TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ARAÇ SÜRÜŞ VERİLERİNDEN MAKİNE ÖĞRENMESİ TEKNİKLERİNİ KULLANARAK SÜRÜCÜ SINIFLANDIRMA

YÜKSEK LİSANS TEZİ

Batuhan KARATAŞ

Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Osman ABUL

NİSAN 2018

Fen Bilimleri Enstitüsü Onayı

.....
Prof. Dr. Osman EROĞUL
Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onaylarım.

.....
Prof. Dr. Oğuz ERGİN
Anabilimdalı Başkanı

TOBB ETÜ, Fen Bilimleri Enstitüsü'nün 151111021 numaralı Yüksek Lisans öğrencisi **Batuhan KARATAŞ**'ın ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı **"ARAÇ SÜRÜŞ VERİLERİNDEN MAKİNE ÖĞRENMESİ TEKNİKLERİNİ KULLANARAK SÜRÜCÜ SINIFLANDIRMA"** başlıklı tezi **03.04.2018** tarihinde aşağıda imzaları olan jüri tarafından kabul edilmiştir.

Tez Danışmanı:	**Doç. Dr. Osman ABUL** TOBB Ekonomi ve Teknoloji Üniversitesi	.....
Jüri Üyeleri:	**Doç. Dr. Hacer KARACAN (BAŞKAN)** Gazi Üniversitesi	.....
	Dr. Öğr. Üyesi Mehmet TAN TOBB Ekonomi ve Teknoloji Üniversitesi	.....

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, alıntı yapılan kaynaklara eksiksiz atıf yapıldığını, referansların tam olarak belirtildiğini ve ayrıca bu tezin TOBB ETÜ Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırlandığını bildiririm.

Batuhan KARATAŞ

ÖZET

Yüksek Lisans Tezi

ARAÇ SÜRÜŞ VERİLERİNDEN MAKİNE ÖĞRENMESİ TEKNİKLERİNİ KULLANARAK SÜRÜCÜ SINIFLANDIRMA

Batuhan KARATAŞ

TOBB Ekonomi ve Teknoloji Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Osman ABUL

Tarih: NİSAN 2018

Araç donanım teknolojisindeki gelişmeler büyük ölçekli araç sürüş verilerinin toplanmasına olanak sağlamıştır. Bu veriler özellikle kentsel alan trafik yönetimi ve araç sürüş destek sistemi uygulamaları için önemli bir kaynak teşkil etmektedir. Bu çalışmada, bu verilerin sürücü ile ilgili çıkarım yapabilme yeteneği ile ilgilenilmiştir. Veri kaynağı olarak Uyanık veri kümesi [1] CAN(Controller Area Network) verileri kullanılmıştır. Sürücü kümeleme, sürücü cinsiyet sınıflandırma ve sürücü tanıma ile ilgili deneyler gerçekleştirilmiştir. Sürücü kümeleme deneylerinde Dynamic Time Warping ve kendi geliştirdiğimiz Dynamic Distance Warping veri dönüşüm metodları uygulanarak farklı mesafe metriklerine göre hiyerarşik sürücü kümeleme işlemi gerçekleştirilmiştir. Bu işlemin sonucunda tutarlı sürücü gruplamaları elde edilmiştir. Sürücü cinsiyet sınıflandırma deneylerinde veri örnekleme, öznitelik çıkarımı, öznitelik eleme ve ayrıştırma veri ön işleme metodları uygulanarak 0.97 doğruluk oranına ulaşılmıştır. Sürücü tanıma deneylerinde kendi geliştirdiğimiz bir örüntü parçalama tekniği ve öznitelik çıkarımı veri ön işleme metodları uygulanarak 105 adet sürücü arasından 0.1

doğruluk oranında sürücü sınıflandırma işlemi gerçekleştirilmiştir. Tüm bu deneyler ile ortaya çıkan çıkarımlar neticesinde literatürde yeni bir tartışma konusu ortaya çıkmaktadır; Sürüş verisi hassas kişisel veri kapsamında değerlendirilmeli midir?

Anahtar Kelimeler: Araç sürüş verileri, Makine öğrenmesi, Sürücü tanıma, Sürücü cinsiyet sınıflandırma, Uyanık veri kümesi.

ABSTRACT

Master of Science

DRIVER CLASSIFICATION WITH USING MACHINE LEARNING METHODS ON VEHICLE DRIVING DATA

Batuhan KARATAŞ

TOBB University of Economics and Technology
Institute of Natural and Applied Sciences
Department of Computer Engineering

Supervisor: Assoc. Prof. Dr. Osman ABUL

Date: APRIL 2018

The advances in vehicle equipment technology enabled us collecting large-scale vehicle driving data. This data is an important resource for urban area traffic management and vehicle driving support system applications. In this study, we are interested making inferences ability of these data about the driver. Uyanık data set [1] CAN(Controller Area Network) bus data are used as the data source. Experiments are carried out on driver gender classification, driver identification, and driver clustering. In the driver clustering experiments, hierarchical driver clustering is performed with using the Dynamic Distance Warping developed by us and Dynamic Distance Time data conversion methods according to different distance metrics. As a result, consistent driver groupings are achieved. In driver gender classification experiments, gender classification is performed with applying data sampling, feature extraction, feature elimination and discretization data preprocessing methods. Best classifiers reached up to 0.97 accuracy rate. In driver identification experiments, driver classification is carried out with applying driver pattern splitting technique developed by us and feature extraction data

preprocessing methods and driver identification performance reached 0.1 accuracy rate among the 105 drivers. All these experiment results open up a new thread of discussion: whether the driving data should be treated as a sensitive personal feature?

Keywords: Vehicle driving data, Machine learning, Driver identification, Driver gender classification, Uyanik dataset.

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren hocam Doç. Dr. Osman ABUL'a, yüksek lisans boyunca her zaman yanımda olan ve desteğini hiç esirgemeyen sevgili annem Ayfer KARATAŞ'a ve babam Salih KARATAŞ'a, değerli fikirleriyle bu çalışmaya katkıda bulunan mesai arkadaşlarım Sarp MERTOL, Yusuf Alper BİLGİN ve İbrahim ARSLAN'a, yüksek lisans boyunca araştırma bursu ile eğitim aldığım TOBB ETÜ'ye ve TOBB ETÜ Bilgisayar Mühendisliği Bölümü öğretim üyelerine çok teşekkür ederim.

İÇİNDEKİLER

ŞEKİL LİSTESİ

ÇİZELGE LİSTESİ

KISALTMALAR

CAN : Controller Area Network
GSP : Güvenli Sürüş Projesi
GPS : The Global Positioning System
GMM : Gaussian Mixture Model
HMM : Hidden Markov Model
SMOTE : Synthetic Minority Oversampling Technique
SMO : Sequential Minimal Optimization
RF : Random Forest
ABM1 : Ada Boost M1
MLP : Multilayer Perceptron
VP : Voted Perceptron
BN : Bayesian Network
SVM : Support Vector Machine
DAG : Directed Acylic Graph
TP : True Positive
FP : False Positive
PCA : Principal Component Analysis
EHK : En hızlı kadın sürücü
EYK : En yavaş kadın sürücü
EHE : En hızlı erkek sürücü
EYE : En yavaş erkek sürücü
DTW : Dynamic Time Warping
DDW : Dynamic Distance Warping
ECG : Electro Cardiogram

1. GİRİŞ

Endüstri 4.0 ile birlikte birçok farklı endüstri sektöründe otomasyon oranı yüksek ve kullanıcı odaklı çözümler üreten siber fiziksel sistemler ortaya çıkmaya başlamıştır. Bu sektörlerden biri de otomotiv sektörüdür. Günümüzde araç üreticileri, sürücü kaynaklı trafik kazalarını azaltmak için araç aktif güvenlik sistemlerini ve sürüş konforunu geliştirmek adına sürücü odaklı birçok uygulamayı araçlarında kullanmaktadır. Bu uygulamalarda ise yapay zekanın bir alt dalı ve endüstri 4.0'ın temel bileşenlerden biri haline gelen makine öğrenmesi kullanılmaktadır [2].

Günümüzde çoğu araç CAN hattı ile donatılmıştır. Bu hat üzerinden araçta bulunan bazı sensörlerin verileri sürekli bir akış içerisinde geçer. Bu hat araç hızı, direksiyon, motor devri, gaz, fren, debriyaj ve benzeri birçok veri tipini içerebilmektedir [3]. Elbette, aynı koşullara sahip farklı sürücüler sürüş alışkanlığı ve psikolojik durum gibi kişisel nedenlerden ötürü farklı sensör veri değerleri üretebilir. Potansiyel olarak çok kullanışlı olmasına rağmen, veri akışı nadiren üçüncü parti uygulamalar tarafından kullanılır. (örn. kontrollü deney amaçlı projeler)

Uyanık veri kümesi, Sabancı Üniversitesi'nde yürütülen GSP(Güvenli Sürüş Projesi) kapsamında oluşturulmuştur [1]. GSP'nin veri toplama işlemi, Uyanık adlı özel donanımlı bir araç ve gönüllü sürücüler ile gerçekleştirilmiştir. Sürücülerden İstanbul'da belirlenmiş ve gerçek trafikte 25 km'lik bir rotada sürüş yapmaları istenmiştir ve deneyin bir parçası olarak sürücülerin dikkatlerini dağıtmak amacı ile telefon görüşmeleri yapılmıştır. Uyanık aracında gerçek zamanlı CAN hattı verisine ek olarak; GPS alıcı, lazer mesafe ölçer ve sürücü araç içi video kamerası gibi diğer araç sensör kaynaklarından alınan veriler her sürücü için eş zamanlı olarak kaydedilir. Araştırmamızda 17 kadın ve 88 erkek sürücüden gelen verilerden sadece CAN hattı verileri kullanılmıştır.

GSP'nin üç ana hedefi vardır: sürücü tanıma, kazaya sebebiyet verebilecek sürücü davranışlarının tespiti ve rota-manevra tespiti. Araştırmamız bu kapsamda düşünüldüğünde; temel amacımız CAN verilerini kullanarak sürücü tanımayı gerçekleştirmektir. Bunun yanında kritik olabilecek sürücü özelliklerinin çıkarımı ve sürüş davranışlarına göre sürücüleri gruplandırma konuları da araştırmamızın diğer amaçlarını oluşturmaktadır. Fakat Uyanık veri kümesinde cinsiyet haricinde eğitim seviyesi ve yaş grubu gibi herhangi bir sürücü özelliği net bir şekilde belirtilmemiştir. Bu yüzden ilgili çalışmamızı sadece bu sürücü özelliği üzerinde sınıflandırma yaparak gerçekleştirdik. Tüm bu işlemler gerçekleştirilirken makine öğrenme teknikleri kullanılmıştır. Bir başka hedefimiz ise en yüksek ve güvenilir doğruluk oranını sağlayan teknikleri ve CAN veri tipi kombinasyonlarını tespit ederek literatüre kazandırmaktır.

Yüksek doğruluk oranlarına ulaşıldığında bu çalışmanın; (i) CAN verilerinden sürücü ile ilgili çıkarımların yapılabildiğini ispatlayarak bu verilerin hukuki boyutta hassas kişisel veri olarak değerlendirilmesini sağladığına, (ii) gelecekte kullanımının giderek artacağı öngörülen kişiye özel sürüş çözümleri konseptine bir teknolojik alt yapı olarak fayda sağlayacağına inanmaktayız. Böylece araç üreticileri bazı araç parametrelerini kullanıcı özelliklerine göre özel olarak uyarlar ve bu tarz verilerin paylaşımı da gizlilik koruma teknikleri ile yapılır.

1.1 Literatür Araştırması

Güvenli Sürüş Projesi [1], uluslararası bir projenin parçasıdır. Bu uluslararası proje kapsamında, ABD ve Japonya'da da bir araç ile gerçek trafikte sürüş verilerini toplama ve bu verileri işleyerek çıkarımlarda bulunma temelinde projeler gerçekleştirilmiştir. Dolayısıyla tüm bu projeler amaç olarak birbirleriyle benzeşmektedir. Bu amaçlardan bir tanesini oluşturan sürücü tanıma kapsamında literatürde birçok çalışma gerçekleştirilmiştir. Çalışma [4], gaz ve fren pedalı durum sinyallerini kullanarak sürücü çıkarımını gerçekleştirmek amacıyla Nagoya Üniversitesi ile Toyota arasında yapılan müşterek bir araştırmadır. Bu çalışmada her sürücünün sürüş sinyallerinin spektral analizinin yapılması ile elde edilen Cepstral öznitelikleri GMM ile modellenmiştir. Deneylerde hem gerçek araç hem de simülatör uygulaması sürüş sinyalleri uygulanmıştır. Gaz ve fren verileri ayrı olarak ve birleştirerek deneyler gerçekleştirilmiştir. 12 adet sürücünün sürüş simülatörü verileri üzerinden sürücü tanıma doğruluk oranı %89,6, 16 adet sürücünün gerçek araçtaki sürüş verileri üzerinden sürücü tanıma doğruluk oranı ise %76,8 olarak gerçekleşmiştir. Spektral analiz yapılmadan ham veriler üzerinden bu işlem gerçekleştirildiğinde sırasıyla %61 ve %51 doğruluk oranları elde edilmiştir. Benzer şekilde, UT-Drive projesinde [5] dokuz adet sürücünün CAN hattı verilerden direksiyon açısı, fren pedal durumu, gaz pedal durumu ve araç hızı kullanılarak sürücü tanıma üzerine bir çalışma gerçekleştirilmiştir. HMM ve GMM modellerinin kullanıldığı araştırmada sürücü tanımada %25 doğruluk oranı elde edilmiştir. Benzer bir başka çalışmada [6], sürücü davranışsal sürüş sinyalleri olan gaz-fren pedal ve takip ettiği araca olan uzaklık bilgilerini kullanarak sürücü tanıma sistemi geliştirilmiştir. Bu işlem için GMM kullanılmıştır. Bu özniteliklerin farklı birleşim kombinasyonları kullanılarak deneyler gerçekleştirilmiştir. 23 sürücü üzerinde yapılan deneylerde %57,39 doğruluk oranı, üç sürücü kullanılarak yapılan deneyde %85,21 doğruluk oranı elde edilmiştir.

Başka bir çalışmada, davranışsal sürüş sinyalleri olarak gaz ve fren pedal basınç değerleri kullanılarak sürücü biyometrik tanımlaması gerçekleştirilmiştir [7]. Deneylerde bu özniteliklerden statik ve dinamik (zamana bağlı) yapıda olanlar denenmiştir ve GMM kullanılmıştır. Dinamik özniteliklerin daha iyi sonuç verdiği tespit edilmiştir. Zheng ve

arkadaşları [8] sürüş performansını değerlendirerek sürücünün acemi olup olmadığını tahmin eden bir sistem ile ilgili ön çalışma gerçekleştirmişlerdir. Bu deneye iki adet 16 yaşında acemi sürücü katılmıştır. Bu sürücülere beş adet temel sürüş komutu sesli olarak verilmiştir. Sürücülerin bu komutlara verdiği manevra tepkileri CAN hattından tespit edilip karşılaştırarak sürüş performans değerlendirmesi yapılmıştır.

Uyanık veri kümesi kullanılarak farklı hız değerlerinde dakikadaki gaz-fren pedal geçiş sayısı analiz edilerek erkek ve kadın sürücülerin sürüş karakteristiği ile ilgili bir çalışma gerçekleştirilmiştir . Bu çalışmada kadınların gaz-fren pedal geçişlerinin erkeklere göre daha az olduğu saptanmıştır. Ayrıca Uyanık veri kümesini kullanan bu çalışmada [9], histogram tekniği kullanılarak CAN hattı üzerinden direksiyon açısı, araç hızı, gaz pedalı yüzdesi ve fren pedalı bilgilerinden sürücü profili çıkarılmaya çalışılmıştır. Deneylerde üç kadın üç erkek sürücü kullanılmıştır. Sonuçlar şu şekilde yorumlanmıştır; (i) direksiyon açı değişimi çok olan şoförler için şerit takip sistemi, (ii) araç hız değerleri normalin üstünde olan sürücüler için seyir kontrol sistemi veya hız sınırlayıcı, (iii) gaz pedalı yüzdesi ve fren kullanımı yüksek olanlar için yakıt tüketim uyarıcısı önerilmiştir. Ayrıca fren pedal kullanımı bilgisinin sürücünün sürüş uzmanlık seviyesi ile ters orantılı olduğu ve bu konu ile ilgili belirleyici rol oynayacağı belirtilmektedir. Aracın hızının ve gaz pedal kullanımı yüzdesinin sürücü cinsiyeti için belirleyici olabileceği de not edilmelidir.

CAN hattı verileri birer zaman serisidir. Bu veriler üzerinde kümeleme gibi işlemlerin uygulanabilmesi için verilerin uygun formlara dönüştürülmesi gerekmektedir. Bununla ilgili olarak ECG kalp atım zaman serisi verilerine, DTW yöntemini uygulayarak bu veriler içerisindeki anomalilerin kümeleme işlemi gerçekleştirilerek tespit edildiği bir çalışma gerçekleştirilmiştir [10]. Başka bir çalışmada ise XBox oyun konsolunun *Kinect* hareket algılama donanımının zaman serisi verilerine, HMM ve DTW uygulanarak el-kol hareketleri tespit edilmeye çalışılmıştır [11].

Bazı özel sürüş davranışları araç sürücüsünün tanınmasında ayırt edici olabilmektedir. Bir araştırmada, araç tek dönüş davranışı kullanılarak sürücü tanıma gerçekleştirilmiştir [12]. Veri kümesi; Ingolstadt-Almanya'da gerçek trafikte sürücülerin tek dönüşü esnasında tork, direksiyon açısı, direksiyon dönüş hızı, direksiyon ivmesi, motor devri ve gaz-fren pedal pozisyonu gibi 12 farklı sensör verisinin kaydedilmesi ile oluşturulmuştur. Bu zaman serisi verilerinden sürücü sınıflandırırken kendi geliştirdikleri bir sınıflandırıcıyı kullanmışlardır. Sınıflandırmalar, veri kümesinde en çok gerçekleştirilen 12 farklı tek dönüş tipi'nin (kırsal, şehir, otoyol vb.) her biri için gerçekleştirilmiştir. Belirtilen araç verileri kullanılarak 5 sürücünün 12 farklı dönüş tipinin her biri için yapılan sınıflandırmalarda ortalama %50,1 doğruluk oranı elde edilmiştir. Bir başka çalışmada, motor hızlanma ve yavaşlama verilerini kullanarak sürücü sınıflandırma gerçekleştirmektedir [13]. Bu çalışmayı diğerlerinden ayıran en önemli faktör deneyde

ki sürücülerin yaşının 70 ve üzeri olmasıdır. Veri kümesi 14 adet stabil sağlığa sahip sürücünün 1 yıl boyunca sürüş yapması sonucunda oluşturulmuştur. Bu verilere çok sınıflı LDA sınıflandırıcıları uygulanması sonucunda; hızlanma durumunda ortalama %34, yavaşlama durumunda ise %30 doğruluk oranına ulaşmıştır. Ayrıca yapılan analizler sonucunda, dönme ve duraksama gibi sürüş manevralarının sürücü tanıma için daha yüksek potansiyele sahip olduğu tespit edilmiştir. Bunların yanında, gerçek bir aracın CAN hattı verilerinden hangilerinin sürücü tanımada daha karakteristik özellikler oluşturduğuna dair bir çalışma yapılmıştır [14]. Analiz sonuçlarına göre duraklama ve dönme manevralarının hızlanmaya göre daha iyi performans gösterdiği gözlemlenmiştir. Ayrıca bu tip verileri tek başına kullanmak yerine birleştirerek kullanmanın sürücü ayrıştırmasında daha başarılı olduğu tespit edilmiştir.

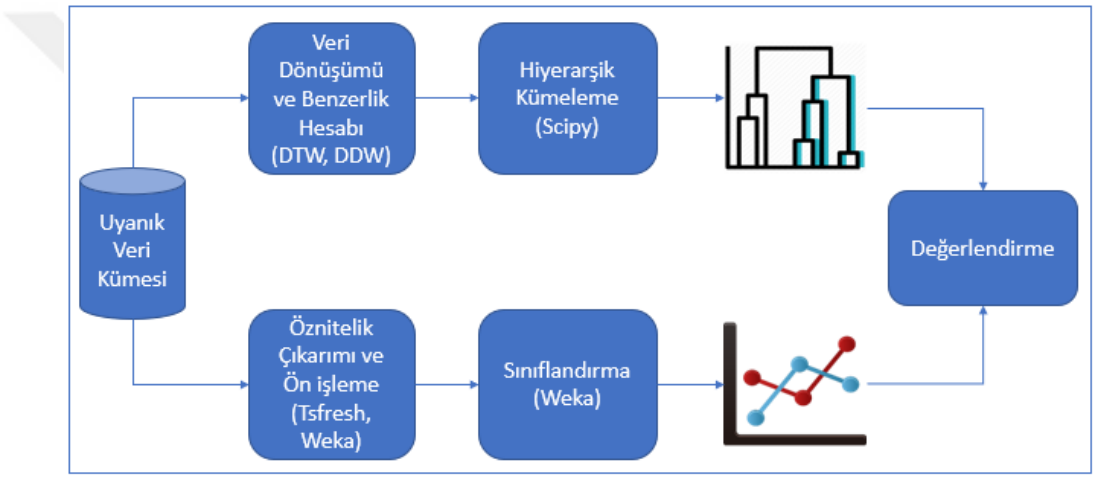
1.2 Tezin Organizasyonu

Bölüm 2'de, araştırmamızda kullandığımız sistemin metodolojisi ve Uyanık veri kümesinin ön analizi yer almaktadır. Bölüm 3'de, hiyerarşik kümeleme yöntemi ve bu yöntemle gerçekleştirilen sürücü kümeleme deneyleri anlatılmıştır. Bölüm sonunda, deney sonuçları analiz edilmiştir. Bölüm 4 ve 5'de sınıflandırma deneyleri gerçekleştirilmiştir. Bu deneylerin bir gereği olarak uygulanan öznitelik çıkarımı ve sınıflandırma algoritmalarına Bölüm 4'de yer verilmiştir. Bu genel kısımların haricinde bu bölümde sürücü cinsiyet sınıflandırma deneylerinin metodolojisi, veri ön işleme aşamaları ve deney sonuçlarının analizi yer almaktadır. Bölüm 5'de sürücü tanıma deneyleri gerçekleştirilmiştir. Bu deneylerin metodolojisi, veri ön işleme aşamaları ve deney sonuçlarının analizi anlatılmıştır. Ayrıca bu bölümde sürücü tanıma deneylerinin bir sonucu olarak; uygulanması konusunda görüş belirttiğimiz kişisel veri mahremiyeti konusuna da yer verilmiştir. Son bölümde ise araştırmamızın genel değerlendirmesi yapılmıştır.

2. ARAŞTIRMA BİLEŞENLERİ ANALİZİ VE METODOLOJİSİ

2.1 Araştırma Metodolojisi

Şekil 2.1, araştırmada kullandığımız metodolojileri aşamalar halinde göstermektedir. Üst diyagram akışı sürücü kümeleme deneyleri için kullandığımız metodolojiyi, alt diyagram akışı ise sürücü cinsiyet sınıflandırma ve sürücü tanıma deneylerinde kullandığımız metodolojiyi ifade etmektedir.



Şekil 2.1: Araştırma metodolojisi.

