile ne kadar süre seyahat edildiği grafik şeklinde sunulmaktadır.

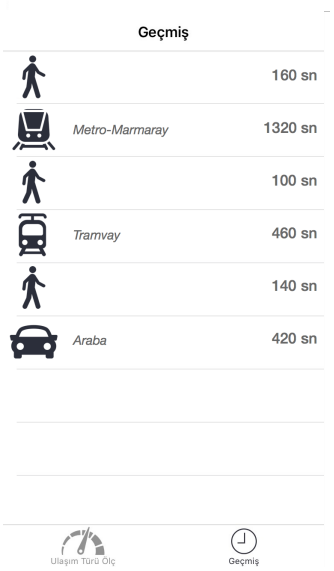
7
Deneysel Sonuçlar

Sistemin 2 farklı güzergah üzerinde test edilmiştir. İlk güzergah Tablo 7.1'deki gibidir.

Tablo 7.1 1. Test güzergahı

Başlangıç	**Ulaşım Türü**	**Bitiş**
Yenikapı Marmaray İstasyonu	Yürüme	Yenikapı Metro İstasyonu
Yenikapı Metro İstasyonu	Haifa Raylı	Zeytinburnu Metro İstasyonu
Zeytinburnu Metro İstasyonu	Yürüme	Zeytinburnu Tramvay İstasyonu
Zeytinburnu Tramvay İstasyonu	Tramvay	Cevizlibağ Tramvay İstasyonu
Zeytinburnu Tramvay İstasyonu	Yürüme	Cevizlibağ Taksi Durağı
Cevizlibağ Taksi Durağı	Araba	Esenler Metrosu

Tablo 7.1'de belirtilen 1. güzergaha ait Şekil 7.1'deki uygulama ekran çıktısı incelendiğinde sistem hafif raylı sınıfını metro-marmaray sınıfı ile karıştırmaktadır. Yürüme, tramvay ve araba sınıfları başarılı bir şekilde ayırt edilmektedir. Kullanıcı eğer çok yavaş yürürse sistem bu sınıfı ayırt edememektedir ve metro-marmaray sınıfı ile karıştırmaktadır.

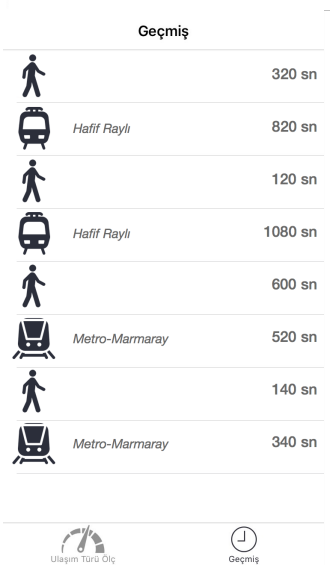


Şekil 7.1 1. güzergaha ait uygulama ekran çıktısı

Tablo 7.2 2. Test güzergahı

Başlangıç	**Ulaşım Türü**	**Bitiş**
FSM Öğrenci Yurdu	Yürüme	YTÜ Davutpaşa Otobüs Durağı
YTÜ Davutpaşa Otobüs Durağı	Otobüs	Cevizlibağ Otobüs Durağı
CevizlibağOtobüs Durağı	Yürüme	Cevizlibağ Metrobüs Durağı
Cevizlibağ Metrobüs Durağı	Metrobüs	Mecidiyeköy Metrobüs Durağı
Mecidiyeköy Metrobüs Durağı	Yürüme	Mecidiyeköy Metro İstasyonu
Mecidiyeköy Metro İstasyonu	Metro	Yenikapı Metro İstasyonu
Yenikapı Metro İstasyonu	Yürüme	Yenikapı Marmaray İstasyonu
Marmaray İstasyonu	Marmaray	Üsküdar Marmaray İstasyonu

Tablo 7.2'de belirtilen 2. güzergaha ait Şekil 7.2'deki uygulama ekran çıktısı incelendiğinde sistem otobüs-metrobüs sınıfını hafif raylı sınıfı ile karıştırmaktadır. Yürüme, metro-marmaray sınıfları başarılı bir şekilde ayırt edilmektedir. Yürüme sınıfı en düşük ivmelenmeye sahip olduğu için sistem bu sınıfı diğer sınıflardan kolayca ayırt edebilmektedir. Hafif raylı vasıtasının ortalama hızı metro ve marmaray vasıtalarının ortalama hızına yakın olması sebebiyle sistem hafif raylı ile metro-marmaray sınıflarını birbirine karıştırmaktadır. Tramvay sınıfının hızı diğer raylı vasıtalardan daha az olması sebebiyle sistem tarafından bu sınıf kolayca ayırt edilebilmektedir.



Şekil 7.2 2. güzergaha ait uygulama ekran çıktısı

Genel olarak sistem 6 sınıf (araba, yürüme, metro-marmaray, tramvay, hafif raylı, otobüs-metrobüs) içerisinden araba, yürüme, metro-marmaray ve tramvay sınıflarını başarılı bir şekilde ayırt edebilmektedir.