Üst diyagram akışında, ilk aşama veri dönüşümü ve benzerlik hesabıdır. Bu aşamada girdi olarak gelen ham zaman serisi verileri, veriler arası benzerlik hesabının gerçekleştirilebilmesi için veri dönüşüm işlemleri yapılarak uygun bir forma dönüştürülür. Ardından, farklı ölçüm metriklerine göre veri çiftleri arasında benzerlik hesabı yapılarak bir mesafe vektörü oluşturulur. Sonra, bu vektöre Python Scipy kütüphanesinin hiyerarşik sınıflandırma algoritmaları uygulanarak bir dendogram oluşturulur. En son aşamada ise uygulanan deney konfigürasyonunun başarımı değerlendirilir.

Alt diyagram akışında, ilk aşama öznitelik çıkarımı ve veri önişlemedir. Bu aşamada girdi olarak gelen ham zaman serisi verilerinden Tsfresh kütüphanesi kullanılarak belli sayılarda öznitelik çıkarımı yapılır ve veri üzerinde bazı ön işleme prosedürleri gerçekleştirilir. Bu işlemler öznitelik seçme, ayrıştırma ve sınıf veri sayısı dengeleme olarak

ortaya çıkabilmektedir. Sonra, ön işlemeye tabi tutulmuş veri kümesine sınıflandırma algoritmaları uygulanır. Veri ön işleme ve sınıflandırma işlemlerinde Weka programı kullanılmaktadır. En son aşamada ise uygulanan sınıflandırma algoritmalarının doğruluk oranları değerlendirilir.

2.2 Uyanık Veri Kümesi Ön Analizi

Bu bölümde Uyanık veri kümesinin bazı özelliklerini ayrıntılı olarak açıklayacağız ve sürücü kümeleme işleminden önce veri kümesi ön analizi gerçekleştireceğiz.

Uyanık veri kümesini Sabancı Üniversitesi VPA laboratuvarından aldık [15]. 88 erkek ve 17 kadın sürücünün sürüş verileri, İstanbul'da otoyol ve şehir içi yolları içeren 25 km'lik sabit bir rotada kaydedilmiştir. Verinin boyutu yaklaşık 750 gigabayttır. Bu verilerin içerisinde video kayıtları, ses kayıtları, lazer mesafe ölçer kayıtları, GPS kayıtları, jiroskop kayıtları ve CAN hattı kayıtları bulunmaktadır. Her sürücü için bu kayıt alanları mevcuttur. Sürücüler sisteme, kaydın yapıldığı şehir kodu, cinsiyetleri ve sürücü sistem kodları girilerek kayıt edilmiştir. Dolayısıyla, sürücüler ile ilgili bilinen tek kişisel özellik cinsiyetleridir.

Araştırmamızda bu veri kümesinden sadece CAN hattı verilerini kullandık. Araçlar için CAN hattı kritik bir görev üstlenmektedir. Araç bileşenleri bu hat üzerinden kendi aralarındaki iletişimi sağlar. Aynı zamanda sürücüyü araç ile ilgili bilgilendirmek üzere giden tüm sinyaller de bu hat üzerinden geçer. Uyanık aracında bu hat üzerinden gelen 19 farklı tipte veri kayıt altına alınmıştır. Bu kayıtlar, her sürücünün CAN hattı sinyali sisteme ulaştıkça zamana bağlı olarak kaydedilmiştir. Yani, veri kümesi bir çeşit çok boyutlu zaman serisi verisidir. CAN hattı ölçümleri hakkında ayrıntılı bilgi Çizelge 2.1'de verilmektedir.

ADC1 ve ADC2 öznitelikleri birçok sürücü için veri kümesinde mevcut değildir. Bu sebepten ötürü araştırmaya dahil edilmemişlerdir. Buna ek olarak, WSFR, WSFL, WSRR ve WSRL değerleri VS değeri ile yüksek oranda korelasyon göstermektedir. Bu yüzden artık oldukları düşünülerek çalışmaya dahil edilmemişlerdir. 25 kilometrelik rotada geri vites kullanılmadığı için deneyde herhangi bir ayırt edici özelliği olmayan RG verileri de araştırmaya dahil edilmemiştir. Sonuç olarak, araştırmamızda 10 adet CAN hattı verisi kullanılmıştır. Bunlar SWA, SWRS, VS, PGP, ERPM, NS, YR, CS, BS ve C'dir.

Zaman serilerinin nasıl göründüğünü göstermek amacıyla rastgele seçilen ve kod adları Erkek-2003, Kadın-1007 olan sürücüler için sırasıyla Şekil 2.2'de VS zaman grafikleri gösterilmiştir. Bu şekilde, Erkek-2003 sürücüsünün parkuru tamamlama süresi yaklaşık 36 dk'dır. Maksimum ulaştığı hız 122 km/s'dir. Kadın-1007 sürücüsünün par-

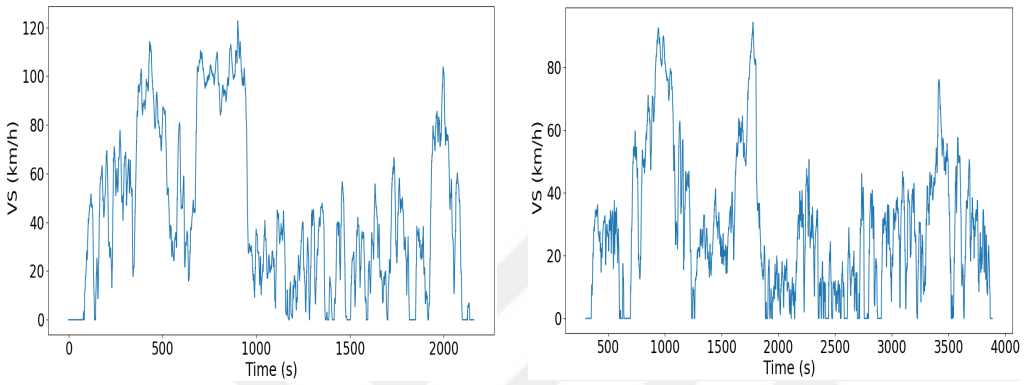
kuru tamamlama süresi yaklaşık 60 dk'dır. Maksimum ulaştığı hız 94 km/s'dir. Grafiklerde görüldüğü üzere, erkek sürücü kadın sürücüye göre daha hızlıdır. Sürücülerin hız değerleri farklı olmasına rağmen, iki sürücü de rotanın aynı kısımlarında benzer grafiksel yükselişler ve inişler sergilemiştir.

Çizelge 2.1: Uyanık aracı CAN hattı verileri.

Kod	**İsim**	**Metrik**
SWA	Direksiyon Açısı	(deg) – Float
SWRS	Direksiyon Radyal Dönüş Hızı	(deg/s) – Float
VS	Araç Hızı	(km/h) – Float
WSFR	*Sağ Ön Teker Hızı*	*(km/h) – Float*
WSFL	*Sol Ön Teker Hızı*	*(km/h) – Float*
WSRL	*Sağ Arka Teker Hızı*	*(km/h) – Float*
WSRR	*Sol Arka Teker Hızı*	*(km/h) – Float*
PGP	Gaz Pedalı Basma Yüzdesi	(% percent) – Float
ERPM	Motor Devri	(rpm) – Float
NS	Vites Boşta veya Değil	(0/1) – Bool
YR	Savrulma Oranı	(rad/s) – Float
CS	Debriyaj Durumu	(0/1) – Bool
RG	*Geri Vites*	*(0/1) – Bool*
BS	Fren Durumu	(0/1) – Bool
C	Debriyaj Maksimum	(0/1) – Bool
ADC1	*Fren Pedalı Basıncı*	*(kgf/cm2) – Float*
ADC2	*Gaz Pedalı Basıncı*	*(kgf/cm2) – Float*

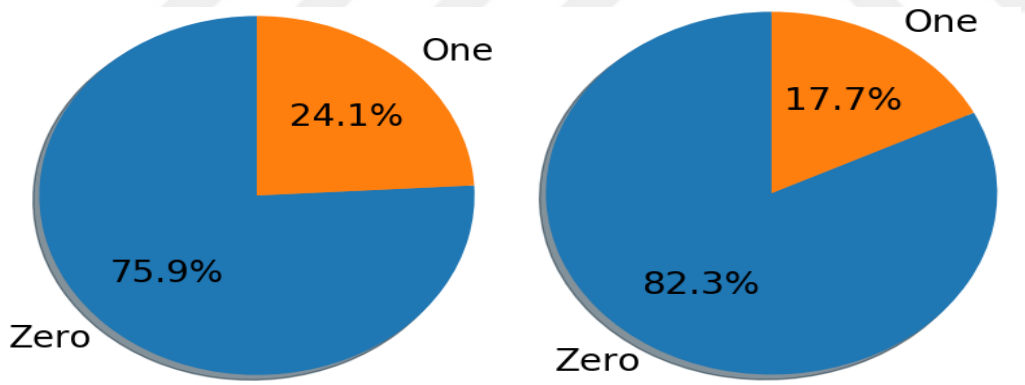
Zaman serilerinin nasıl göründüğünü göstermek amacıyla rastgele seçilen ve kod adları Erkek-2003, Kadın-1007 olan sürücüler için sırasıyla Şekil 2.2'de VS zaman grafikleri gösterilmiştir. Bu şekilde, Erkek-2003 sürücüsünün parkuru tamamlama süresi yaklaşık 36 dk'dır. Maksimum ulaştığı hız 122 km/s'dir. Kadın-1007 sürücüsünün parkuru tamamlama süresi yaklaşık 60 dk'dır. Maksimum ulaştığı hız 94 km/s'dir. Grafiklerde görüldüğü üzere, erkek sürücü kadın sürücüye göre daha hızlıdır. Sürücülerin hız değerleri farklı olmasına rağmen, iki sürücü de rotanın aynı kısımlarında benzer grafiksel yükselişler ve inişler sergilemiştir. Şekil 2.3, aynı sürücülerin fren pedalı kullanım oranlarını göstermektedir. Bu veri tipi ikili olarak ifade edilir ve fren pedalının anlık durumunu gösterir. 0 fren pedalına basılmadığını, 1 ise basıldığını göstermektedir. Erkek-2003 sürücüsü sürüşü boyunca %24 oranında fren pedalını kullanmıştır.

Kadın-1007 sürücüsü ise sürüşü boyunca %17.7 oranında fren pedalını kullanmıştır. Oranlar incelendiğinde anlaşılıyor ki, erkek sürücünün fren pedalını kullanma oranı kadın sürücüye göre daha yüksektir. Bunun sebepleri ise hızın daha yüksek olması ve daha sık gerçekleşen yukarı-aşağı hız değişimlerdir. Bununla birlikte, kadın sürücünün daha dengeli bir sürüş izlediği görülmektedir. Dört grafik birlikte değerlendirildiğinde söyleyebiliriz ki, rastgele seçilen iki sürücünün sürüş davranışları gerçekten birbirlerinden farklıdır. Dolayısıyla, bu sonuç sınıflandırma çalışmasının etkili olacağına dair iyi bir ipucudur.



(a) (b)

Şekil 2.2: (a) Erkek-2003 VS Zaman Serisi , (b) Kadın-1007 VS Zaman Serisi.



(a) (b)

Şekil 2.3: (a) Erkek-2003 Fren Pedalı Kullanma Oranı , (b) Kadın-1007 Fren Pedalı Kullanma Oranı.

3. SÜRÜCÜ KÜMELEME

Bu araştırmamızda 105 adet sürücünün CAN hattı VS ve ERPM zaman serilerine hiyerarşik kümelendirme işlemi uygulanmıştır. Amacımız sürücülerin zaman serisi verilerinin birbirlerine olan uzaklıklarını farklı metriklere göre hesaplayarak sürücü gruplandırma işlemini gerçekleştirmektir. Bu bize sürücülerin sürüş davranışlarına göre ayrıştırılabildiğini gösterecektir. Bu kısımda hiyerarşik kümeleme yöntemi ve bu yöntemle gerçekleştirilen sürücü kümeleme deneyleri anlatılmıştır. Bu deneyleri gerçekleştirirken kullandığımız veri dönüşüm metodları (DTW, DDW), veriler arası mesafe hesaplama ve veri kümeleri arası mesafe hesaplama yöntemleri anlatılmıştır. Bölüm sonunda hiyerarşik kümelendirme sonucunda ortaya çıkan dendrogram sonuçları analiz edilmiştir.

3.1 Veri Ön İşlemesi

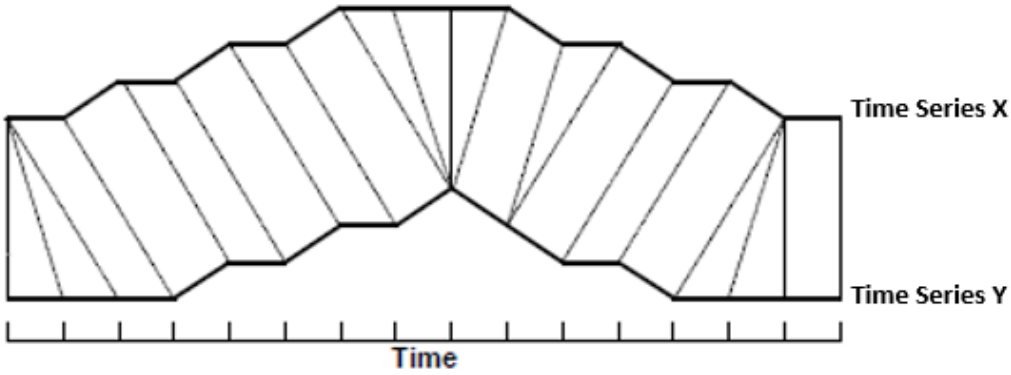
Sürücü a için bir zaman serisi verisini (örneğin VS) $TS^a =< (t_1^a, v_1^a), (t_2^a, v_2^a), \ldots, (t_{na}^a, v_{na}^a) >$ vektörü ile gösterelim. Burada, na =$|TS^a|$ vektörün boyutunu, her bir eleman (t_i^a, v_i^a) ise serinin değerinin t_i^a zamanında v_i^a olduğunu gösterir. Benzer şekilde, sürücü b için zaman serisini $TS^b = < (t_1^b, v_1^b), (t_2^b, v_2^b), \ldots, (t_{nb}^b, v_{nb}^b) >$ ile gösterelim. Her iki sürücü aynı parkuru (i) farklı farklı sürelerde tamamlayabileceğinden, yada (ii) örnekleme zaman aralıkları farklı olabileceğinden na=nb olmak zorunda değildir.

DTW gibi nokta silme, ekleme ve eşleme yaparak serileri olabildiğince birbirine benzetmeye çalışan *edit-distance* tabanlı metrikler, genetik süreçlerde olduğu gibi, farklı serilerin aynı sürecin evrimleşmiş çıktıları olduğu kabulüne dayanır. Fakat, sürüş verileri bu şekilde seriler olmadığından sürücü zaman serilerinin DTW ile karşılaştırılması yanlış sonuçlara götürebilir. Örneğin, zaman serisi değerleri VS'yi göstermek üzere $TS^a =< (0.1, 49), (0.2, 50), (0.3, 51) >$ ve $TS^b = < (0.1, 49), (0.2, 150), (0.3, 50), (0.4, 51) >$ olsun. Bu durumda b sürücüsünün (0.2, 150) noktası DTW tarafından silinerek iki seri arasındaki uzaklık sıfır olarak bulunabilir. Oysa, a sürücüsü yaklaşık 50 km/s sabit hızla giden, b sürücüsü ise ani hızlanıp (49km/s dan 150 km/s) ani yavaşlayan (150km/s dan 50 km/s) birbirine benzemez iki sürücüdür.

Uyanık veri kümesinde tüm sürücüler aynı parkuru tamamladıklarından sürüş verileri farklı olsa da katettikleri mesafe aynıdır. Bu özellikten yararlanarak aşağıda farklı iki sürücünün zaman serisi verileri arasındaki uzaklığı ölçmek amacıyla DDW (Dynamic Distance Warping) olarak adlandırdığımız bir dönüşüm tanımlanmaktadır.

3.1.1 Dynamic time warping

DTW, zaman serilerinin uzunluklarındaki farklılıklardan bağımsız olarak iki seri arasında mesafe ölçümleri gerçekleştirir. İşlem sonucunda, Şekil 3.1'de görüldüğü üzere iki zaman serisinin birbirlerine en yakın elemanları eşleştirilerek, seriler arasındaki bağlantıların yer aldığı bir bükülme örüntüsü oluşturulur. Bu yöntemin farklı kullanım alanları vardır. Buna örnek olarak seslerden sözcük tanıma sistemi verilebilir.



Şekil 3.1: İki zaman serisi arasındaki bükülme örüntüsü [16].

Problem Formülasyonu. TS^a'nın değer bileşeni X = $< x_1 = v_1^a, x_2 = v_2^a, \ldots,$ $x_{|X|} = v_{na}^a >$ ve TS^b'nin değer bileşeni Y = $< y_1 = v_1^b, y_2 = v_2^a, \ldots, y_{|Y|} = v_{nb}^b >$ olarak iki zaman serisi tanımlanmıştır. |X| ve |Y| bu serilerin uzunluklarını ifade etmektedir.

Bu iki seri kullanılarak W = $< w_1, w_2, \ldots, w_K >$ bükülme örüntüsü oluşturulmuştur. K bükülme örüntüsünün uzunluğudur ve $max(|X|, |Y|) \leq K < |X| + |Y|$ eşitsizliğine uymaktadır. Bu örüntünün k^{th} elemanı $w_k = (i, j)$ olarak gösterilir. i notasyonu X serisinin indeksini, j notasyonu ise Y serisinin indeksini göstermektedir. Aynı zamanda bu indekslerin bükülme örüntüsü içerisinde monoton bir şekilde artış göstermesi gerekmektedir.

Bükülme örüntüsünün uzunluğu Eşitlik 3.1'de [16] belirtilen formüle göre hesaplanır.

$$Dist(W) = \sum_{k=1}^{k=K} Dist(w_{ki}, w_{kj}) \tag{3.1}$$

Dist(W) bükülme örüntüsünün kümülatif uzaklığıdır. $Dist(w_{ki}, w_{kj})$ ise; X ve Y serilerinin sırasıyla i ve j indekslerinde bulunan elemanlarının arasındaki uzaklığı ifade etmektedir. Burada Euclidean mesafe hesaplama yöntemi kullanılmıştır. Aynı zamanda W serisinin k indeksinde bulunan değeridir.

Dist(W) kümülatif uzaklığının optimal sonuç için minimal olması gerekmektedir. Bunu sağlamak amacıyla dinamik programlama tekniği kullanılmıştır. Bu yaklaşımda, tüm problem tek bir seferde çözülmek yerine alt problemlere ayrıştırılır ve bu alt problemler çözülerek çözüme ulaşılır. Zaman serileri elemanları arasındaki minimum mesafeyi bulmak için Eşitlik 3.2'de belirtilen dinamik programlama formülasyonu kullanılır.

$$D(i,j) = Dist(i,j) + min[D(i-1,j), D(i,j-1), D(i-1,j-1)] \tag{3.2}$$

Bu formüle göre kümülatif uzaklık hesaplanırken, her bir eleman çifti arasında hesaplanan uzaklığa, bu elemanlardan bir birim gerideki eleman çiftleri için hesaplanan kümülatif uzaklıkların en küçüğü eklenir. Böylece eşleşen son elemanlara varıldığında, hesaplanan son kümülatif uzaklık değeri olabilecek en küçük değerde olur. Böylece iki zaman serisi arasındaki mesafe hesaplanmış olur [16].

3.1.2 Dynamic distance warping

Parkurun uzunluğu P metre olsun. Parkuru sabit hızla kateden sanal bir araç/sürücü düşünelim ve parkuru her biri p metre olan M (yani, M=P/p) parçaya ayırdığımızı varsayalım. Ayrıca her bir parçada bir adet zaman serisi ölçümü yaptığımızı (toplam M ölçüm) düşünelim. Bu durumda sanal araç s'nin bu parkurdaki zaman serisi $TS^s =< (1,v_1^s),(2,v_2^s),\ldots,(M,v_M^s) >$ olarak gösterilir. Gösterimde (j,v_j^s) çiftini, j. parçadaki serinin değeri v_j^s'dir şeklinde okuruz. Bu zaman serisini eşdeğer olarak indisleri 1 den M ye kadar olan v_j^s dizisi olarak da düşünebiliriz.

Sürücü a için $TS^a =< (t_1^a,v_1^a),(t_2^a,v_2^a),\ldots,(t_{na}^a,v_{na}^a) >$ verildiğinde bu zaman serisini M uzunluktaki sanal araç hareketinde anlatılan $TS^s = < (1,v_1^s),(2,v_2^s),\ldots,(M,v_M^s) >$ zaman serisine dönüştürme işlemi DDW olarak adlandırılır ve formal olarak DDW: $TS^a \rightarrow TS^s$ fonksiyonu olarak ifade edilir.

DDW fonksiyonu her bir $(t_i^a,v_i^a) \in TS^a$ noktasını (j, v_j^s) $\in TS^s$ noktasına eşler. Burada;

$j \leftarrow max(1, \lceil M * (t_i^a - t_1^a)/(t_{na}^a - t_1^a) \rceil)$ olarak bulunur ve $v_j^s \leftarrow v_i^a$ olarak belirlenir.

Kısaca açıklamak gerekirse, birinci aşamada, DDW fonksiyonu $[t_1^a, t_{na}^a]$ zaman aralığındaki TS^a zaman serisini [1, M] tamsayı indis aralığında tanımlı TS^s zaman serisine ölçekleyerek yerleştirir. İşlem sonrasında eşleşmeyen j noktaları ikinci bir aşamada (enterpolasyon) ise eşleşen en yakın komşusunun değeri kullanılır.

DDW fonksiyonu tüm sürücüler için uygulandığında her bir sürücü için M boyutlu bir

vektör elde edilir. Her bir vektör M boyutlu uzayda bir nokta olarak temsil edilerek bu noktalar arasındaki Minkowski uzaklıkları kolayca hesaplanabilir.

Örnek. a ve b sürücüleri için TS^a= <(0.1, 49), (0.2, 50), (0.3, 51)> ve TS^b = <(0.1, 49), (0.2, 150), (0.3, 50), (0.4, 51)> verilmiş olsun. Ayrıca M = 5 seçilmiş olsun. Bu durumda,

Birinci aşama sonunda:

DDW(TS^a)=<(1, 49), (2, ?), (3, 50), (4, ?), (5, 51)>

DDW(TS^b)=<(1, 49), (2, 150), (3, ?), (4, 50), (5, 51)> olarak hesaplanır.

İkinci aşamada ise hesaplanmayan değerler (? İle gösterilen) için kendisine en yakın indisdeki değer kullanılır. Eğer eşit uzaklıkta iki indis varsa soldaki değer en yakın kabul edilir. Bu durumda ikinci aşama sonunda dönüştürülmüş zaman serileri aşağıdaki gibi hesaplanır.

DDW(TS^a)=<(1, 49), (2, 49), (3, 50), (4, 50), (5, 51)>

DDW(TS^b)=<(1, 49), (2, 150), (3, 150), (4, 50), (5, 51)>.

DDW dönüşümü sonrası a ve b sürücülerinin vektörleri eşit boyuttadır (M=5 boyutlu). DTW'nin aksine örnektende anlaşılacağı üzere bu iki DDW vektörü arasındaki Euclidean uzaklığı hesaplanabilirdir ve daha da önemlisi uzaklık sıfır değildir.

DDW dönüşümü Algoritmik olarak Şekil 3.2'de verilmiştir. Algoritma da iki adet M ve bir adet na sayısında döngü olduğundan algoritmanın zaman karmaşıklığı O(max{na, M})'dir. Burada son döngüde j'ye en yakın indis olan k'yı bulmanın O(1)'de yapılacağı na >> M yada M >> na olmadığı müddetçe kesindir. Diğer bir deyişle eğer na/M < K (bir sabit) olduğunda, birinci aşama sonunda ? içeren vektör elemanlarının sayısı M/K'dan fazla olamaz. Hatta, ? sembolleri düzgün dağıldığından ardarda K değerden en az birisi ? olamaz. Böylelikle, bir ? değerine en yakın ? olmayan indisi bulmak O(K = sabit) = O(1)'dir.

DDW işleminin şeklin görünümünü koruduğunu göstermek amacıyla, veri kümesinden rastgele 3 adet sürücü seçilmiştir. Bu sürücülerin VS-Zaman ve ERPM-Zaman grafikleri orjinal haliyle ve DDW dönüşümü uygulanmış haliyle Şekil 3.3 ve 3.4'de gösterildiği üzere çizdirilmiş ve karşılaştırma yapılmıştır. Bunun sonucunda görülmektedir ki; her iki veri tipi içinde orjinal grafik örüntüsü ile DDW dönüşümlü grafik örüntüsü çok yüksek oranda benzeşmektedir. Buradan da DDW algoritmasının dönüşüm yaparken şekli olabildiğince koruduğunu söyleyebiliriz.

Deneylerimizde bahsi geçen veri dönüşüm tekniklerinin uygulanmasının ardından me-

safe ölçümü için uygun forma gelen veriye farklı mesafe ölçüm yöntemleri uygulanır. Deneylerde uygulanan mesafe hesaplama metrikleri DTW, Euclidean, Cityblock, Chebyshev'dir. DTW için DTW dönüşüm yöntemi, diğer 3 mesafe metriği için DDW dönüşüm yöntemi uygulanmıştır. Bunun neticesinde bir mesafe vektörü oluşturulur. Bu mesafe vektörüne Python Scipy kütüphanesinin linkage metodu average opsiyonu ile uygulanarak hiyerarşik kümelendirme işlemi gerçekleştirilir. Bu işlemin ardından ortaya çıkan sonuç vektörü dendrogram olarak görsel bir ağaç yapısında ifade edilir.

3.2 Hiyerarşik Kümeleme

Hiyerarşik kümelemede, veri kümesinde bulunan alt kümelerin hiyerarşik bağlantıları çeşitli metodlar uygulanarak oluşturulmaya çalışılır. Bu işlemin sonunda dendrogram ismi verilen bir ağaç yapısı oluşur. İki çeşit hiyerarşik kümeleme stratejisi vardır. İlki aşağıdan yukarı doğru ağacın oluşturulması yaklaşımıdır. Bu yaklaşımda her bir veri birleşerek ağacı oluşturur. Diğer yaklaşım ise yukarıdan aşağıya doğru ağacın oluşturulmasıdır. Burada ise tek bir kümeden özyinelemeli olarak dağılmalar gerçekleşir ve ağaç yapraklara doğru oluşur. Küme birleşme ve dağılma işlemleri açgözlü algoritma mantığına göre çalışır. Ağaç oluşturma yöntemi seçiminden sonra veriler arasındaki benzeşmenin ölçülebilmesi için uygun bir mesafe ölçüm yönteminin seçilmesi gerekmektedir. Bu yöntemlere, Eşitlik 3.3, 3.4 ve 3.5'da formülleri gösterilen metrikler örnek verilebilir.

Algoritma 1. Dynamic Distance Warping, DDW

Girdi. $TS^a = <(t^a_1, v^a_1), (t^a_2, v^a_2), \ldots, (t^a_{na}, v^a_{na})>$, ve M

Çıktı. $TS^s = <(1, v^s_1), (2, v^s_2), \ldots, (M, v^s_M)>$

For j=1 to M do // İlk değer atama

  $(j, v^s_j) \leftarrow (j, ?)$

For i= t^a_1 to t^a_{na} do // Birinci aşama

  $j \leftarrow \max\{1, \lceil M*(t^a_i - t^a_1)/(t^a_{na} - t^a_1) \rceil\}$

  $(j, v^s_j) \leftarrow (j, v^a_i)$

For j=1 to M do // İkinci aşama (Enterpolasyon)

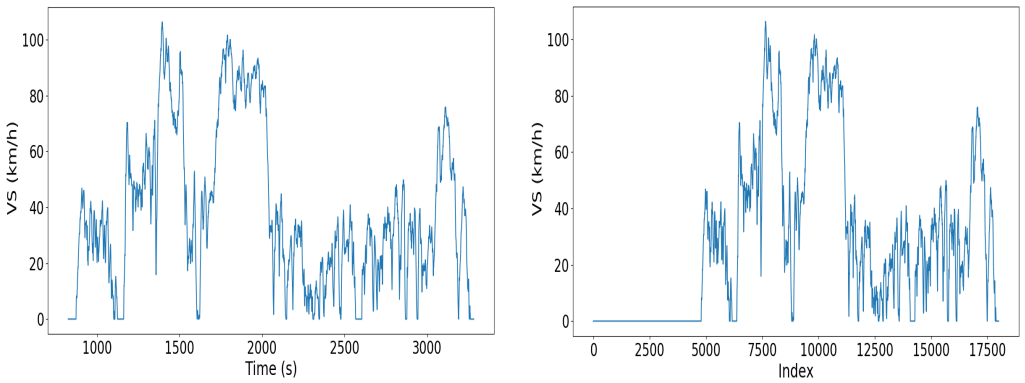
  If $(j, v^s_j) == (j, ?)$ then

    $k \leftarrow \min \{|j-k| : (k, v^s_k) != (k, ?)\}$

    $(j, v^s_j) \leftarrow (k, v^s_k)$

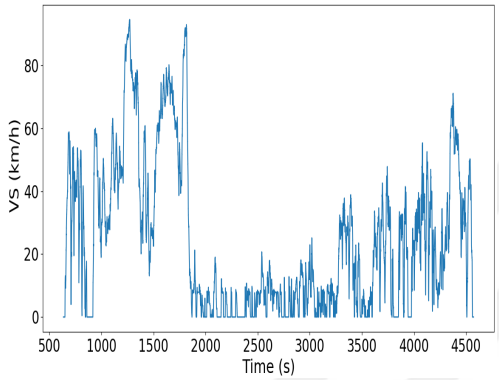
return TS^s

Şekil 3.2: Dynamic distance warping (DDW) algoritması.

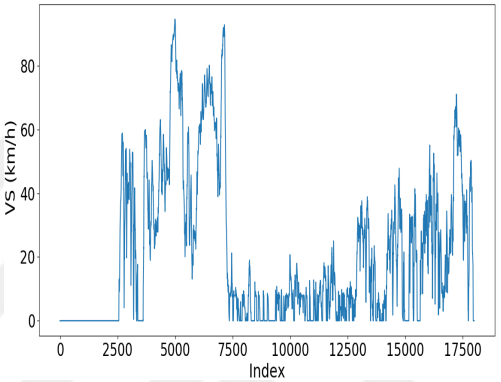


(a)

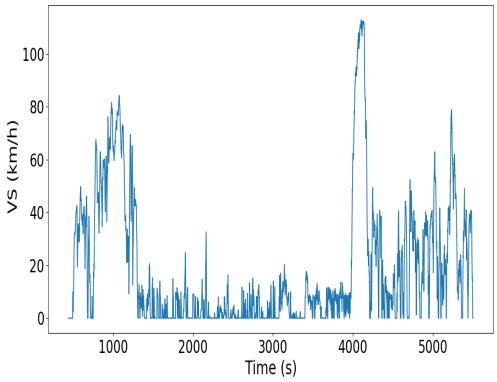
(b)



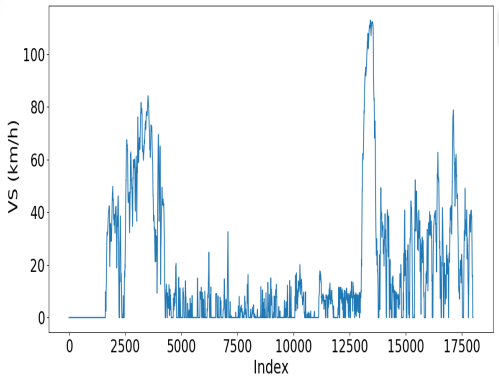
(c)



(d)

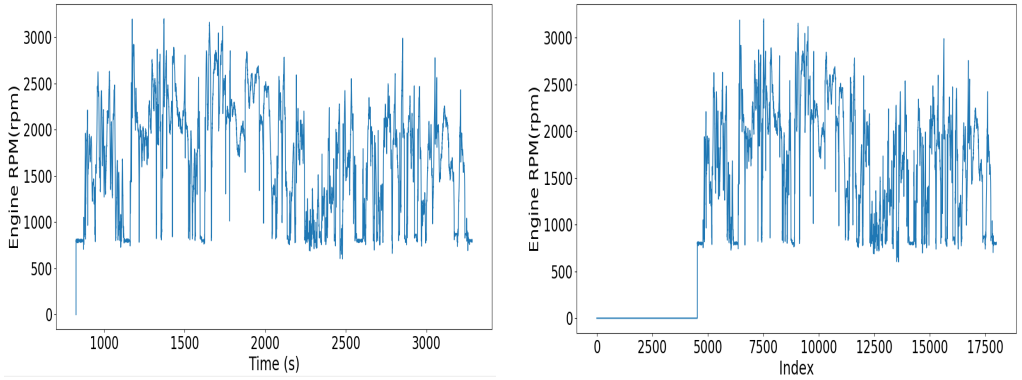


(e)

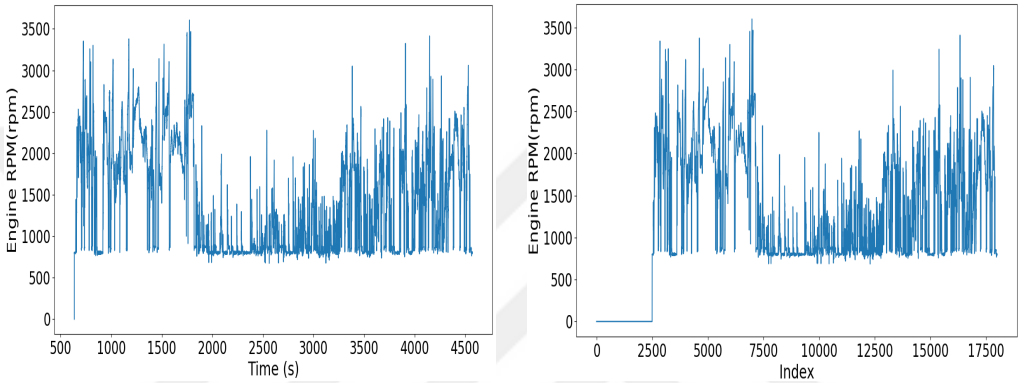


(f)

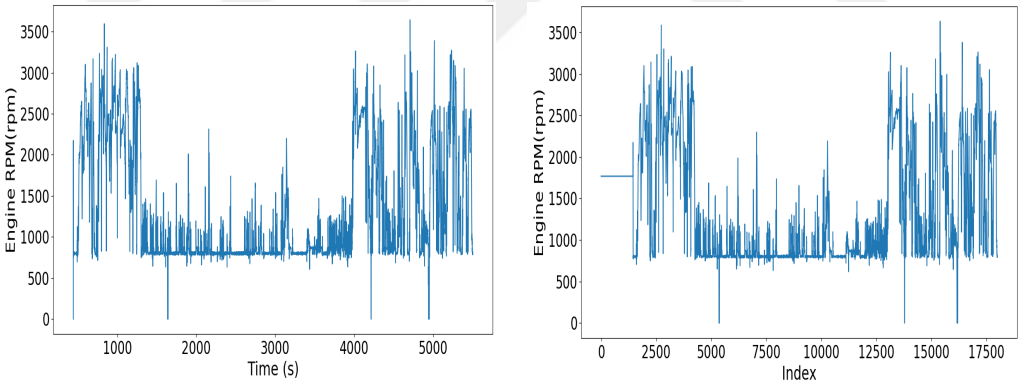
Şekil 3.3: (a) Erkek-2015(31) VS zaman serisi, (b) Erkek-2015(31) DDW dönüşümlü VS zaman serisi, (c) Erkek-2034(49) VS zaman serisi, (d) Erkek-2034(49) DDW dönüşümlü VS zaman serisi, (e) Erkek-2035(50) VS zaman serisi, (f) Erkek-2035(50) DDW dönüşümlü VS zaman serisi.



(a) (b)



(c) (d)



(e) (f)

Şekil 3.4: (a) Erkek-2015(31) ERPM zaman serisi, (b) Erkek-2015(31) DDW dönüşümlü ERPM zaman serisi, (c) Erkek-2034(49) ERPM zaman serisi, (d) Erkek-2034(49) DDW dönüşümlü ERPM zaman serisi, (e) Erkek-2035(50) ERPM zaman serisi, (f) Erkek-2035(50) DDW dönüşümlü ERPM zaman serisi.

$$Chebyshev\ Mesafesi = max(\ |v_k^a - v_k^b| : k\ \in\ \{1,2,..,M\}\ ) \tag{3.3}$$

$$Euclidean\ Mesafesi = \sqrt{\sum_{k=1}^{M}(v_k^a - v_k^b)^2} \tag{3.4}$$

$$City\ Block\ Mesafesi = \sum_{k=1}^{M}|v_k^a - v_k^b| \tag{3.5}$$

Veri uzayı için uygun metrik belirlendikten sonra *linkage* kriteri belirlenir. Bu kriter ile alt iki veri kümesi arasındaki benzeşmenin hesaplama yöntemi kararlaştırılmış olur. Bu yöntemlere örnek olarak, Eşitlik 3.6'de formülü gösterilen *average* yöntemi gösterilebilir. Bu formülde A ve B veri uzayındaki iki alt kümeyi temsil etmektedir. Dist(a,b) ifadesi ise bu kümeler içerisindeki iki noktanın arasındaki mesafeyi hesaplamada kullanılan mesafe fonksiyonunu ifade etmektedir. |A| ise A kümesinin uzunluğunu ifade eder. Bu işlem sonucunda iki küme arasındaki mesafe hesaplanmış olur. Böylece artık hiyerarşik kümelemeyi gerçekleştirmeye hazır hale geliriz.

$$Dist(A,B) = \frac{1}{|A||B|}\sum_{a\in A}\sum_{b\in B}Dist(a,b) \tag{3.6}$$

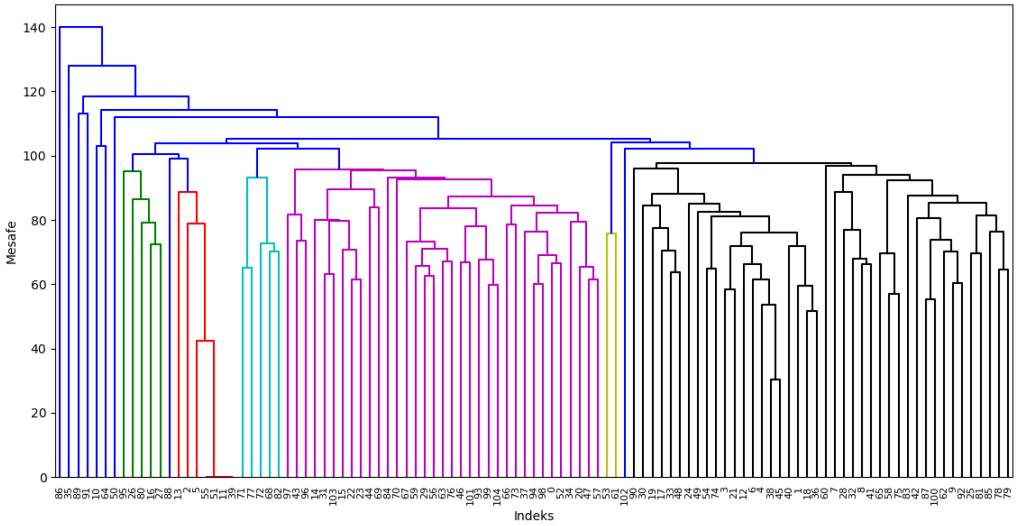
3.3 Deney Sonuçları ve Yorumlar

Bu kısımda DTW ve DDW(Euclidean, Cityblock, Chebyshev) mesafe ölçüm yöntemleri kullanılarak oluşturulan dendrogram sonuçlarını inceleyeceğiz.

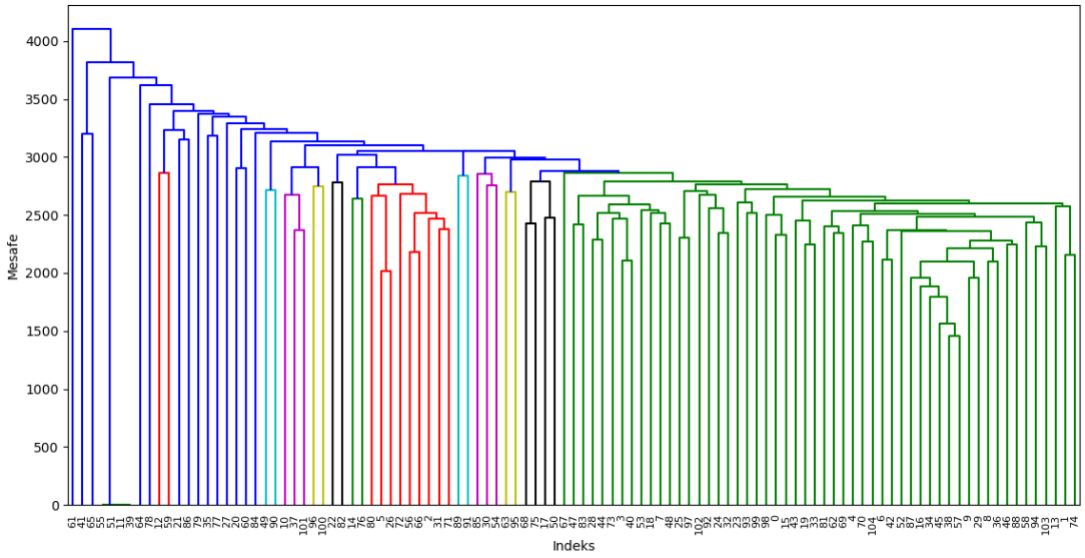
Şekil 3.5, 3.6, 3.7, 3.8, 3.9, 3.10 ve 3.11'de belirtilen dendrogramlarda y ekseninde sürücüler arasındaki mesafe, x ekseninde ise sürücülerin indeksleri belirtilmektedir. Veri kümemizde 105 adet sürücünün 17 tanesi kadın, 88 tanesi erkektir. Dendrogramlarda belirtilen sürücü indekslerinden [0,16] aralığındakiler kadın sürücüleri, [17,104] aralığındakiler ise erkek sürücüleri belirtmektedir. Dendrogram kümeleri belli mesafe eşik değerlerine göre renklendirilmişlerdir. Sonuçlar incelendiğinde kadın sürücülerin dendrogram içerisinde farklı kümelenmeler içerisinde dağılım gösterdiği, kendilerini büyük oranda kapsayan bir kümelenme yapısı oluşturamadıkları gözlemlenmiştir. Buradan VS ve ERPM CAN verileri kullanarak hiyerarşik kümeleme işlemini bahsi geçen mesafe metrikleri ile gerçekleştirdiğimizde cinsiyet ayrımının gerçekleştirilemediği çıkarımını yapabiliriz.

Dendrogramlar incelendiğinde farklı ölçüm metriklerinin farklı sürücü kümelenmelerine sebep olduğu görülmektedir. Örneğin Şekil 3.11'de görüldüğü üzere kümelendirmeler birbirine çok yakındır ve kesin bir ayrıklaştırma gözlemlenmemektedir. Bu benzerliğin sebebinin DTW'nin mesafe hesaplarken bazı uç verileri yok saymasından kaynaklandığını tespit ettik. Bu kıstas ve sürüş süreleri dikkate alınarak sonuçlar incelendiğinde daha ideal formda olan kümelenmeye euclidean deneyi kullanılarak ulaşılmıştır.

Şekil 3.7 ve 3.8'de gösterilen dendrogramlarda ki sürücü dağılımının doğruluğunu test etmek amacıyla her bir CAN verisi için iki farklı deney gerçekleştirilmiştir. İlk deneyde ağacın en alt seviyelerinde aynı köke bağlı 6 kişilik bir sürücü grubu seçilmiştir. Dendrogram yapısı gereği bu kardeşlerin mesafelerinin birbirine yakın olması gerektiğini beklemekteyiz. Diğer deneyde ise ağacın farklı derinlik seviyelerinden 4 sürücü seçilmiştir. Burada ise dendrogram yapısı gereği farklı derinlik seviyelerinde ki verilerin aralarında ki mesafelerin fazla olması gerektiğini beklemekteyiz. Şekil 3.7 ve 3.8'de deneyler için seçilen sürücüler çerçeve içerisinde gösterilmiştir. Bu grafiklerin ve varsayımlarımızın doğruluğunu göstermek amacıyla seçilen sürücülerin VS zaman ve ERPM zaman grafikleri oluşturulmuştur. Şekil 3.12 ve 3.13'de ağacın en alt seviyelerinden aynı köke bağlı olarak seçilen 6 adet sürücünün sırasıyla VS zaman ve ERPM zaman grafikleri gösterilmektedir. Grafikler incelendiğinde sürücülerin sürüş sürelerinin birbirine yakın olduğu ve grafik örüntülerinin benzer olduğu gözlemlenmektedir. Şekil 3.14'de ve 3.15'da ağacın farklı derinlik seviyelerinden seçilen 4 adet sürücünün sırasıyla VS zaman ve ERPM zaman grafikleri gösterilmektedir. Grafikler incelendiğinde sürüş sürelerinin birbirine yakın olmadığı ve grafik örüntülerinin farklı olduğu gözlemlenmektedir.

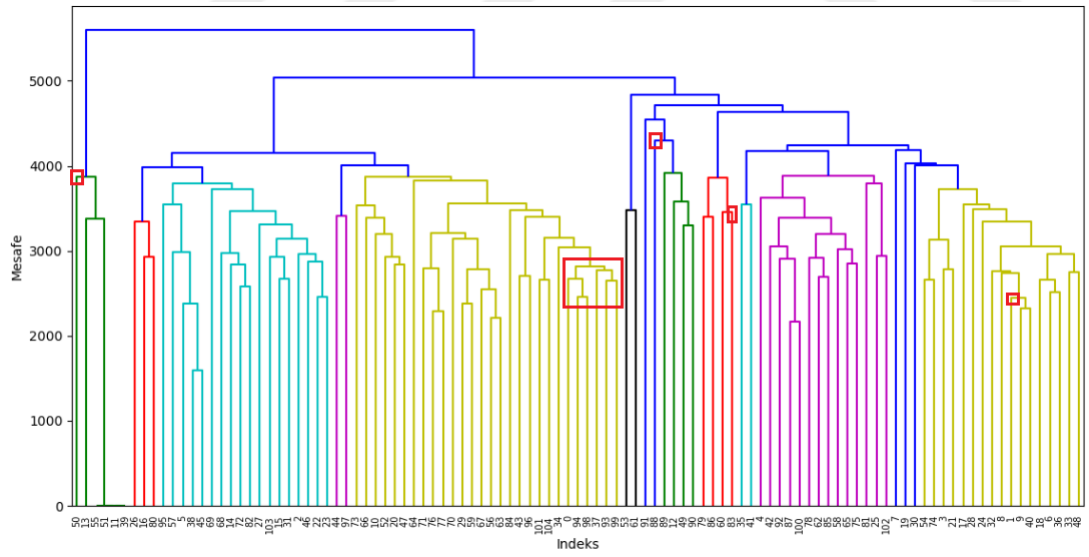


Şekil 3.5: DDW(Chebyshev) CAN VS dendrogram sonucu.