8
Sonuç

Yolculuk esnasında kullanıcı ulaşım türü tespiti projesinde, akıllı telefonlarda bulunan ivmeölçer ve jiroskop sensörleri aracılığı ile kullanıcının anlık ulaşım türünün belirlenmesi amaçlanmaktadır. Tespit edilmesi amaçlanan ulaşım türleri araba, metro, marmaray, tramvay, hafif raylı, metrobüs ve otobüsdür. Belirtilen sınıfların sistem tarafından tespit edilebilmesi için makine öğrenmesi teknikleri kullanılmıştır. Herbir ulaşım türüne ait 20 dakikalık eğitim ve 20 dakikalık test verisi toplanmıştır. Toplanan veriler özellik çıkarımı işleminden geçirilmiştir. Özellik çıkarımı işlemi 20 saniye, 40 saniye ve 60 saniyelik pencere aralıkları için gerçekleştirilmiştir. Elde edilen yeni veri setleri makine öğrenmesi algoritmalarından geçirilmiştir. En iyi sonucu 20 saniye pencere aralığına sahip veri seti için %65 doğruluk oranı ile J48 algoritması vermiştir. Metro - marmaray sınıfı ile otobüs - metrobüs sınıfları birleştirilerek yeni veri setleri oluşturulmuştur. Elde edilen yeni veri setleri test edildiğinde en iyi sonucu 20 saniye pencere aralığına sahip veri seti için %85 doğruluk oranı ile J48 algoritması vermiştir. J48 algoritmasının oluşturduğu model uygulamaya eklenmiştir. Bu modele ek olarak sistemde elde edilen sonuçların iyileştirilmesi için yeni bir algoritma geliştirilmiştir. Algoritma şu şekildedir: ardışık iki yürüme işlemi arasında yer alan taşıt bilgilerinden en fazla olan bulunur; aradaki tüm taşıt bilgileri bulunan taşıt bilgisi ile etiketlenir. Bu algoritma sayesinde doğruluk oranı %87.5 seviyesine çıkmıştır.

Yapılan deneyler sonucunda tramvay, araba, yürüme, metro-marmaray sınıfları başarılı bir şekilde tespit edilebilmektedir. Otobüs - metrobüs ve hafif raylı sınıflarının tespiti başarılı bir şekilde gerçekleştirilememiştir. Gerçekleştirilen çalışma sonunda başlangıçta belirlenen hedefe büyük ölçüde ulaşılmıştır. Başlangıçta hedef, araba, metro, marmaray, tramvay, hafif raylı, metrobüs ve otobüs olmak üzere 7 farklı ulaşım türünün tespit edilmesiydi. Çalışma sonucunda araba, metro-marmaray, tramvay, hafif raylı ve otobüs-metrobüs olmak üzere 5 farklı ulaşım türü sınıflandırılabilmiştir.

Gelecek çalışmalarda makine öğrenmesinin doğruluk oranının artırılabilmesi adına iki yöntem önerilebilir. Bir tanesi eğitim veri setindeki veri miktarının artırılması. Makine öğrenmesinde ne kadar fazla veri var ise o kadar iyi bir doğruluk oranına sahip olunur. Bu bağlamda farklı kullanıcılar tarafından saatlerce toplanan sensör verileri gelecek çalışmalarda olumlu bir ilerleme sağlayacaktır.
Kullanılan sensör sayısının da artırılması doğruluk oranını artıracaktır. Günümüz akıllı telefonlarında ivmeölçer ve jiroskop sensörlerinin yanında magnetometre de bulunmaktadır. Bu 3 sensörden alınan verilerde doğruluk oranını artıracaktır.

Referanslar

[1] S. Reddy, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Determining transportation mode on mobile phones," in *2008 12th IEEE International Symposium on Wearable Computers*, IEEE, 2008, pp. 25–28.

[2] S. Wang, C. Chen, and J. Ma, "Accelerometer based transportation mode recognition on mobile phones.," *APWCS*, vol. 2010, pp. 44–46, 2010.

[3] H. Sökün, H. Kalkan, and B. Cetişli, "Classification of physical activities using accelerometer signals," in *2012 20th Signal Processing and Communications Applications Conference (SIU)*, IEEE, 2012, pp. 1–4.

[4] T. Feng and H. J. Timmermans, "Transportation mode recognition using gps and accelerometer data," *Transportation Research Part C: Emerging Technologies*, vol. 37, pp. 118–130, 2013.

[5] L. Stenneth, O. Wolfson, P. S. Yu, and B. Xu, "Transportation mode detection using mobile phones and gıs information," in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ACM, 2011, pp. 54–63.

Özgeçmiş

1. ÜYENİN KİŞİSEL BİLGİLERİ

İsim-Soyisim: Yunus Emre DEMİRBULUT
Doğum Tarihi ve Yeri: 28.05.1996, Trabzon
E-mail: y.emre.demirbulut@gmail.com
Telefon: 0549 326 20 26
Staj Tecrübeleri: SPEXCO Bilişim Yaz. San.ve Tic. Ltd. Şti.

Proje Sistem Bilgileri

Sistem ve Yazılım: iOS, Swift
Gerekli RAM: 1GB
Gerekli Disk: 30 MB