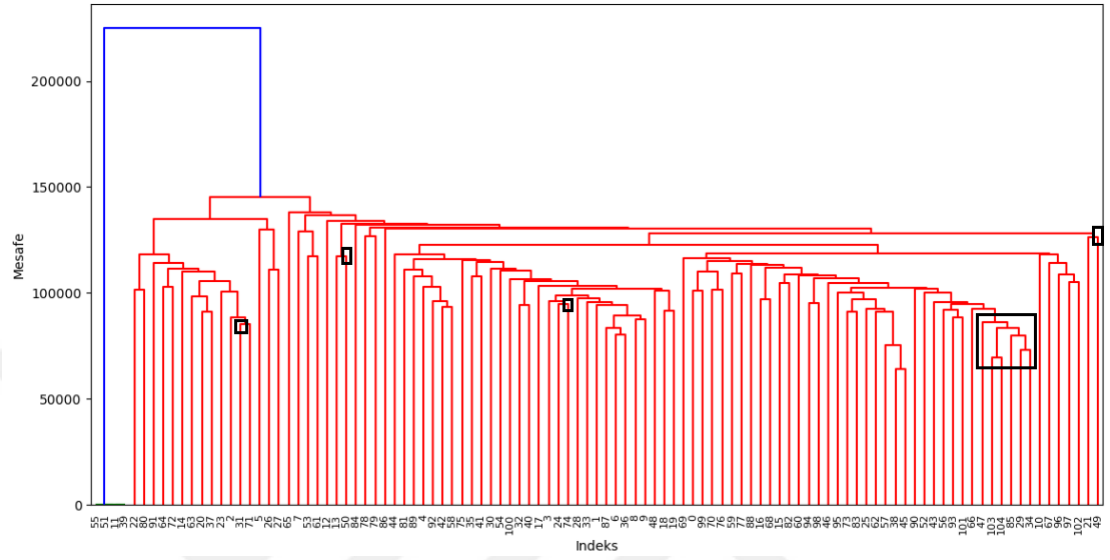
Şekil 3.6: DDW(Chebyshev) CAN ERPM dendrogram sonucu.

Şekil 3.7'de yer alan çerçeveler Şekil 3.12 ve 3.14'de grafikleri çizdirilen sürücüleri belirtmektedir. Görselde toplamda 5 farklı sürücü kümesi (çerçeve) bulunmaktadır. Bu kümeler içerisinde yer alan sürücüler sistem numarası(indeks) kullanılarak soldan sağa doğru sırasıyla [2035(50)], [1003(0), 2079(94), 2083(98), 2022(37), 2078(93), 2084(99)], [2073(88)], [2068(83)], [1004(1)] olarak ifade edilmektedir.

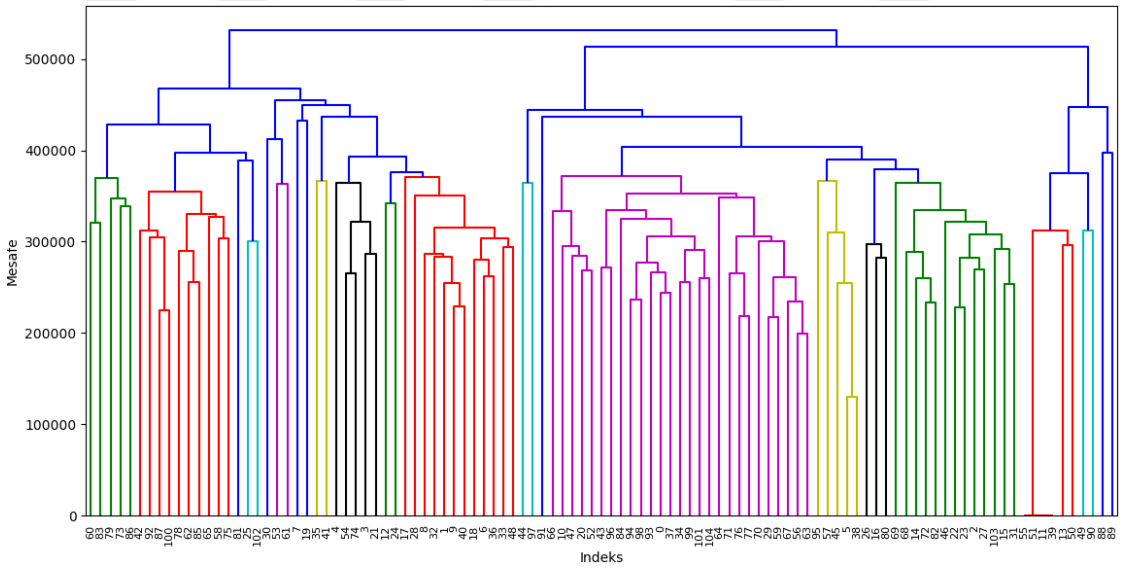


Şekil 3.7: DDW(Euclidean) CAN VS dendrogram sonucu.

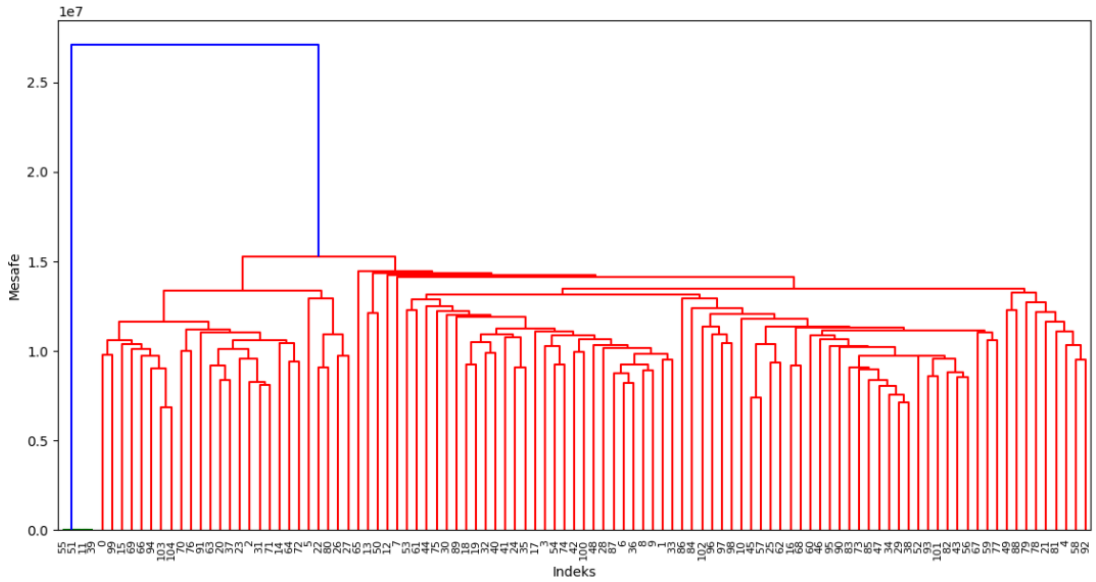
Şekil 3.8'de yer alan çerçeveler Şekil 3.13 ve 3.15'da grafikleri çizdirilen sürücüleri belirtmektedir. Görselde toplamda 5 farklı sürücü kümesi (çerçeve) bulunmaktadır. Bu kümeler içerisinde yer alan sürücüler sistem numarası(indeks) olarak soldan sağa doğru sırasıyla [2015(31)], [2035(50)], [2059(74)], [2032(47), 2080(103), 2089(104), 2070(85), 2013(29), 2019(34)], [2034(49)] olarak ifade edilmektedir.



Şekil 3.8: DDW(Euclidean) CAN ERPM dendrogram sonucu.

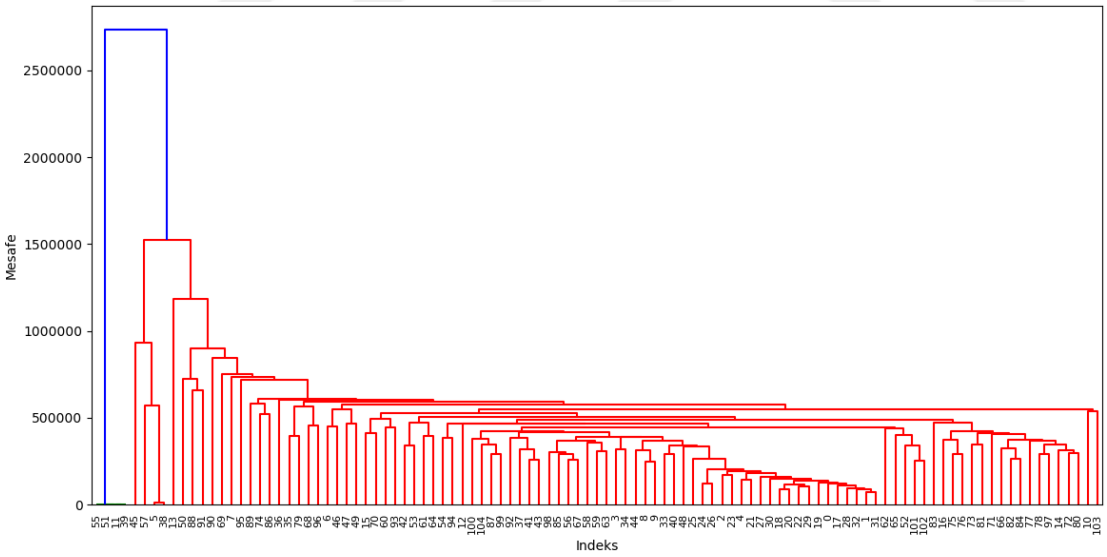


Şekil 3.9: DDW(City Block) CAN VS dendrogram sonucu.

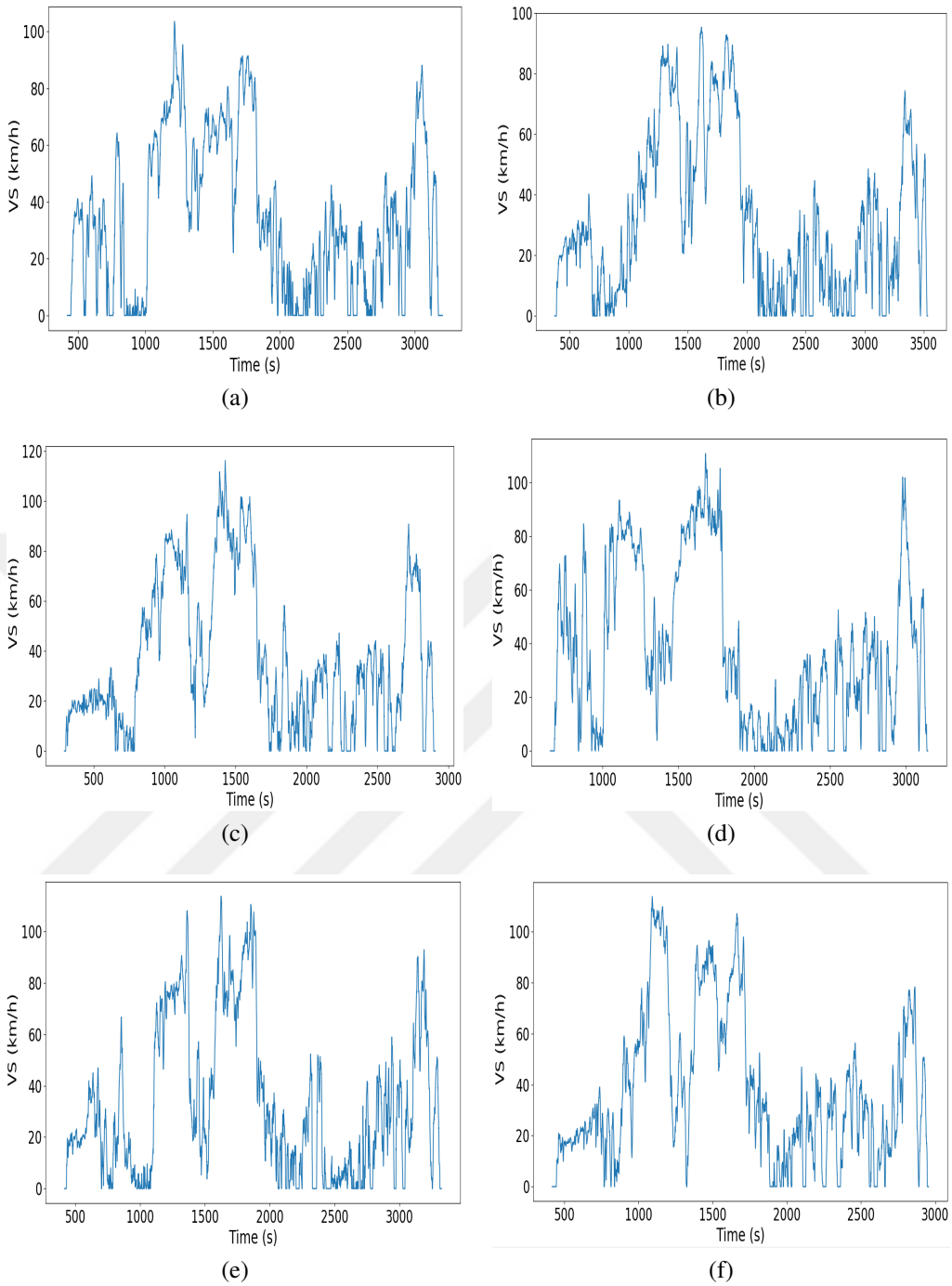


Şekil 3.10: DDW(City Block) CAN ERPM dendrogram sonucu.

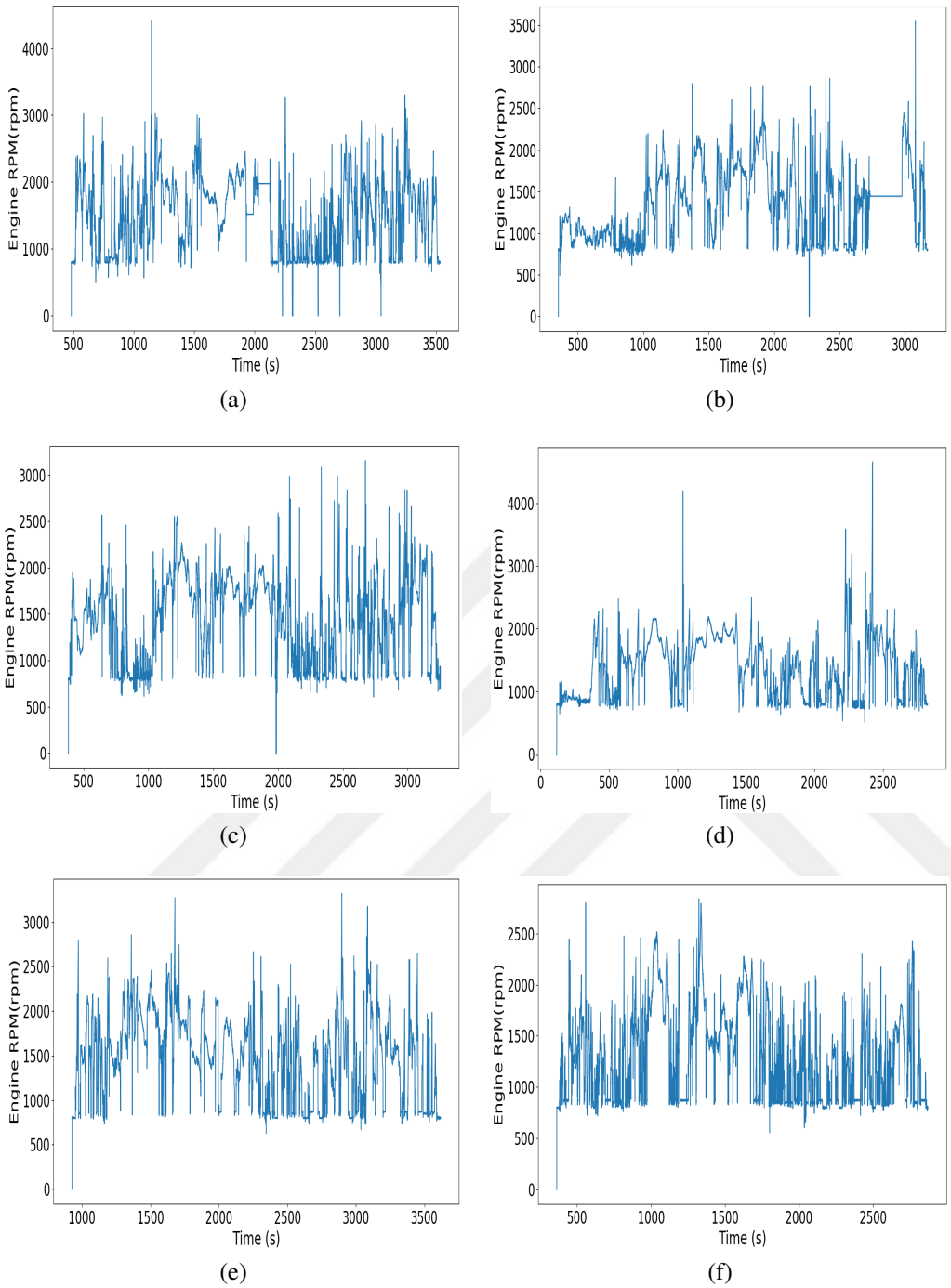
Şekil 3.11'de dendrogram sonucu gösterilen deneyde, mesafe ölçüm yöntemi olarak euclidean kullanılmıştır. Görselde görüldüğü üzere sürücü kümelerinin kapasitesi az ve birleşme noktalarının derinlik seviyeleri birbirine çok yakındır. Bu yüzden ayırt edici bir sonuç alınamamıştır.



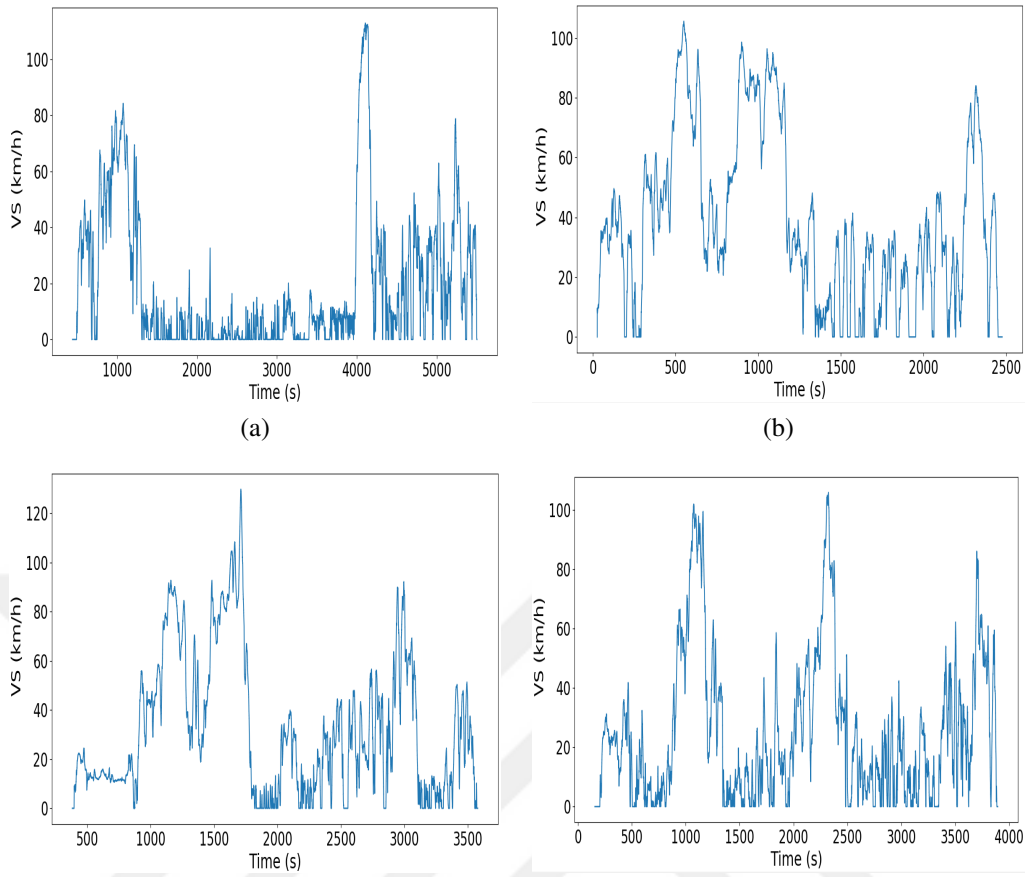
Şekil 3.11: DTW(Euclidean) CAN VS dendrogram sonucu.



Şekil 3.12: (a) Kadın-1003(0) VS zaman serisi, (b) Erkek-2079(94) VS zaman serisi, (c) Erkek-2083(98) VS zaman serisi, (d) Erkek-2022(37) VS zaman serisi, (e) Erkek-2078(93) VS zaman serisi, (f) Erkek-2084(99) VS zaman serisi.



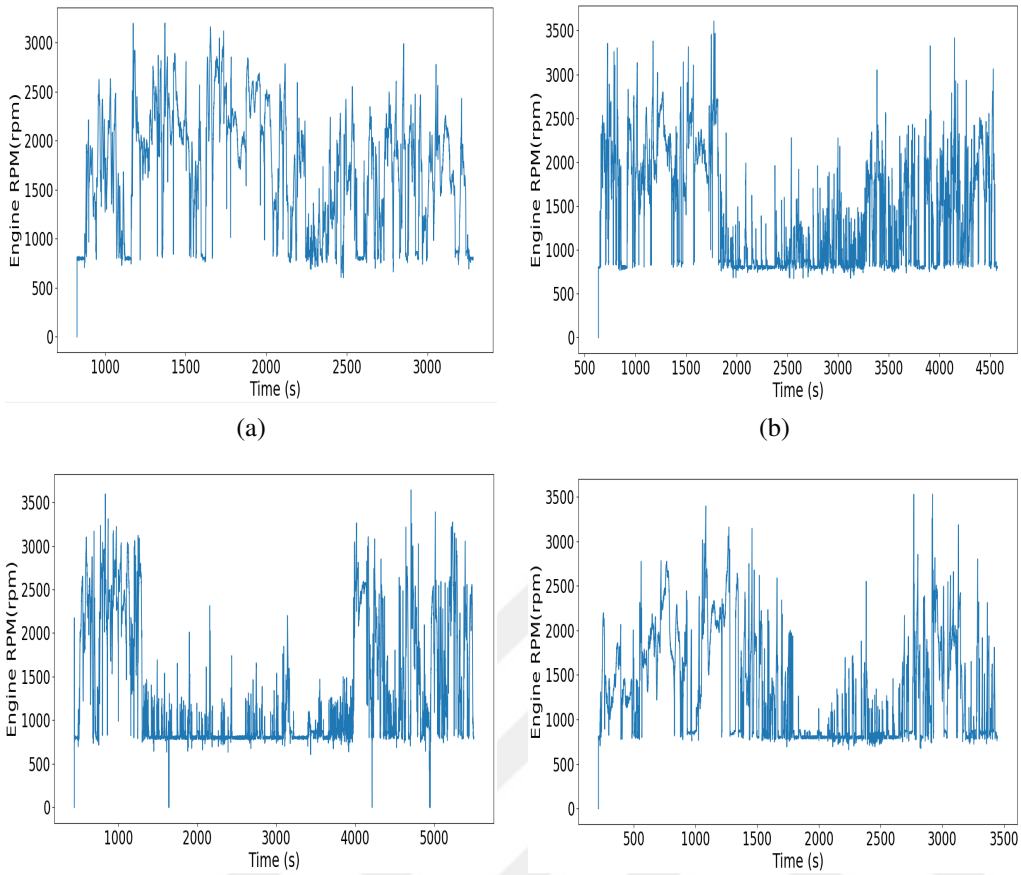
Şekil 3.13: (a) Erkek-2032(47) ,ERPM zaman serisi, (b) Erkek-2088(103) ERPM zaman serisi, (c) Erkek-2089(104) ERPM zaman serisi, (d) Erkek-2070(85) ERPM zaman serisi, (e) Erkek-2013(29) ERPM zaman serisi, (f) Erkek-2019(34) ERPM zaman serisi.



Şekil 3.14: (a) Erkek-2035(50) VS zaman serisi, (b) Kadın-1004(1) VS zaman serisi, (c) Erkek-2068(83) VS zaman serisi, (d) Erkek-2073(88) VS zaman serisi.

3.4 Tartışma

Sürücü kümelendirme araştırmamızda sürücü VS ve ERPM verilerini kullanarak 4 farklı mesafe ölçüm yöntemi ile hiyerarşik kümelendirme yapılmıştır. Dendrogram grafiklerinde cinsiyet ayrımının gerçekleştirilemediği tespit edilmiştir. Buna kadın sürücülerin sürüş sürelerinin, kendilerinden sayıca çok daha fazla olan erkek sürücülerin sürüş süreleri ile benzerlik göstermesi ve hiyerarşik kümelendirme yönteminin VS ve ERPM verileri ile tek başına bu konuda ayırt edici olamamasının sebep olduğu anlaşılmıştır. Bu konunun haricinde Euclidean sonuçları incelendiğinde sürüş sürelerine göre sürücü gruplandırmanın tutarlı olduğu gözlemlenmiştir.



Şekil 3.15: (a) Erkek-2015(31) ERPM zaman serisi, (b) Erkek-2034(49) ERPM zaman serisi, (c) Erkek-2035(50) ERPM zaman serisi, (d) Erkek-2059(74) ERPM zaman serisi.

4. CİNSİYET SINIFLANDIRMA

Bu araştırma sürücü CAN hattı verileri üzerinden sürücünün cinsiyetini tespit etmeyi amaçlamaktadır. Bu kısım da ve Bölüm 5'de gerçekleştirilen deneyler veri sınıflandırma deneyleridir. Bu deneylerin bir gereği olarak uygulanan öznitelik çıkarımı ve sınıflandırma algoritmalarına bu bölümde yer verilmiştir. Bunların haricinde, cinsiyet sınıflandırma deneyleri için gerekli olan veri ön işleme süreçleri anlatılmış ve bu deneylerin sonuçları analiz edilmiştir.

4.1 Öznitelik Çıkarımı

Tüm sürücüler aynı rotada sürüş yapmasına rağmen, sürüşlerini tamamlama süreleri aynı değildir. Şekil 2.2'de de açıkça görüldüğü üzere sürüş tamamlanma süreleri sırasıyla yaklaşık 36 dakika ve 60 dakikadır. Sürüşler gerçek trafikte farklı zamanlarda gerçekleştirildiğinden o andaki trafik yoğunluğu ve birçok çevresel faktör sürüş hızına etki edebilmektedir. Dolayısıyla, sürüş süresinden bağımsız ve sabit bir öznitelik kümesine ihtiyacımız vardır. Bu amaçla Python Tsfresh kütüphanesi [17] kullanılmıştır. Bu kütüphane, herhangi bir uzunluktaki zaman serisi verisinden istenilen sayıda öznitelik çıkarımı yapabilmektedir.

Tsfresh matematiksel fonksiyonlardan oluşan bir Python kütüphanesidir. Her fonksiyon bir özniteliğe karşılık gelmektedir. Temel olarak görevi bu fonksiyonlara girdi olarak verilen zaman serisi verilerinden sınıflandırma için kritik olabilecek istatiksel sonuçlar oluşturmaktır. Kullanılacak fonksiyonlar konfigürasyona bağlı olarak seçilebilir. Biz araştırmamızda iki farklı konfigürasyon kullandık. İlki 8 adet öznitelik çıkarımı yapmaktadır. Bunlar; *sum_values*, *median*, *mean*, *length*, *standard_deviation*, *variance*, *maximum* ve *minimum* matematiksel fonksiyonlarıdır. İkincisi ise bu fonksiyonları da içerisinde barındırmaktadır yani daha kapsamlıdır ve 216 adet öznitelik çıkarımı yapabilmektedir. 216 adet öznitelik çıkarımda kullanılan matematiksel fonksiyonlar şunlardır;

Formüllerde yer alan x notasyonu her bir sürücünün bir adet CAN veri tipi için kayıt altına alınan zaman serisini ifade etmektedir. Bu durumda sürücü s'nin tek bir CAN veri tipi için zaman serisi $\mathbf{x} = < (1,x_1),(2,x_2),\ldots,(n,x_n) >$ olarak gösterilir.

Abs_Energy(x). Zaman serisi verilerinin karelerinin toplamıdır. Eşitlik 4.1'de belirtilen formüle göre hesaplanır.

$$R = \sum_{i=1}^{n} x_i^2 \tag{4.1}$$

Absolute_Sum_of_Changes(x). Zaman serisinde ardışık veri çiftlerinin arasındaki farkın mutlak değerlerinin toplamıdır. Eşitlik 4.2'de belirtilen formüle göre hesaplanır.

$$R = \sum_{i=1}^{n-1} |\, x_{i+1} - x_i \,| \tag{4.2}$$

Ar_Coefficient(x, (coeff, k)). Bu fonksiyon Auto Regressive işleminin maksimum koşulsuz benzerliğini hesaplamaktadır. (coeff, k) = $< (\varphi_1, k_1), (\varphi_2, k_2) >, \ldots, (\varphi_m, k_m) >$ olarak ifade edilir. m ifadesi bu fonksiyon kullanılarak üretilecek öznitelik sayısını belirtmektedir. Yani bu dizin içerisindeki her bir eş sürücü için yeni bir özniteliği belirtmektedir. k ifadesi ise o öznitelik için kullanılacak maksimum lag değerini belirtmektedir. Bu öznitelikler Eşitlik 4.3'de belirtilen formüle göre hesaplanır.

$$x_t = \varphi_0 + \sum_{i=1}^{k} \varphi_i x_{t-i} + \varepsilon_t \tag{4.3}$$

Augmented_Dickey_Fuller(x). Augmented dickey fuller hipotez testi zaman serisinde birim kökün olup olmadığını kontrol eder ve ilgili testlerin istatistik değerlerini hesaplar.

Autocorrelation(x, lag). Bu fonksiyon parametre olarak gelen lag değerinin oto korelasyonunu Eşitlik 4.4'de belirtilen formüle göre hesaplar. Her bir lag değeri yeni bir öznitelik olarak hesaplanır. n ifadesi zaman serisinin uzunluğunu, σ^2 varyans değerini, μ ise ortalama değerini belirtmektedir.

$$R = \frac{1}{(n-1)\sigma^2} \sum_{t=1}^{n-1} (x_t - \mu)(x_{t+1} - \mu) \tag{4.4}$$

Count_Above_Mean(x). Bu fonksiyon zaman serisinin ortalama değerinden büyük değerlerin sayısını hesaplar.

Count_Below_Mean(x). Bu fonksiyon zaman serisinin ortalama değerinden küçük değerlerin sayısını hesaplar.

Cwt_Coefficients(x, (coeff, w)). Bu fonksiyon *Mexican hat wavelet* olarak da bilinen *Ricker wavelet* için devamlı dalgacık dönüşümünü hesaplar. (coeff, w) = $< (\varphi_1, \alpha_1),$

$(\varphi_2, \alpha_2), \ldots, (\varphi_m, \alpha_m) >$ olarak ifade edilir. α dalgacık fonksiyonunun genişlik parametresini belirtir. m ifadesi bu fonksiyon kullanılarak üretilecek öznitelik sayısını belirtmektedir. Yani bu dizin içerisindeki her bir eş sürücü için yeni bir özniteliği ifade etmektedir.

First_Location_of_Maximum(x). Zaman serisinin maksimum değerinin ilk konumunu dönmektedir.

First_Location_of_Minimum(x). Zaman serisinin minimum değerinin ilk konumunu dönmektedir.

Has_Duplicate(x). Zaman serisi içerisindeki herhangi bir verinin birden fazla olup olmadığını kontrol eder.

Has_Duplicate_Max(x). Zaman serisinin maksimum değerinin birden fazla olup olmadığını kontrol eder.

Has_Duplicate_Min(x). Zaman serisinin minimum değerinin birden fazla olup olmadığını kontrol eder.

Index_Mass_Quantile((x,q)). Bu fonksiyon zaman serisinin kümelenme indeks değerini q%'ya göre hesaplar. Her bir q değeri sürücü için yeni bir öznitelik anlamına gelmektedir.

Kurtosis(x). Zaman serisinin Kurtosis değerini döner.

Large_Standard_Deviation(x, r). Eşitlik 4.5'de belirtilen formüle göre zaman serisinin standart sapmasının karşılaştırmasını yapar. Her bir r değeri sürücü için yeni bir öznitelik anlamına gelmektedir.

$$R = std(\mathbf{x}) > r \times (max(\mathbf{x}) - min(\mathbf{x})) \tag{4.5}$$

Last_Location_of_Maximum(x). Zaman serisinin maksimum değerinin bulunduğu son konumu döner.

Last_Location_of_Minimum(x). Zaman serisinin minimum değerinin bulunduğu son konumu döner.

Length(x). Zaman serisinin uzunluğunu döner.

Longest_Strike_Above_Mean(x). Zaman serisinin ortalama değerinden büyük verilerden oluşan en uzun ardışık alt serinin uzunluğunu hesaplar.

Longest_Strike_Below_Mean(x). Zaman serisinin ortalama değerinden küçük veri-

lerden oluşan en uzun ardışık alt serinin uzunluğunu hesaplar.

Maximum(x). Zaman serisinin maximum değerini döner.

Mean(x). Zaman serisi verilerinin ortalama değerini döner.

Mean_Abs_Change(x). Ardışık zaman serisi verilerinin aralarındaki farkın mutlak değerlerinin ortalamasını Eşitlik 4.6'da belirtilen formüle göre hesaplar.

$$R = \frac{1}{n}\sum_{i=1}^{n-1}|x_{i+1} - x_i| \tag{4.6}$$

Mean_Change(x). Ardışık zaman serisi verilerinin aralarındaki farkların ortalamasını Eşitlik 4.7'de belirtilen formüle göre hesaplar.

$$R = \frac{1}{n}\sum_{i=1}^{n-1}x_{i+1} - x_i \tag{4.7}$$

Median(x). Zaman serisinin medyan değerini döner.

Minimum(x). Zaman serisinin minimum değerini döner.

Number_Cwt_Peaks(x, n). Bu fonksiyon zaman serisinde farklı veri aralıklarındaki tepe noktalarını tespit eder ve sayılarını hesaplar. n parametresi maksimum veri genişliğini ifade eder. Her bir n değeri sürücü için yeni bir öznitelik anlamına gelmektedir.

Percentage_of_Reoccurring_Datapoints_to_All_Datapoints(x). Bu fonksiyon zaman serisinde birden fazla bulunan verilerin toplam veriye oranını yüzde olarak hesaplar.

Quantile(x, q). Bu fonksiyon zaman serisinin q dağılımını hesaplar. Bu sonuç zaman serisi sıralandığında %q'dan daha büyük değerleri ifade eder. Her bir q değeri sürücü için yeni bir öznitelik anlamına gelmektedir.

Range_Count(x, min, max). Bu fonksiyon zaman serisinde [Min-Max) aralığındaki değerlerin sayısını döner.

Ratio_Value_Number_to_Time_Series_Length(x). Bu fonksiyon zaman serisinde sadece tek bir örneği bulunan veri sayısının tüm veri sayısına oranını hesaplar.

Skewness(x). Bu fonksiyon zaman serisinin örnek çarpıklık değerini hesaplar.

Spkt_Welch_Density(x, coeff). Bu fonksiyon, farklı frekanslarda zaman serisinin

çapraz güç spektrum yoğunluğunu tahmin eder. Bunu yapmak için, zaman serileri önce zaman etki alanından frekans etki alanına kaydırılır.

Standard_Deviation(x). Zaman serisinin standart sapma değerini döner.

Sum_of_Reoccurring_Data_Points(x). Zaman serisinde birden fazla bulunan veri noktalarının toplamını döner.

Sum_of_Reoccurring_Values(x). Zaman serisinde birden fazla bulunan verilerin değerlerinin toplamını döner.

Sum_values(x). Zaman serisindeki verilerin değerlerinin toplamını döner.

Symmetry_Looking(x, r). Eşitlik 4.8'de belirtilen formül uygulanarak zaman serisinin dağılımının simetrik olup olmadığına karar verilir. r parametresi veri aralık yüzdesini belirtmektedir. Her bir r değeri sürücü için yeni bir öznitelik anlamına gelmektedir.

$$R = |mean(\mathbf{x}) - median(\mathbf{x})| < r \times (max(\mathbf{x}) - min(\mathbf{x})) \tag{4.8}$$

Time_Reversal_Asymmetry_statistic(x, lag). Bu fonksiyon Eşitlik 4.9'da belirtilen formülü hesaplamaktadır. Her bir lag değeri sürücü için yeni bir öznitelik anlamına gelmektedir.

$$R = \frac{1}{n-2lag} \sum_{i=0}^{n-2lag} x_{i+2lag}^2 \, x_{i+lag} - x_{i+lag} \, x_i^2 \tag{4.9}$$

Value_Count(x, value). Zaman serisindeki değerlerin bulunma sayılarını döner. value sayılacak değeri ifade etmektedir. Her bir value değeri sürücü için yeni bir öznitelik anlamına gelmektedir.

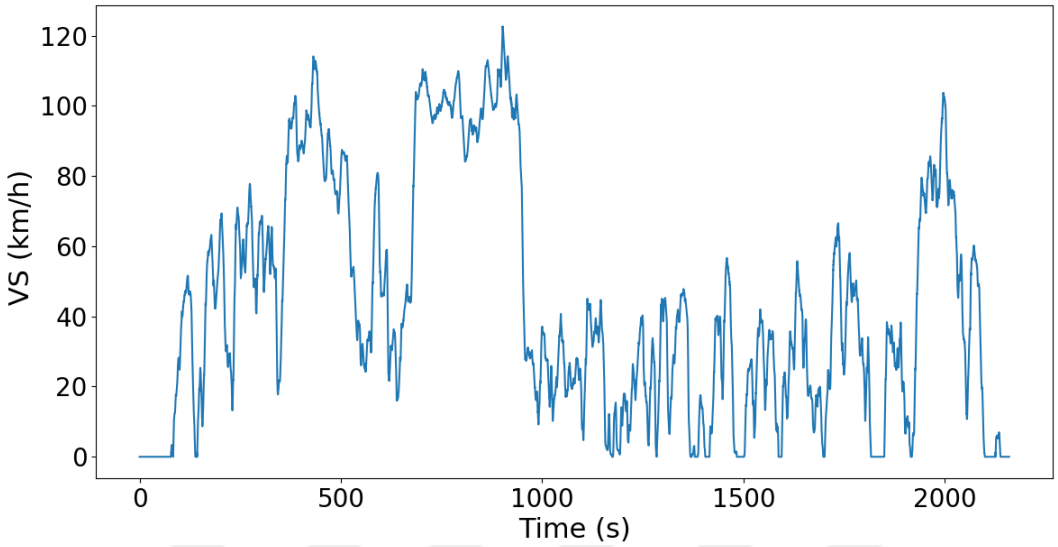
Variance(x). Zaman serisinin varyansını hesaplar.

Variance_Larger_than_Standard_Deviation(x). Zaman serisinin varyansının standart sapmasından büyük olup olmadığını gösterir [17].

Öznitelik çıkarım işlemi uygulandıktan sonra, her bir sürücü için CAN hattı verisine özel karakteristiklerini ifade eden özniteliklerin olduğu, tablo formunda yeni bir uzay oluşturmuş oluruz.

Şekil 4.1'de Erkek-2003 sürücüsünün ham hız zaman serisi verisi grafik olarak gösterilmektedir. Çizelge 4.1'de ise bu sürücünün Şekil 4.1'de gösterilen ham VS CAN hattı verisine öznitelik çıkarımı uygulanması sonucunda ortaya çıkan 216 adet özniteliğin

bir parçası ifade edilmektedir. Ayrıca bu çizelgede koyu renk ile yazılanlar Tsfresh 8'li öznitelik konfigürasyonunda yer alan öznitelikleri belirtmektedir. Veri ön işleme ve sınıflandırma aşamalarında artık bu yeni oluşan veriler işlem görmektedir.



Şekil 4.1: Erkek-2003 VS zaman serisi.

4.2 Sınıflandırma Algoritmaları

Deneylerimizde Weka SVM, J48, RF, ABM1, MLP, VP and BN sınıflandırma algoritmaları kullanılmıştır. Algoritmaların prensipleri ve çalışmamıza adaptasyonu sonraki adımda anlatılmaktadır.

SVM'nin temel mantığı doğrusal olarak ayrıştırılabilen veri yapıları için en iyi ayırıcı düzlemin belirlenmesidir. SVM sınıflandırıcıları, aralığı maksimum yapan en optimal ayırıcı düzlemi oluşturmaya çalışır. Doğrusal olarak ayrıştırılamayan veri yapıları, dönüşüm tekniği ile farklı bir boyuta taşınarak çözülür. Bu dönüşüm kernel fonksiyonları uygulanarak gerçekleştirilir. Deneylerde polinom kernel fonksiyonu kullanılmıştır. Weka'da bulunan SMO yöntemi SVM algoritması tabanlıdır. Bu yöntem sezgiler kullanarak SVM eğitim verisini daha küçük problemlere böler ve çözer. Böylece bu süreç daha hızlı bir hale getirilebilir. Aynı zamanda veri normalleştirmesi de yapmaktadır [18]. Deneylerimizde SMO yöntemi kullanılmıştır.

J48 algoritması bir karar ağacı algoritmasıdır. C4.5 algoritmasının Weka'da ki açık kaynak kodlu halidir. Öznitelik entropi hesaplarında bilgi kazanım teorisini kullanır.

Budama yöntemi olarak ağaç oluşum sonrası budama kullanılır.

Çizelge 4.1: Erkek-2003 VS CAN hattı verisine öznitelik çıkarımı uygulanması sonucunda ortaya çıkan 216 özniteliğin bir kısmı.

Öznitelik Adı	**Değer**
Variance_larger_than_standard_deviation	1
Has_duplicate_max	0
Has_duplicate_min	1
Has_duplicate	1
Sum_values	860726.89
Augmented_dickey_fuller	-4.04
Abs_energy	59803957.91
Mean_abs_change	0.21
Mean_change	-7.20E-19
Median	35.71
Mean	43.58
Length	19748
Standard_deviation	33.59
Variance	1128.65
Skewness	0.51
Kurtosis	-0.90
Absolute_sum_of_changes	4321.96
Longest_strike_below_mean	1478
Longest_strike_above_mean	2685
Count_above_mean	8362
Count_below_mean	11386
Last_location_of_minimum	1
First_location_of_minimum	0
Percentage_of_reoccurring_datapoints_to_all_datapoints	0.95
Sum_of_reoccurring_values	101118.32
Sum_of_reoccurring_data_points	851498.74
Ratio_value_number_to_time_series_length	0.094
Maximum	122.56
Minimum	0
Time_reversal_asymmetry_statistic_lag_1	-10.41
Time_reversal_asymmetry_statistic_lag_2	-39.40
Time_reversal_asymmetry_statistic_lag_3	-86.18
Large_standard_deviation_r_0.0	1
Large_standard_deviation_r_0.1	1
Large_standard_deviation_r_0.05	1

Bu algoritmanın tercih edilme sebebi anlaşılması ve yorumlanması basittir ve problemimizde ki öznitelikler arasındaki korelasyonun az olması sebebiyle tercih edilmiştir [19].

Random forest kolektif bir karar ağacı makine öğrenme algoritmasıdır. Bu algoritmada tüm veri için tek bir karar ağacı oluşturmak yerine, veri önceden boyutu belirlenmiş parçalara bölünür ve her bir parça için bir karar ağacı oluşturulur. Daha sonra bu karar ağaçlarından çıkan sonuçların birleştirilmesi ile nihai sonuç elde edilir. Verimizin karar ağaçları yapısına uygunluğundan ve literatürdeki kullanımının fazla oluşundan dolayı tercih edilmiştir [20].

AdaBoostM1 kolektif bir makine öğrenme algoritmasıdır. Amacı zayıf sınıflandırıcılar kullanarak güçlü bir sınıflandırıcı oluşturmaktır. Bu algoritma sınıflandırıcı olarak tek seviyeli karar ağaçları (Decision Stamp) kullanarak iki sınıflı bir problemde sınıflandırma yaptığında yüksek oranda doğruluk oranları vermektedir. Algoritma sınıflandırıcılarının verdiği ağırlıklı tahminlerin toplamının sonucuna göre sınıflandırma yapar. Deneylerimizde zayıf sınıflandırıcı olarak tek seviyeli karar ağacı kullanılmıştır [21].

MLP ileri beslemeli bir yapay sinir ağı algoritmasıdır. MLP en az 3 adet nöron katmanından oluşur. Girdi nöronları hariç diğer nöronlar doğrusal olmayan aktivasyon fonksiyonları kullanırlar. Weka MLP'de aktivasyon fonksiyonu olarak Sigmoid fonksiyonu kullanılır. Bu algoritma eğitimde geri beslemeli (hata düzeltimli) bir sinir ağı yapısı kullanır. Bu algoritmanın Weka konfigürasyonunda gizli katman sayısı "a" olarak belirtilmiştir. Bu şu anlama gelmektedir; 1 adet gizli katman vardır ve bu gizli katmandaki nöron sayısı (öznitelik sayısı + sınıf sayısı) / 2 kadardır. Dolayısıyla deneylerimizde MLP'de bir adet gizli katman kullanılmıştır ve bu yüzden sığ bir yapay sinir ağıdır. Ayrıca deneylerde kullanılan öznitelik sayısına göre gizli katman nöron sayısı da değişiklik göstermektedir. Doğrusal ayrıştırılamayan verileri sinir ağı yapısı sayesinde problemi alt parçalara bölerek ayrıştırabilir. Karmaşık problemler için iyi bir yöntem olarak kabul edilir. Fakat anlaşılması ve yorumlanması kolay değildir ayrıca yüksek işlem gücü ister. Literatürdeki kullanımının fazla oluşundan dolayı tercih edilmiştir.

Voted perceptron, doğrusal sınıflandırıcı perceptron algoritmasını kullanarak yüzeyler arası maksimum aralıklı sınıflandırma gerçekleştiren bir makine öğrenme algoritmasıdır. Bu algoritma çok boyutlu uzaylarda polinom kernel fonksiyonunu kullanarak sınıflandırma gerçekleştirebilir. SVM ile kıyaslandığında işlem süresi daha kısadır [22].

Bayesian network özniteliklerin birbirleriyle bağımlılıklarına göre bir DAG ağı oluşturup olasılık hesabı kullanılarak sınıflandırma yapan bir makine öğrenme algoritmasıdır. Bu algoritmada Weka K2 lokal arama algoritması basit bir kestirici ile birlikte kullanılmıştır [23].

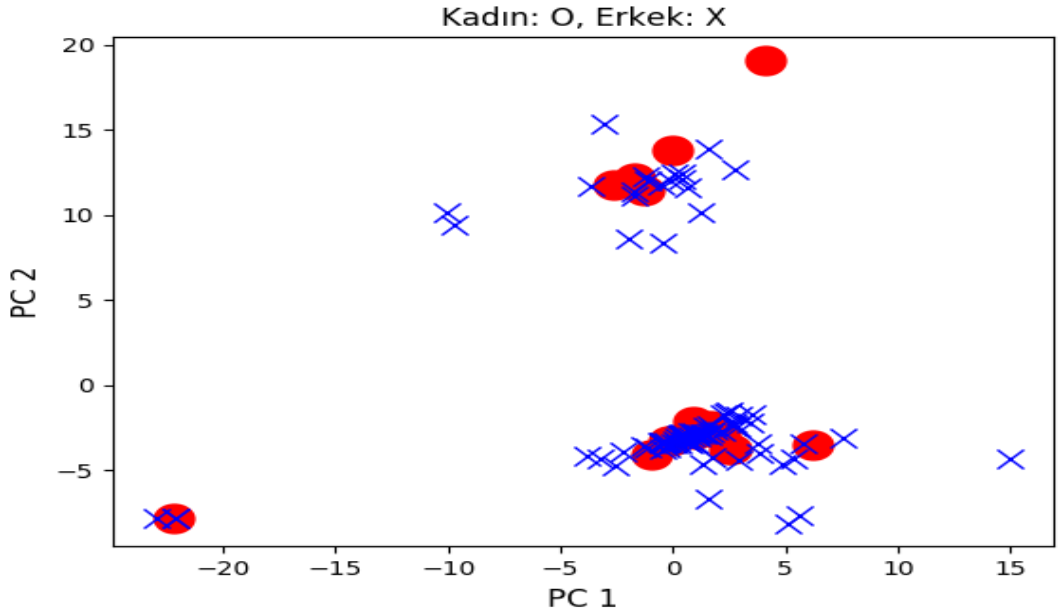
Tüm deneylerimizde Weka on katlamalı çapraz doğrulama tekniği kullanılmıştır.

4.3 Veri Ön İşlemesi

Öznitelik çıkarımından sonra daha ideal bir sınıflandırma için (i) sınıf veri sayısı dengeleme, (ii) öznitelik içerisinde gruplandırma, (iii) uzay boyutunu azaltma veri ön işleme süreçlerine ihtiyacımız vardır.

Uyanık veri kümesi sınıf veri sayıları kıyaslandığında oldukça dengesizdir. Kadın sürücü sayısı (17) erkek sürücü sayısına (88) göre oldukça düşüktür. Dengeli bir veri kümesi üzerinde sınıflandırma işlemi gerçekleştirmek daha ideal bir yaklaşım olacaktır. Bu yaklaşım çoğunlukla daha iyi ve güvenilir sınıflandırma doğruluk oranları elde etmek içindir. Veri dengesizliği sorununun üstesinden gelmek için bazı veri kopyalama/örnekleme teknikleri uygulamamız gerekmektedir. Bu araştırmada aşırı örnekleme ve SMOTE olmak üzere iki teknik düşünülmüştür. Aşırı örnekleme yöntemi, sınıfların örnek sayısı eşit olana dek azınlık sınıfın örneklerini kopyalayarak arttırmaya dayanır. Bu yöntem veri kümemizdeki 17 kadın sürücüden oluşan kadın azınlık sınıfına beş veya altı kez kopyalama yaparak uygulandığında, 88 erkek ve 88 kadın sürücü elde edilecektir. SMOTE yöntemi ise en yakın komşu tekniğini kullanarak azınlık sınıfı için yeni sentetik veriler oluşturur [24]. Bu yöntem veri setimize uygulandığında, 17 gerçek kadın sürücüden 61 yeni sentetik kadın sürücü üretilir. Herhangi iki sürücünün aynı rota boyunca aynı sinyalleri üretmediğini bildiğimizden dolayı, araştırmamızda aşırı örnekleme yöntemi yerine SMOTE yöntemi tercih edilmiştir. Tsfresh çıktısı sonucunda oluşan özniteliklerin kendi içlerinde gruplandırılması, karar ağaçları gibi bazı sınıflandırma algoritmaları için ideal bir yapı oluşturacaktır. Bu işlemi gerçekleştirmek için Weka gözetimli ayrıştırma filtresi kullanılmıştır. Bu filtre nominal olmayan sayısal öznitelikleri kendi içerisinde gruplara ayrıştırır. Bu yöntemle veri kümesi bazı sınıflandırma algoritmaları için daha kullanışlı bir hale gelir. Bu durum daha güçlü ayrıştırmalara sebep olarak doğruluk oranlarını artırabilir. Aynı zamanda bazı sınıflandırma algoritmalarında bu dahili ayrıştırma mevcuttur. Tsfresh ile tek bir CAN hattı verisinden 216 öznitelik üretebiliriz. Eğer iki CAN hattı verisini birlikte kullanırsak (örn. VS ve CS öznitelikleri) 432 (=216*2) öznitelik, 10 adet CAN hattı verisinin tamamını birlikte kullanırsak 2160 (=216*10) özniteliğe sahip oluruz. Ama çok az sayıda veri (105 sürücüden 176 veri) ve çok fazla miktarda özniteliğe sahip olmamız çok boyutluluğun lanetinden dolayı sorunlu bir durum yaratabilir. Bu yüzden öznitelik seçimi gibi uzay boyutunun azalmasını sağlayan yöntemlere ihtiyacımız vardır. Neyse ki, birçok sınıflandırma algoritması dahili öznitelik seçim tekniklerine sahiptir. Doğrudan öznitelik seçimi için Weka derecelendirme algoritmasının bilgi kazanım kriteriyle kullanımının iyi bir alternatif olacağını düşündük. Çünkü derecelendirme algoritması tüm öznitelikleri bireysel olarak sınıflandırılabilme yeteneğine göre derecelendirmek-

tedir. Tüm öznitelikler derecelendirildikten sonra, gereksiz olanları veri kümesinden kaldırabiliriz. Bu yöntemle sınıf tahmini için ayırt edici olmayan öznitelikler elenmiş olur. Sınıflandırma aşamasının detaylarına geçmeden önce, veri kümesinin cinsiyet bazında ayrıştırılabilir olup olmadığı ile ilgili bir veri analizi ön araştırması yaptık. Bu amaç doğrultusunda, bazı 216 öznitelik çıkarımı yapılan CAN veri tipleri üzerinde Weka iki boyutlu temel bileşen analizi gerçekleştirilmiştir. Şekil 4.2'de (CAN C hattı için) ve Şekil 4.3'de (CAN ERPM hattı için) iki boyutlu temel bileşen analizi yapılmış verilerin cinsiyetlerine göre dağılımları görülmektedir. Dağılım göstermektedir ki veri kümesi içerisinde bulunan bazı öbeklerin cinsiyetleri ayrıştırılabilir. Bu sonuçlar bizi daha detaylı bir sınıflandırma çalışması yapmamız konusunda cesaretlendirmiştir.



Şekil 4.2: CAN C hattı cinsiyet dağılımı.

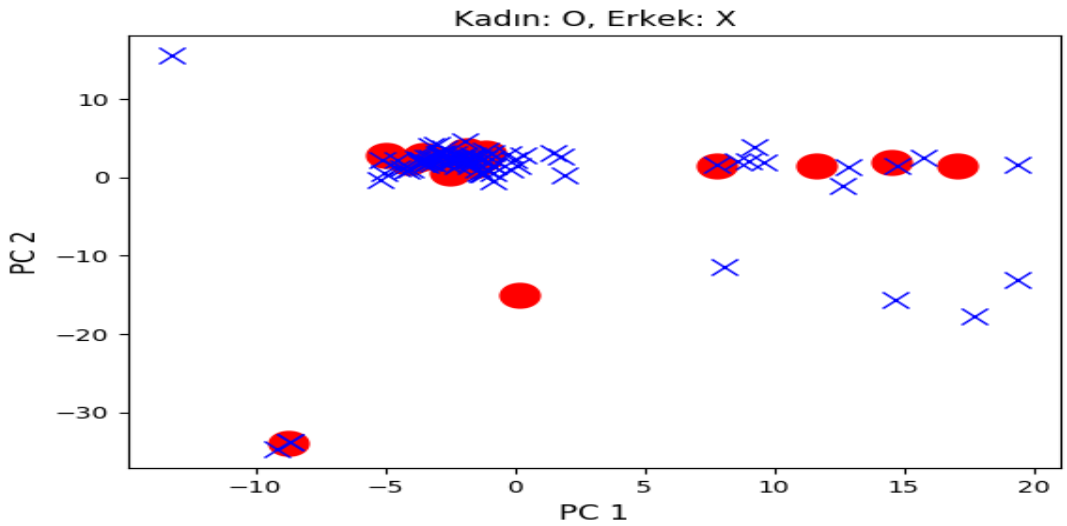
4.4 Deney Sonuçları ve Yorumlar

Veri sayısı dengesizliği veri kümemiz için önemli bir konudur. Sınıflara ait veri sayılarını dengeli bir seviyeye getirmek için tüm deneylerimizde SMOTE metodu uygulanmıştır.

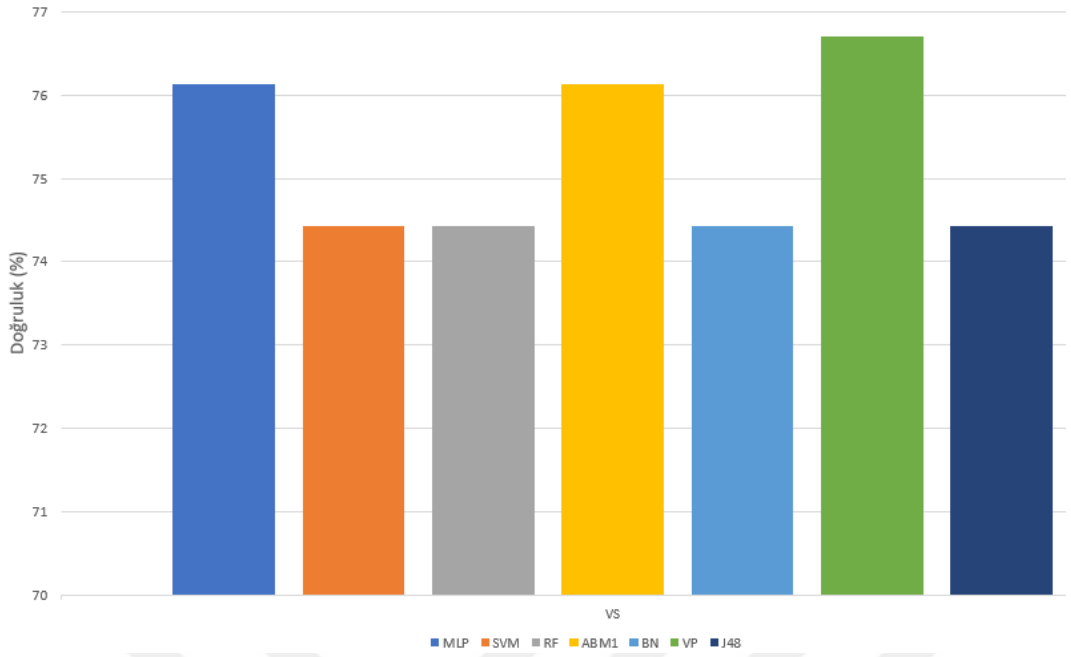
Başlangıç analizi olarak ayrıştırma işleminin sonuca etkisini inceledik. Bu metodun birçok CAN hattı veri tipi üzerinde oldukça etkili olduğunu gözlemledik. Örneğin, 216 öznitelik çıkarımı yapılmış BS CAN hattı verisine ayrıştırma işlemi uygulandığında doğruluk oranı 85.22% iken bu metot uygulanmadığında doğruluk oranı 70.45% ola-

rak ölçülmüştür. Bu nedenle, sadece ayrıştırma işleminin uygulandığı deneylerin sonuçlarını paylaştık. İlk deneyimizde öznitelik sayısındaki değişimin doğruluk oranına etkisini inceledik. Bunun için farklı miktarlarda öznitelik çıkarımları yaparak oluşturduğumuz veri kümeleri üzerinde sınıflandırma işlemini gerçekleştirdik.

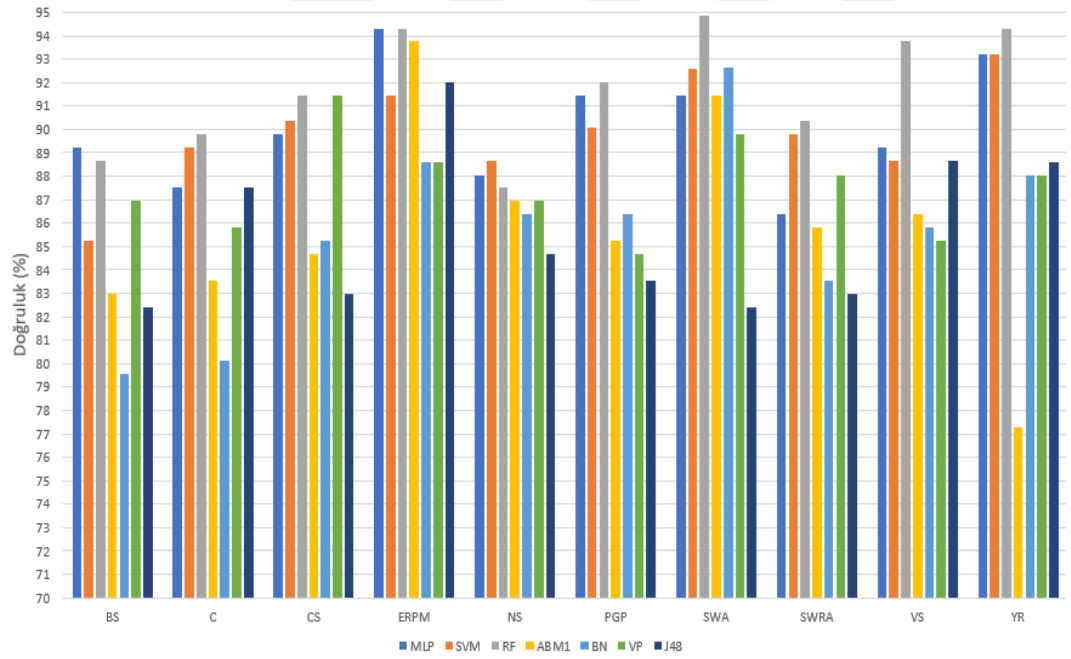
8 ve 216 özniteliğe sahip veri kümelerinin sınıflandırma sonuçlarını karşılaştırdık ve tüm CAN hattı veri tipleri için 8 öznitelik konfigürasyonu ile elde edilen sınıflandırma doğruluk oranının 216 öznitelik konfigürasyonu ile elde edilen sınıflandırma doğruluk oranına göre daha düşük olduğunu tespit ettik. Şekil 4.4 ve 4.5'da CAN VS verisinin sınıflandırma doğruluk oranları bu tespiti doğrular niteliktedir. VS veri tipi için, Şekil 4.4'de ki maksimum doğruluk oranı 77% oranını geçmezken, Şekil 4.5'da doğruluk oranı 93%'e ulaşmıştır. Bu sebepten dolayı, tüm CAN hattı veri tipleri için Tsfresh 216 öznitelik çıkarımını deneme kararı aldık. Şekil 4.5'da hemen hemen tüm sınıflandırma algoritmaları için doğruluk oranı nadiren 80%'nin altına düşmekte ve 90% civarında yer almaktadır. Şekil 4.5 tüm CAN hattı veri tipleri için tekli kombinasyonda sınıflandırma doğruluk oranlarını göstermektedir. Fakat biz iki farklı CAN hattı veri tipinin özniteliklerinin birleştirilmesinin (216 + 216 = 432) sınıflandırma doğruluk oranlarının geliştirilmesine yardımcı olabileceğini düşündük. Bu sebepten ötürü, Şekil 4.6'da gösterilen ve beş farklı CAN veri tipi çiftinin kullanıldığı sınıflandırma deneyini gerçekleştirdik. Şekil 4.5'daki sonuçlarla kıyasladığımızda, neredeyse tüm sınıflandırma algoritmalarının doğruluk oranlarında artma eğilimi olduğunu tespit ettik. Bunun üzerine daha da ileri giderek, tüm CAN hattı veri tiplerini (216 * 10 = 2160) birlikte kullanarak Şekil 4.7'de gösterilen deneyi gerçekleştirdik. En iyi sınıflandırıcı 97% doğruluk oranına ulaştı ve genel olarak önceki deneylere göre doğruluk oranında artış tespit edildi.



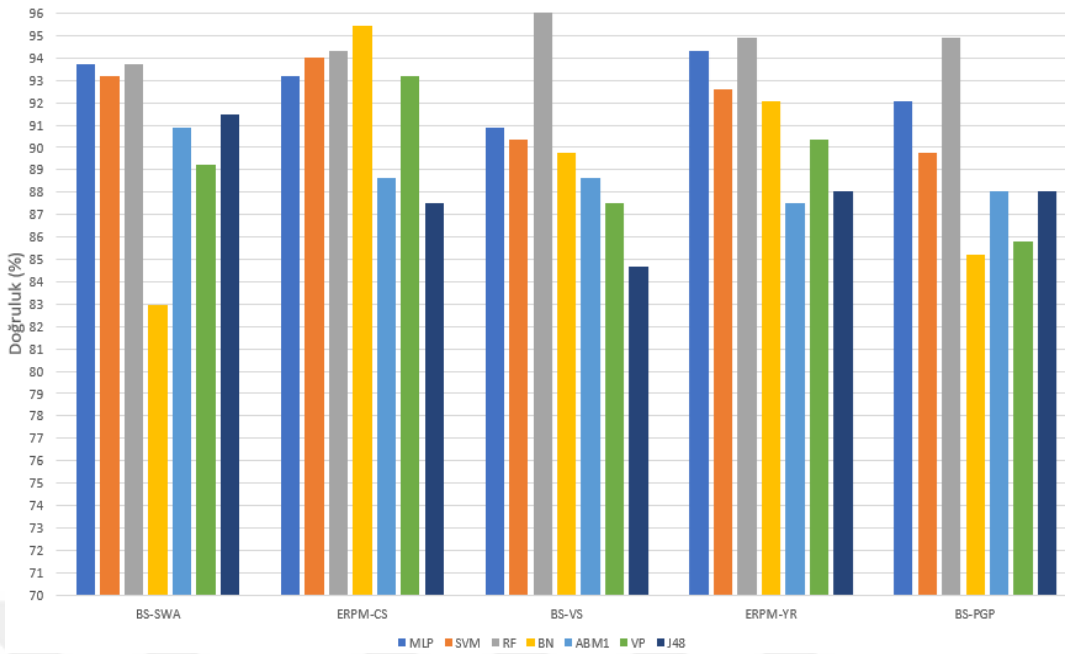
Şekil 4.3: CAN ERPM hattı cinsiyet dağılımı.



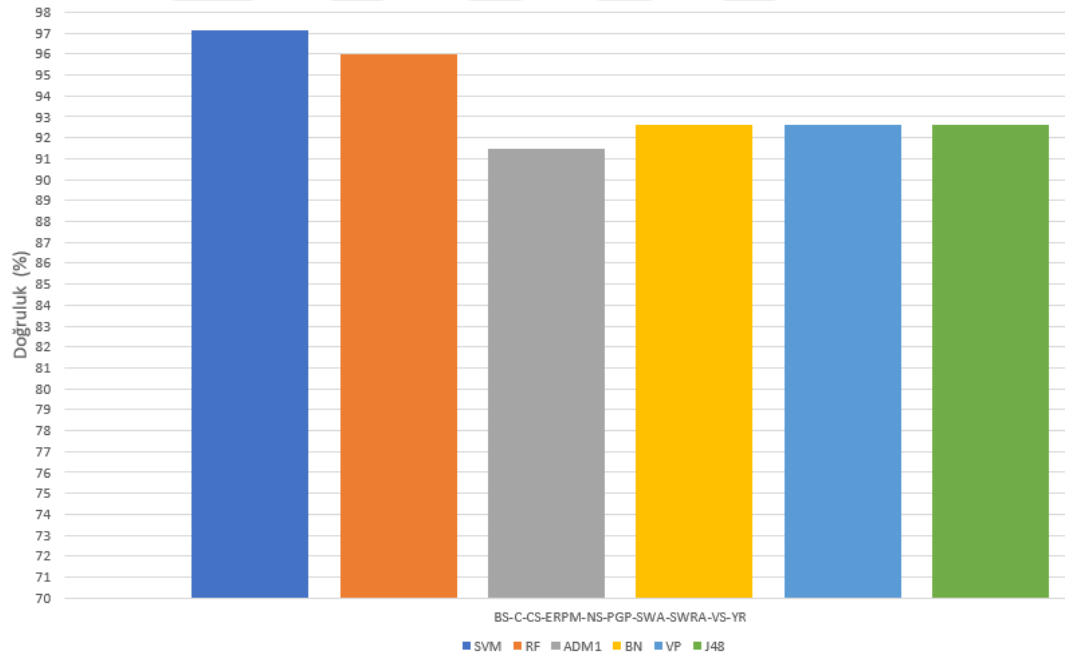
Şekil 4.4: 8 öznitelikli-SMOTE ve ayrıştırma filtresi uygulanan deneyin doğruluk oranları.



Şekil 4.5: 216 öznitelikli-SMOTE ve ayrıştırma filtresi uygulanan deneyin doğruluk oranları.



Şekil 4.6: 432 öznitelikli-SMOTE ve ayrıştırma filtresi uygulanan deneyin doğruluk oranları.



Şekil 4.7: 2160 öznitelikli-SMOTE ve ayrıştırma filtresi uygulanan deneyin doğruluk oranları.

Sınıflandırmada temel performans kıstasımız doğruluk oranıdır. Fakat aynı zamanda TP, FP, Precision, Recall, F−measure ve ROC metrikleri de deneylerimizde hesaplanmıştır. Sırasıyla Çizelge 4.2, Çizelge 4.3 ve Çizelge 4.4'de tekli, ikili ve onlu CAN hattı veri tipi kombinasyonları için en iyi sonuç veren sınıflandırma algoritmaları ve sonuç parametreleri gösterilmektedir. Bu sonuçlar elde edilen yüksek doğruluk oranlarıyla uyumludur.

Tüm bu sonuçlar incelendiğinde ve genel ortalama dikkate alındığında, RF, MLP ve SVM sınıflandırma algoritmalarının diğer algoritmalara göre daha iyi performans gösterdiğine karar verdik. Çizelge 4.2, Çizelge 4.3 ve Çizelge 4.4'de RF, MLP ve SVM algoritmalarının yoğun bir şekilde yer alması bu durumu doğrulamaktadır. Ayrıca yine deney sonuçlarımıza göre, en kötü performans gösteren iki algoritma BN ve ABM1'dir.

Çizelge 4.2: Tekli veri kombinasyonlarının çeşitli metriklere göre en iyi sonuçları.

Veri Tipi, En iyi Alg.	**TP Oranı**	**FP Oranı**	**Precision**	**Recall**	**F−Measure**	**ROC Alanı**
BS, MLP	0.892	0.108	0.893	0.892	0.892	0.944
C, RF	0.898	0.102	0.901	0.898	0.898	0.940
CS, RF	0.915	0.085	0.924	0.915	0.914	0.958
NS, RF	0.886	0.114	0.887	0.886	0.886	0.886
PGP, RF	0.920	0.080	0.921	0.920	0.920	0.973
RPM, RF	0.943	0.057	0.943	0.943	0.943	0.985
SWA, RF	0.949	0.051	0.949	0.949	0.949	0.981
SWRS, RF	0.903	0.097	0.903	0.903	0.903	0.949
VS, RF	0.938	0.063	0.938	0.938	0.937	0.984
YR, RF	0.943	0.057	0.944	0.943	0.943	0.976

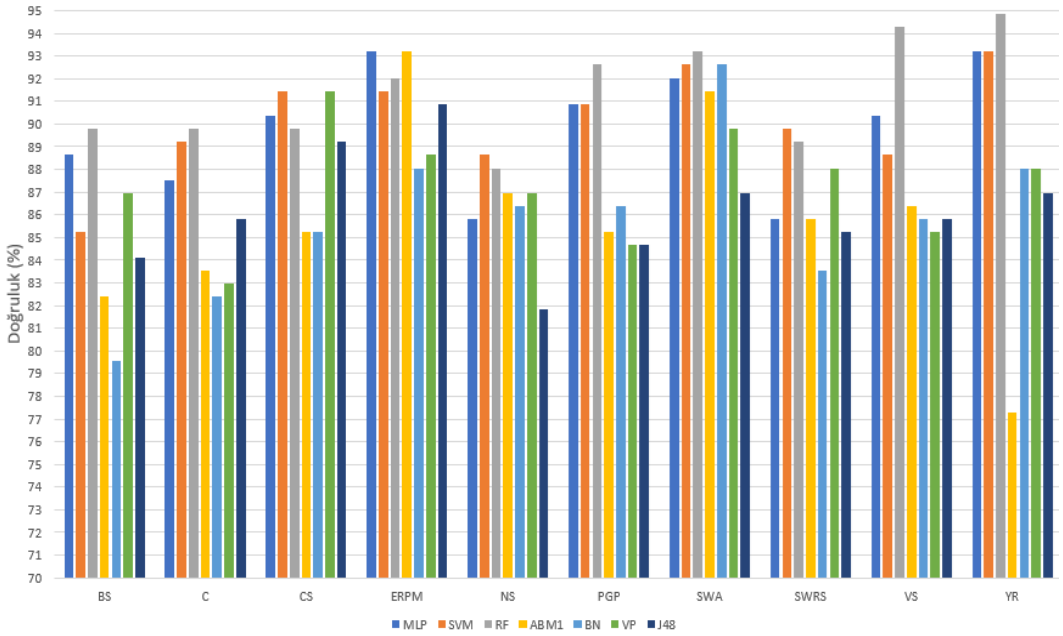
Çizelge 4.3: İkili veri kombinasyonlarının çeşitli metriklere göre en iyi sonuçları.

Veri Tipi, En iyi Alg.	**TP Oranı**	**FP Oranı**	**Precision**	**Recall**	**F−Measure**	**ROC Alanı**
BS−*PGP*, RF	0.949	0.051	0.949	0.949	0.949	0.990
BS−*SWA*, RF	0.938	0.063	0.939	0.938	0.937	0.982
BS−*VS*, RF	0.960	0.040	0.960	0.960	0.960	0.983
RPM−*CS*, BN	0.955	0.045	0.955	0.955	0.955	0.980
RPM−*YR*, RF	0.949	0.051	0.949	0.949	0.949	0.985

Çizelge 4.4: Onlu veri kombinasyonlarının çeşitli metriklere göre en iyi sonuçları.

Veri Tipi, En iyi Alg.	**TP Oranı**	**FP Oranı**	**Precision**	**Recall**	**F−Measure**	**ROC Alanı**
Tüm veriler, SVM	0.972	0.028	0.972	0.972	0.972	0.972

İkinci deneyimizde; Weka derecelendirme algoritmasını bilgi kazanım kriteri ile birlikte, her biri için 216 öznitelik çıkarımı yapılmış 10 adet CAN hattı verisine uyguladık. Tüm öznitelikler derecelendirildikten sonra, gereksiz olanları veri kümesinden kaldırdık. Bu işlemin sonucunda, her bir CAN hattı verisi için geriye kalan öznitelik sayısı Çizelge 4.5'da verilmiştir. Oluşan yeni veri kümesine, Weka gözetimli ayrıştırma filtresi uygulanmış ve veri sınıflandırılmıştır. Şekil 4.5 ile Şekil 4.8 karşılaştırıldığında, her iki grafiğinde üst sınırlarının aynı olduğu görülebilir. Buna ek olarak, her bir CAN hattı verisi için sınıflandırma doğruluk oranının benzer olduğu görülmüştür. Çizelge 4.6'da ki sonuçlar ile bu değerlerdirme uyumluluk göstermektedir. Sonuç olarak, bilgi kazanım öznitelik seçim işlemi sınıflandırma doğruluk oranında önemli bir fark oluşturmamıştır.



Şekil 4.8: 216 öznitelikli-SMOTE, ayrıştırma filtresi ve bilgi kazanım öznitelik seçimi uygulanan deneyin doğruluk oranları.

Çizelge 4.5: Her bir CAN hattı verisine bilgi kazanım öznitelik seçim işlemi uygulanması ile oluşan yeni öznitelik sayıları.

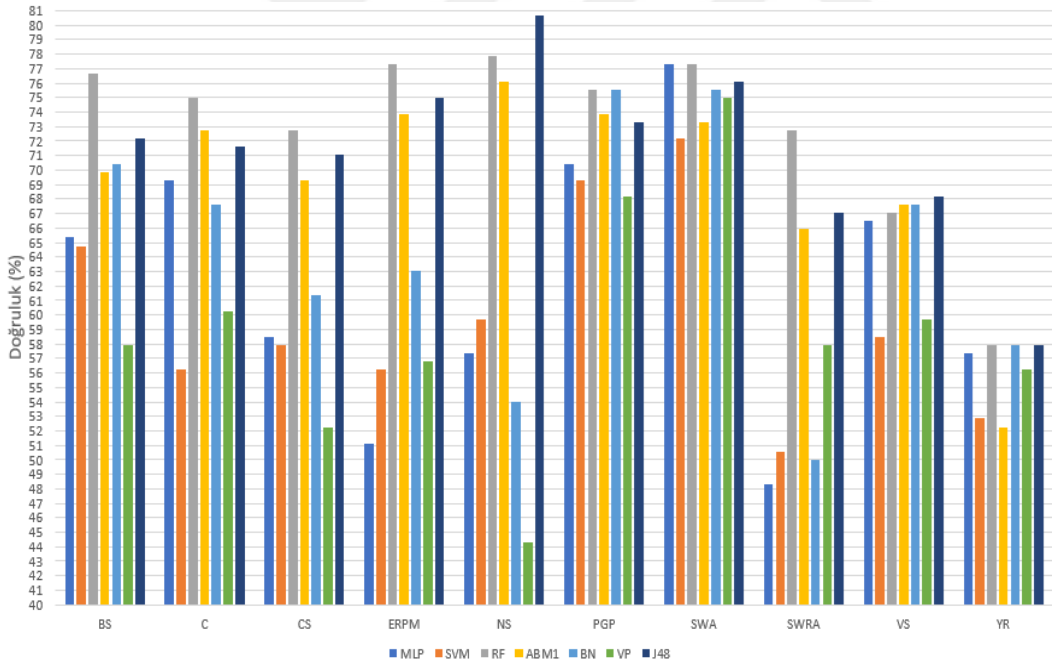
	BS	C	CS	NS	PGP	RPM	SWA	SWRS	VS	YR
Öznitelik Sayıları	91	47	65	91	62	83	58	61	51	71

Üçüncü deneyimizde; Çizelge 4.5'da öznitelik sayıları gösterilen veri kümesine Weka gözetimsiz PCA işlemi uygulanarak her bir CAN hattı verisi için iki boyutlu yeni

bir uzay yaratılmıştır ve bu oluşan yeni yapıda ki veri kümesi üzerinde sınıflandırma işlemi gerçekleştirilmiştir. Şekil 4.5 ile Şekil 4.9 karşılaştırıldığında, Şekil 4.9'da gösterilen sınıflandırma doğruluk oranı bariz bir şekilde daha kötüdür. Çizelge 4.7'de ki sonuçlar ile bu değerlendirme uyumluluk göstermektedir. Sonuç olarak, PCA öznitelik seçim işlemi sınıflandırma doğruluk oranını düşürmektedir.

Çizelge 4.6: Bilgi kazanım öznitelik seçimi uygulanmış tekli veri kombinasyonlarının çeşitli metriklere göre en iyi sonuçları.

Veri Tipi, En iyi Alg.	TP Oranı	FP Oranı	Precision	Recall	F−Measure	ROC Alanı
BS, RF	0898	0.102	0.900	0.898	0.898	0.945
C, RF	0.898	0.102	0.900	0.898	0.898	0.942
CS, SVM	0.915	0.085	0.921	0.915	0.914	0.915
NS, SVM	0.886	0.114	0.887	0.886	0.886	0.886
PGP, RF	0.926	0.074	0.926	0.926	0.926	0.976
RPM, MLP	0.932	0.068	0.934	0.932	0.932	0.983
SWA, RF	0.932	0.068	0.932	0.932	0.932	0.976
SWRS, SVM	0.898	0.102	0.898	0.898	0.898	0.898
VS, RF	0.943	0.057	0.943	0.943	0.943	0.984
YR, RF	0.949	0.051	0.949	0.949	0.949	0.973



Şekil 4.9: 216 öznitelikli-SMOTE, ayrıştırma filtresi ve PCA öznitelik seçimi uygulanan deneyin doğruluk oranları.

Dördüncü deneyimizde; Veri kümesindeki veri dengesizliği problemini gidermek amacıyla veri kümesine SMOTE yöntemi yerine aşırı örnekleme yöntemi uygulanmıştır ve bu veriler üzerinde sınıflandırma işlemi gerçekleştirilmiştir. Şekil 4.5 ile Şekil 4.10 karşılaştırıldığında, Şekil 4.10'da gösterilen doğruluk oranı sonuçlarının ortalama hesabına göre daha yüksek çıktığı söylenebilir.Bahsi geçen aşırı örnekleme deneyi 94.48%, SMOTE deneyi ise 91.87% ortalama doğruluk oranlarına sahiptirler. Fakat aradaki fark kritik bir boyutta değildir ve daha önceden de bahsedildiği üzere aşırı örnekleme metodu çalışma prensibi açısından gerçekliğe yakın olmadığından dolayı güvenilir bir seçenek olmayabilir. Çizelge 4.8'de ki sonuçlar ile de bu değerlendirmeler uyumluluk göstermektedir.

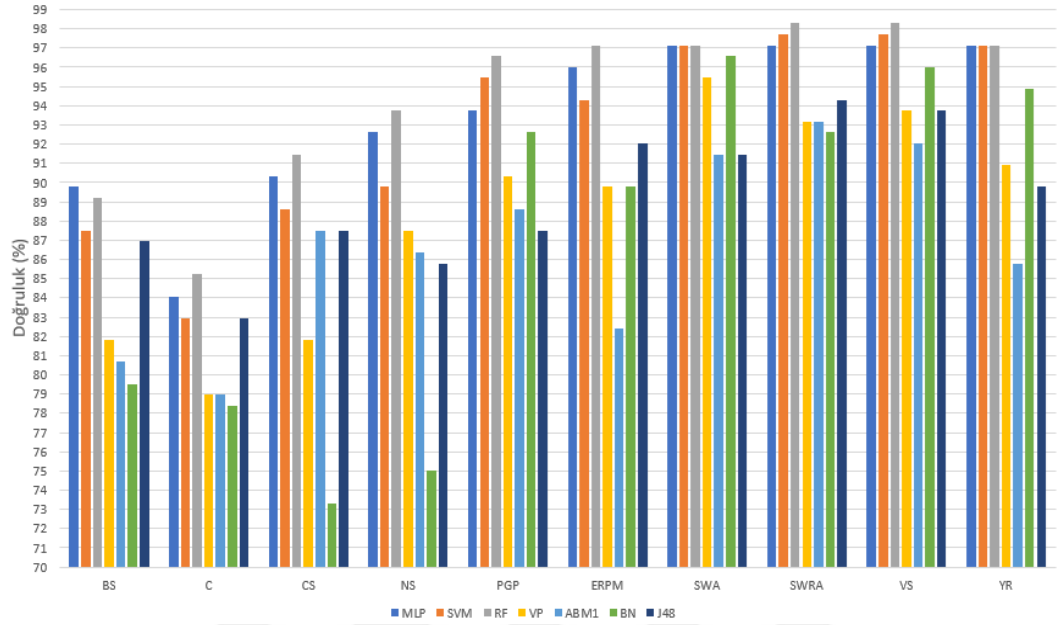
Çizelge 4.7: PCA öznitelik seçimi uygulanmış tekli veri kombinasyonlarının çeşitli metriklere göre en iyi sonuçları.

Veri Tipi, En iyi Alg.	**TP Oranı**	**FP Oranı**	**Precision**	**Recall**	**F−Measure**	**ROC Alanı**
BS, RF	0.767	0.233	0.767	0.767	0.767	0.861
C, RF	0.750	0.250	0.755	0.750	0.749	0.810
CS, RF	0.727	0.273	0.729	0.727	0.727	0.815
NS, J48	0.807	0.193	0.815	0.807	0.806	0.812
PGP, RF	0.756	0.244	0.757	0.756	0.755	0.853
RPM, RF	0.773	0.227	0.773	0.773	0.773	0.819
SWA, RF	0.773	0.227	0.773	0.773	0.773	0.832
SWRS, RF	0.727	0.273	0.727	0.727	0.727	0.769
VS, J48	0.682	0.318	0.702	0.682	0.674	0.633
YR, RF	0.580	0.420	0.580	0.580	0.580	0.614

4.5 Tartışma

Cinsiyet sınıflandırma araştırmamızda dört adet deney gerçekleştirilmiştir. İlk deneyimizde, öznitelik sayısını arttırmanın doğruluk oranını arttırdığı gözlemlenmiştir. İkinci deneyimizde, bilgi kazanım yöntemi kullanılarak öznitelik seçimi yaptığımız durum ile yapmadığımız durum arasında doğruluk oranı açısından önemli bir fark olmadığı görülmüştür. Üçüncü deneyimizde, PCA işlemi uygulayarak iki boyuta düşürdüğümüz veri uzayı üzerinde gerçekleştirilen sınıflandırmaların doğruluk oranlarının gerçekleştirilmeyenlere göre oldukça düşük olduğu tespit edilmiştir. Son deneyimizde ise SMOTE veri çoğaltma yöntemi yerine aşırı örnekleme yöntemi kullanıldığında ortalama doğruluk oranının artığı gözlemlenmiştir. Fakat bu yöntemin gerçek duruma yakın olmayan veri üretim metodu sebebiyle ilk deneyimizde kullanılmamasına karar verilmiştir. Tüm bu deneyler incelendiğinde SMOTE yöntemi ile birlikte ayrıştırma filtresi uygulanan ve öznitelik sayısı yüksek olan veri kümelerinde cinsiyet sınıflan-

dırma doğruluk oranının diğer deney konfigürasyonlarına göre daha iyi sonuç vereceği değerlendirilmektedir.



Şekil 4.10: 216 öznitelikli-aşırı örnekleme ve ayrıştırma filtresi uygulanan deneyin doğruluk oranları.

Çizelge 4.8: Aşırı örnekleme işlemi uygulanmış tekli veri kombinasyonlarının çeşitli metriklere göre en iyi sonuçları.

Veri Tipi, En iyi Alg.	**TP Oranı**	**FP Oranı**	**Precision**	**Recall**	**F−Measure**	**ROC Alanı**
BS, MLP	0.898	0.102	0.911	0.898	0.897	0.898
C, RF	0.852	0.148	0.881	0.852	0.849	0.889
CS, RF	0.915	0.085	0.927	0.915	0.914	0.954
NS, RF	0.938	0.063	0.944	0.938	0.937	0.959
PGP, RF	0.966	0.034	0.968	0.966	0.966	0.997
RPM, RF	0.972	0.028	0.973	0.972	0.972	0.998
SWA, RF	0.972	0.028	0.973	0.972	0.972	0.998
SWRS, RF	0.983	0.017	0.984	0.983	0.983	0.998
VS, RF	0.983	0.017	0.984	0.983	0.983	0.998
YR, RF	0.972	0.028	0.973	0.972	0.972	0.998

5. SÜRÜCÜ TANIMA

Bu araştırma sürücü CAN hattı verileri üzerinden sürücüyü tespit etmeyi amaçlamaktadır. Bu kısımda, gerçekleştirdiğimiz sürücü tanıma deneyleri için gerekli olan veri ön işleme süreçleri anlatılmış ve bu deneylerin sonuçları analiz edilmiştir. Ayrıca sürücü tanıma işleminin bir sonucu olarak; uygulanması konusunda görüş belirttiğimiz kişisel veri mahremiyeti konusuna da yer verilmiştir.

5.1 Kişisel Veri Mahremiyeti

"Kişisel veri, kimliği belirli veya belirlenebilir gerçek kişiye ilişkin her türlü bilgiyi ifade etmektedir. Bu bağlamda sadece bireyin adı, soyadı, doğum tarihi ve doğum yeri gibi onun kesin teşhisini sağlayan bilgiler değil, aynı zamanda kişinin fiziki, ailevi, ekonomik, sosyal ve sair özelliklerine ilişkin bilgiler de kişisel veridir. Bir kişinin belirli veya belirlenebilir olması, mevcut verilerin herhangi bir şekilde bir gerçek kişiyle ilişkilendirilmesi suretiyle, o kişinin tanımlanabilir hale getirilmesini ifade eder. Yani verilerin; kişinin fiziksel, ekonomik, kültürel, sosyal veya psikolojik kimliğini ifade eden somut bir içerik taşıması veya kimlik, vergi, sigorta numarası gibi herhangi bir kayıtla ilişkilendirilmesi sonucunda kişinin belirlenmesini sağlayan tüm halleri kapsar. İsim, telefon numarası, motorlu taşıt plakası, sosyal güvenlik numarası, pasaport numarası, özgeçmiş, resim, görüntü ve ses kayıtları, parmak izleri, genetik bilgiler gibi veriler dolaylı da olsa kişiyi belirlenebilir kılabilme özellikleri nedeniyle kişisel verilerdir" [25]. Dolayısıyla araç CAN hattı verileride kişisel veri olarak değerlendirilebilir. Çünkü araştırmamızda da bahsettiğimiz üzere bu verilerin sürücüyü belirlenebilir kılabilme özelliği vardır. Bu veriler hukuken bu kapsam içerisinde işlem görmelidir.

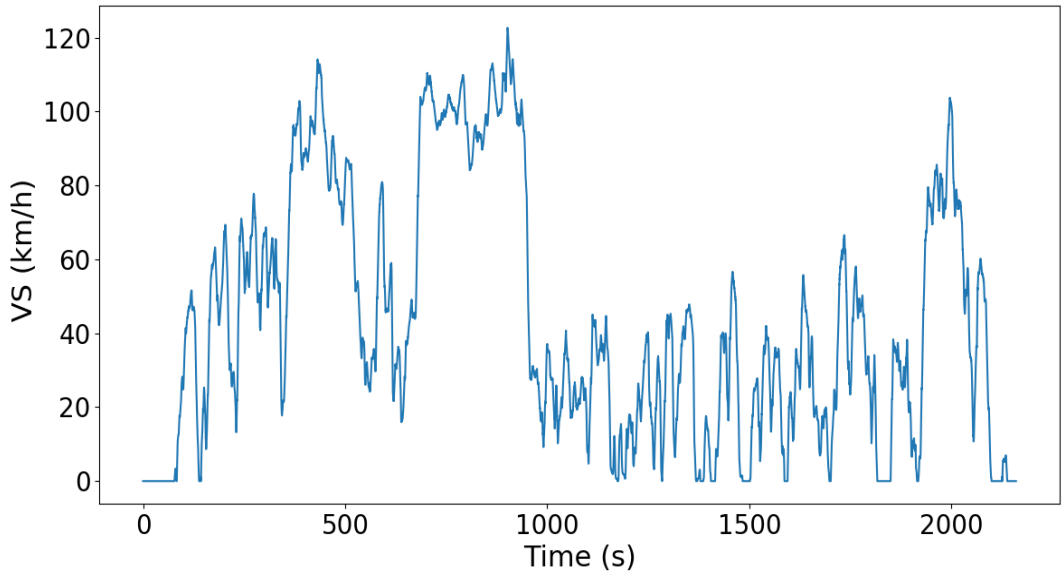
5.2 Veri Ön İşlemesi

Cinsiyet tahmininden farklı olarak bu deneyde her bir sürücü bir sınıfı temsil etmektedir. Fakat veri kümesinde her bir sürücü aynı rotada sadece bir defa sürüş yapmıştır. Dolayısıyla öznitelik çıkarımı yaptığımızda her bir sürücü için sadece bir adet örüntü olacaktır. Bu sınıflandırma için ideal bir durum değildir. Bunu ortadan kaldırmak amacıyla her bir sürücünün sürüş verisi eşit büyüklükte parçalara bölünmüştür ve her parça o sürücü için kendisini tanımlayan yeni bir örüntü oluşturmuştur. Bu işlem sonucunda bir sürücü bölündüğü parça sayısı kadar veri kümesinde ifade edilir. Sürücülerin toplam sürüş süreleri değişiklik göstermektedir. Dolayısıyla her bir sürücü için farklı miktarlarda kayıt altına alınan CAN hattı

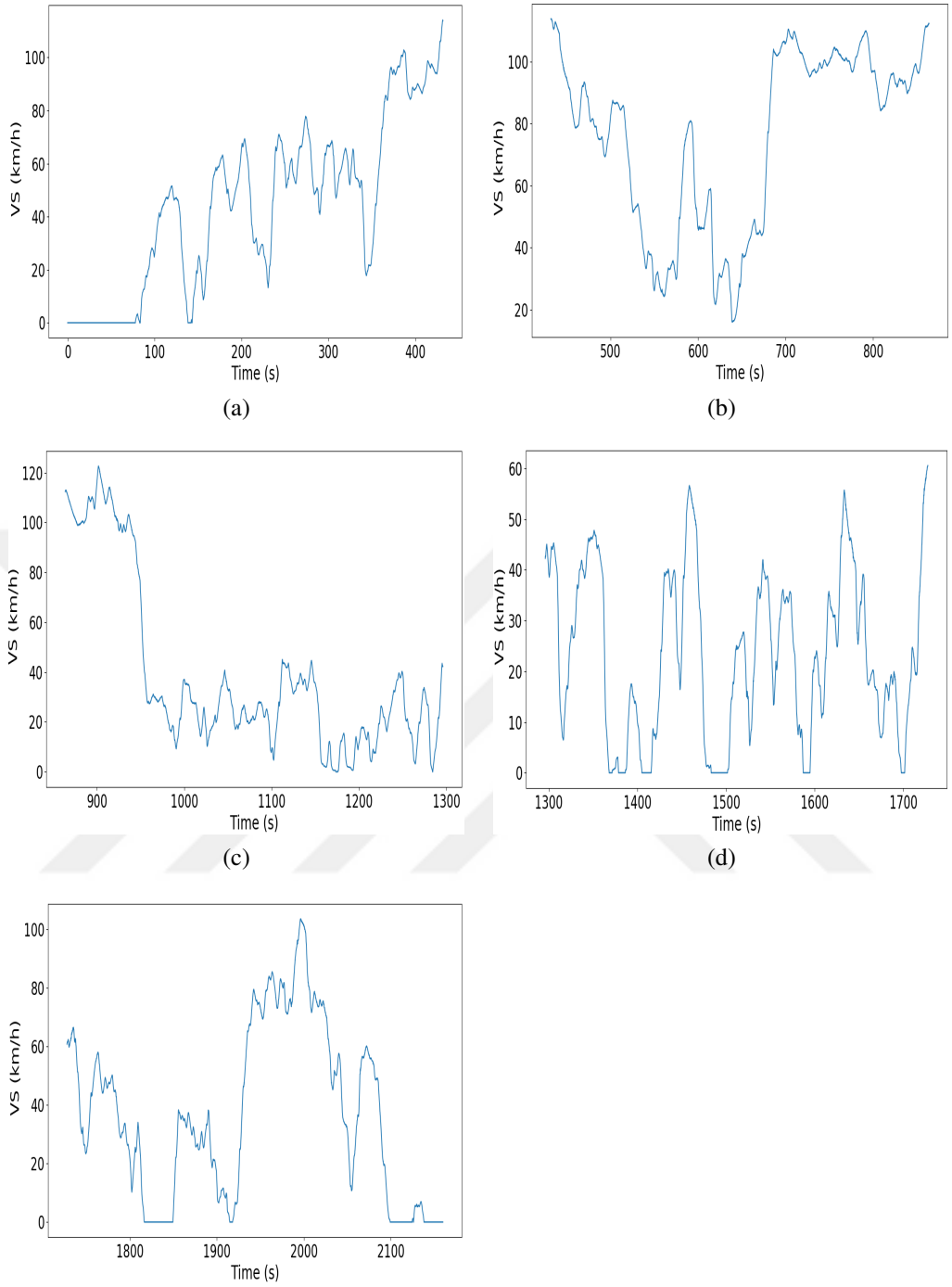
verisi bulunmaktadır. Bu yüzden verileri parçalara böldüğümüzde parçalardaki veri miktarı sürücüye göre değişiklik göstermektedir. Bu durum sürücü örüntüleri için fazla karakteristik öznitelikler oluşturmaktadır ve doğruluk oranı olması gerekenden çok daha yüksek seviyelere çıkmaktadır. Bu durumun sebebinin ise veri parçalama metodolojimiz olduğu ve cinsiyet sınıflandırma deneyinde bu özniteliklerin ayırt edicilik seviyesinin bu oranlarda olmadığı tespit edilmiştir. Bu durumu normalleştirmek ve daha güvenilir bir hale getirmek amacıyla, öznitelik çıkarım aşamasında veri uzunluğu ile direk ilgili olan matematiksel fonksiyonların deneylerden hariç tutulmasına karar verilmiştir.

Deneylerimizde sınıflandırma algoritması olarak cinsiyet sınıflandırması deneylerinde doğruluk oranı yüksek olan RF ve SVM algoritmaları kullanılmıştır. Öznitelik çıkarımı aşamasında ise 216 öznitelik konfigürasyonu içerisinden uzunluktan bağımsız olanlar seçilerek çıkarım gerçekleştirilmiştir. 8 öznitelik konfigürasyonunun denenmemesinin sebebi cinsiyet sınıflandırma deneyinde doğruluk oranının düşük olduğunun bilinmesidir.

Şekil 5.1'de Erkek-2003 sürücüsünün ham VS zaman serisi verisi grafik olarak gösterilmiştir. Şekil 5.2'de ise bu zaman serisi verisinin eşit büyüklükte 5 parçaya bölünmesi ile oluşan örüntüler gösterilmiştir. Burada her bir parça Erkek-2003 sürücüsünü tanımlamaktadır. Bu işlem ile Erkek-2003 sürücüsü veri kümesinde artık 5 parça ile temsil edilebilir hale gelmektedir.



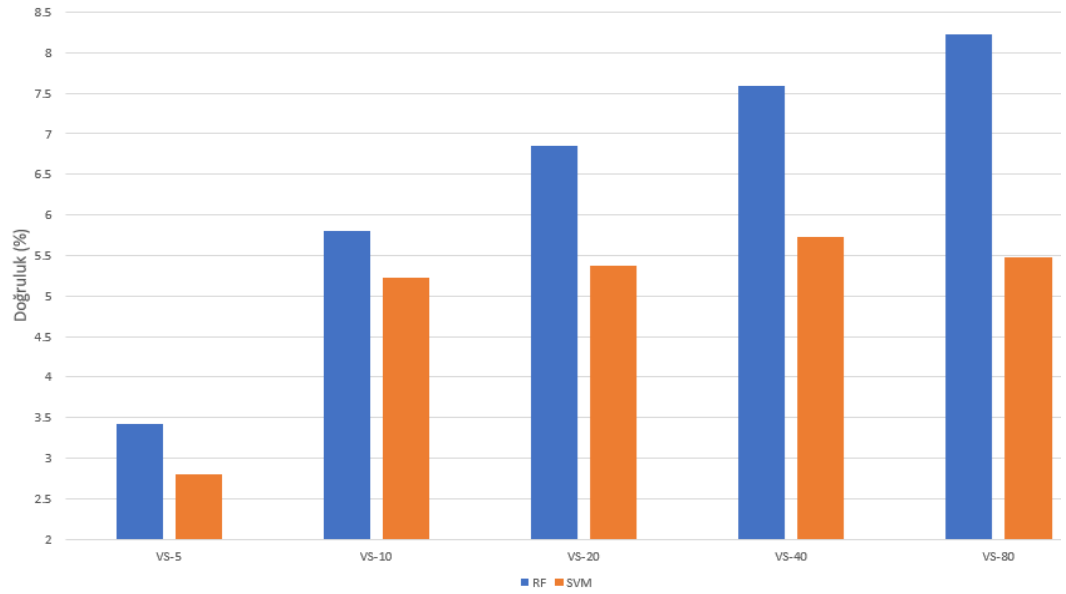
Şekil 5.1: Erkek-2003 VS zaman serisi.



Şekil 5.2: (a) Erkek-2003 parça 1 VS zaman serisi, (b) Erkek-2003 parça 2 VS zaman serisi, (c) Erkek-2003 parça 3 VS zaman serisi, (d) Erkek-2003 parça 4 VS zaman serisi, (e) Erkek-2003 parça 5 VS zaman serisi.

5.3 Deney Sonuçları ve Yorumlar

İlk deneyimizi, veri kümesini farklı sayılarda parçalara böldüğümüzde ortaya çıkan değişimi gözlemlemek amacıyla yaptık. Bu deneyde 105 adet sürücünün verileri 5, 10, 20, 40 ve 80 olmak üzere 5 farklı sayıda parçaya bölündü ve bu farklı konfigürasyondaki veri kümelerinin VS CAN hattı verisi üzerinden öznitelik çıkarım işlemi gerçekleştirdik. Bu işlem sonucunda uzunluktan bağımsız olarak seçilen 136 adet öznitelik ortaya çıkmıştır. Bu deneyin sonuçları Şekil 5.3 gösterilmiştir. Sonuçlar incelendiğinde verileri daha fazla parçaya bölerek sürücülere ait örüntü sayısının arttırılmasının doğruluk oranını arttırdığı tespit edilmiştir. Buradan da anlaşılmaktadır ki sürücüye ait veriyi arttırmak makine öğrenme prensipleri gereği sürücü tahminini geliştirmektedir. Çizelge 5.1 bu çıkarımı doğrular niteliktedir.

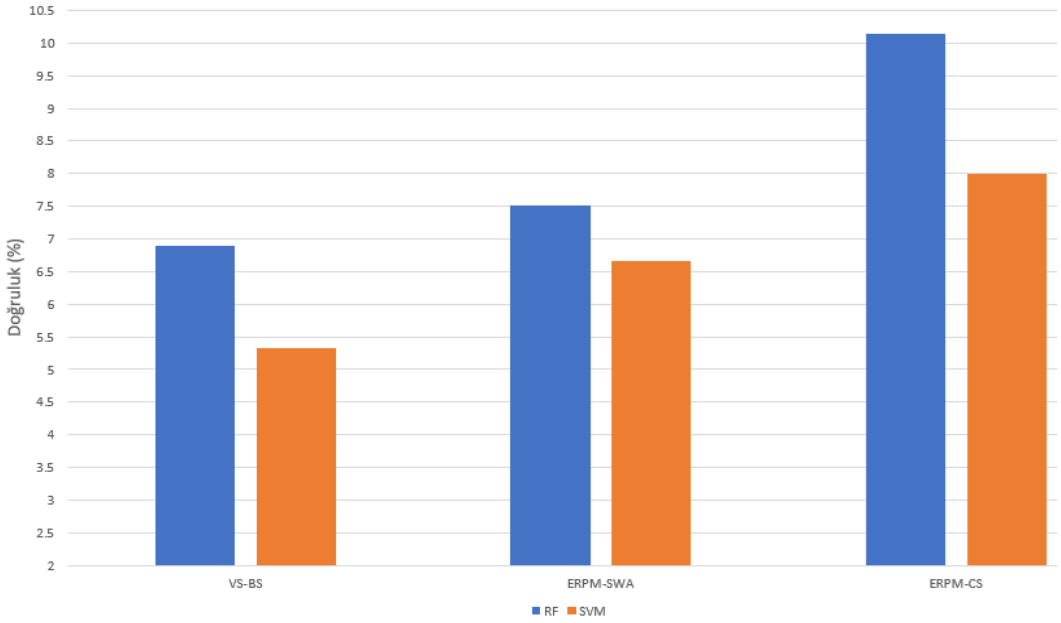


Şekil 5.3: Sürücü veri bölme sayısının doğruluk oranına etkisinin incelendiği deneyin sonuçları.

Çizelge 5.1: Sürücü veri bölme sayısının doğruluk oranına etkisinin incelendiği deneylerin çeşitli metriklere göre en iyi sonuçları.

Veri Tipi, En iyi Alg.	TP Oranı	FP Oranı	Precision	Recall	F−Measure	ROC Alanı
VS-5, RF	0.034	0.009	0.035	0.034	0.034	0.521
VS-10, RF	0.058	0.009	0.054	0.058	0.053	0.602
VS-20, RF	0.069	0.009	0.069	0.069	0.066	0.647
VS-40, RF	0.076	0.009	0.077	0.076	0.074	0.673
VS-80, RF	0.082	0.009	0.088	0.082	0.082	0.680

Cinsiyet sınıflandırma deneyinde iki farklı CAN hattı veri tipinin özniteliklerinin birleştirilmesinin doğruluk oranını arttırdığını gözlemlemiştik. Benzer bir sonucun sürücü tanıma içinde geçerli olup olmadığını tespit etmek için cinsiyet sınıflandırma deneyinde en yüksek doğruluk oranı veren 3 adet CAN veri tipi çifti üzerinde ikinci sürücü tanıma deneyimizi gerçekleştirdik. Bu deneyde 105 adet sürücünün verileri 20 eşit parçaya bölünmüştür ve öznitelik çıkarım işlemi gerçekleştirilmiştir. Bu işlem sonucunda uzunluktan bağımsız 271 adet öznitelik ortaya çıkmıştır. Bu deneyin sonuçları Şekil 5.4'de gösterilmiştir. Sonuçlar incelendiğinde iki farklı CAN hattı veri tipinin birleşimi sonucunda ortaya çıkan doğruluk oranlarının, Şekil 5.3'de verilerin 20 parçaya bölündüğü deneyin doğruluk oranlarına göre daha yüksek olduğu görülmektedir. Buradan da anlaşılmaktadır ki CAN hattı verilerini birleştirerek öznitelik sayısını arttırmak doğruluk oranını arttırmaktadır. Çizelge 5.2'de temel kıstasımız olan doğruluk oranı ile birlikte diğer sonuç parametreleri ile de deney sonucu ifade edilmektedir.

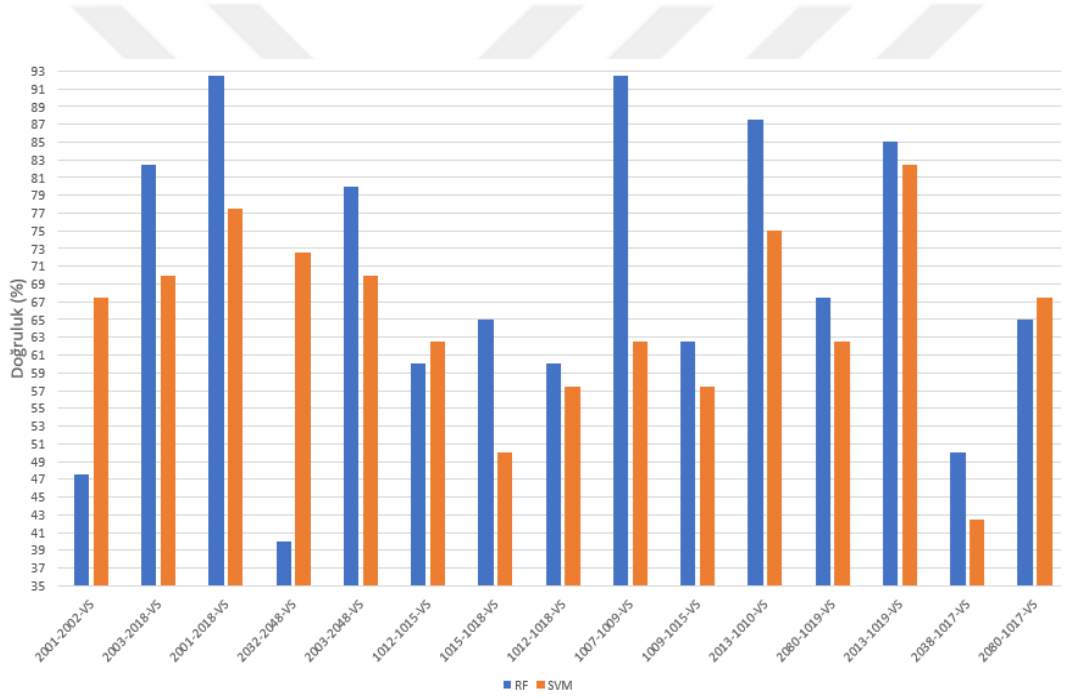


Şekil 5.4: İki adet CAN hattı verisi birleşiminin doğruluk oranına etkisinin incelendiği deneyin sonuçları.

Çizelge 5.2: İki adet CAN hattı verisi birleşiminin doğruluk oranına etkisinin incelendiği deneylerin çeşitli metriklere göre en iyi sonuçları.

Veri Tipi, En iyi Alg.	**TP Oranı**	**FP Oranı**	**Precision**	**Recall**	**F−Measure**	**ROC Alanı**
VS-BS, RF	0.069	0.009	0.074	0.069	0.068	0.631
ERPM-SWA, RF	0.075	0.009	0.080	0.075	0.074	0.676
ERPM-CS, RF	0.101	0.009	0.099	0.101	0.097	0.705

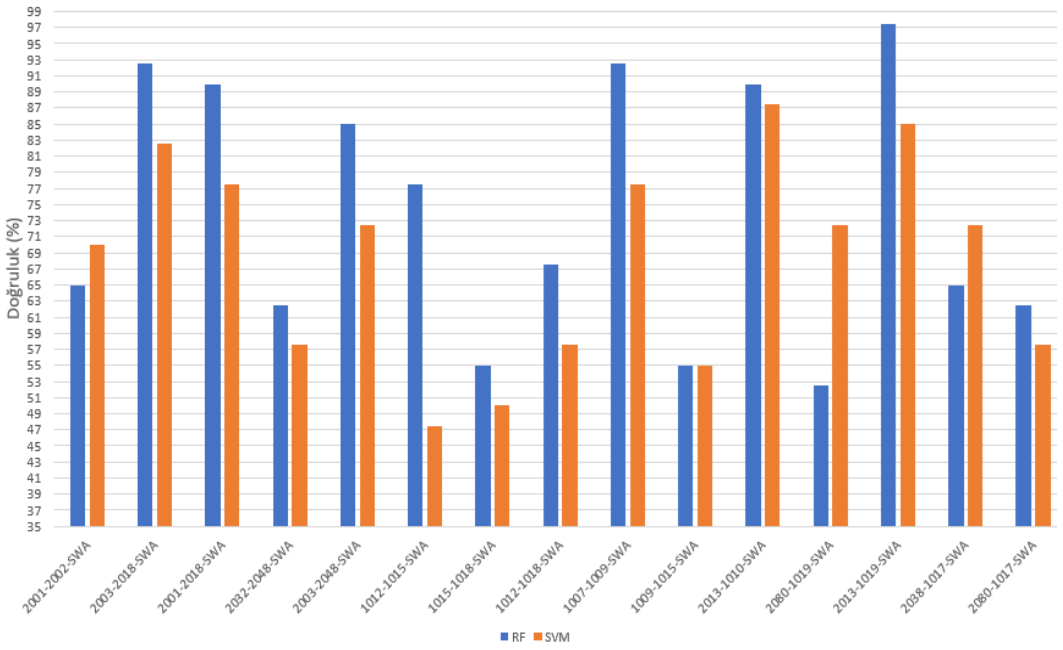
Üçüncü deneyimizde veri kümesinden erkek, kadın ve her iki cinsiyetten sürücü içeren toplamda 15 adet sürücü çifti rastgele seçilmiştir. Bunların her biri sınıflandırmanın yapılacağı bir veri kümesini temsil etmektedir. Dolayısıyla her veri kümesinde iki sınıf bulunmaktadır. Bu sınıflar sürücülerin sistem numaralarıdır. Bu veri hazırlama süreçlerinin ardından VS ve SWA CAN hattı verileri kullanılarak uzunluktan bağımsız öznitelik çıkarımlarını yapılmıştır ve bu öznitelikler kullanarak iki sürücü sınıflandırılmıştır. Buradaki amacımız ise sürücü tahminindeki doğruluk oranının cinsiyet değişimine göre bir farklılık gösterip göstermeyeceğinin tespit edilmesidir. Şekil 5.5 (VS) ve 5.6 (SWA)'da bu deneyin sonuçları gösterilmiştir. Sonuçlar incelendiğinde üç farklı veri kümesinden seçilen sürücü çiftlerinin ortalama doğruluk oranları birbirlerine oldukça yakındır. Buradan da anlaşılacağı üzere cinsiyet sürücü tahmininde bir fark yaratmamaktadır. Doğruluk oranı sürücünün direk kendisi ile ilgilidir. Bu deneyde olduğu gibi cinsiyete göre gruplama yapmak doğru olmayacaktır. Çizelge 5.3 ve 5.4'de temel kıstasımız olan doğruluk oranı ile birlikte diğer sonuç parametreleri ile de deney sonucu ifade edilmektedir.



Şekil 5.5: Cinsiyete bağlı olarak rastgele seçilen sürücü çiftlerinin VS CAN hattı verileri ile yapılan deneyin doğruluk oranları.

Dördüncü deneyimizde kadın ve erkek sürücüler içerisinden en hızlı iki ve en yavaş iki sürücü seçilmiştir. Her sürücü çifti sınıflandırma işleminin gerçekleştirileceği veri kümesini temsil etmektedir. Dolayısıyla her veri kümesinde iki sınıf bulunmaktadır.

Bu sınıflar sürücülerin sistem numaralarıdır. Bu veri hazırlama süreçlerinin ardından VS ve SWA CAN hattı verileri kullanılarak uzunluktan bağımsız öznitelik çıkarımlarını yapılmıştır ve bu öznitelikler kullanarak iki sürücü sınıflandırılmıştır. Buradaki amacımız sürüş süreleri arasındaki farkın sürücü tahminine bir etkisi olup olmadığının gözlemlenmesidir. Beklentimiz sürüş süreleri arasındaki farkın fazla olduğu çiftlerde sınıflandırma doğruluk oranının daha yüksek olmasıdır. Bu deneyde bu çiftler sırasıyla birinci en yavaş- birinci en hızlı ve birinci en yavaş- ikinci en yavaş sürücü çiftleridir. Fakat hem erkek hemde kadın sürücü çiftleri için doğruluk oranlarını incelediğimizde beklentimiz dışında bir durumun gerçekleştiğini görmekteyiz. Şekil 5.7'de bu deneyin sonuçları gözlemlenebilir. Dolayısıyla sürüş süreleri yani hız tek başına sürücü tahmininde etkili olamamaktadır. Ayrıca Çizelge 5.5'de temel kıstasımız olan doğruluk oranı ile birlikte diğer sonuç parametreleri ile de deney sonucu ifade edilmektedir.



Şekil 5.6: Cinsiyete bağlı olarak rastgele seçilen sürücü çiftlerinin SWA CAN hattı verileri ile yapılan deneyin doğruluk oranları.

5.4 Tartışma

Sürücü tanıma araştırmamızda dört adet deney gerçekleştirilmiştir. Bu kısımda kullandığımız araştırma metodolojisinden dolayı mevcut ham sürücü verisi yetersiz gelmektedir. Bu sebepden ötürü her bir sürücünün sürüş verisi eşit büyüklükte parçalara

bölünmüştür ve her parça o sürücü için kendisini tanımlayan yeni bir örüntü oluşturmuştur.

İlk deneyimizde, sürücü parça sayısı değişiminin sürücü sınıflandırma doğruluk oranına etkisi gözlemlenmiştir. Bu gözlem sonucunda, sürücüye ait örüntü sayısını arttırmanın doğruluk oranını arttırdığı tespit edilmiştir.

İkinci deneyimizde, her bir sürücüyü tanımlamak için kullanılan öznitelik sayısını farklı CAN verilerini birleştirerek arttırdığımızda doğruluk oranınında arttığı tespit edilmiştir. Üçüncü deneyimizde, sürücü cinsiyetinin sürücüyü sınıflandırmaya bir etkisi olup olmadığı anlaşılmaya çalışılmıştır. Bu gözlemler sonucunda, farklı ve aynı cinsiyete sahip sürücü çiftlerinin katıldığı deneylerin doğruluk oranlarının birbirine yakın olduğu gözlemlenmiştir. Dolayısıyla cinsiyetin sürücü tahmininde bir fark yaratmadığı anlaşılmıştır.

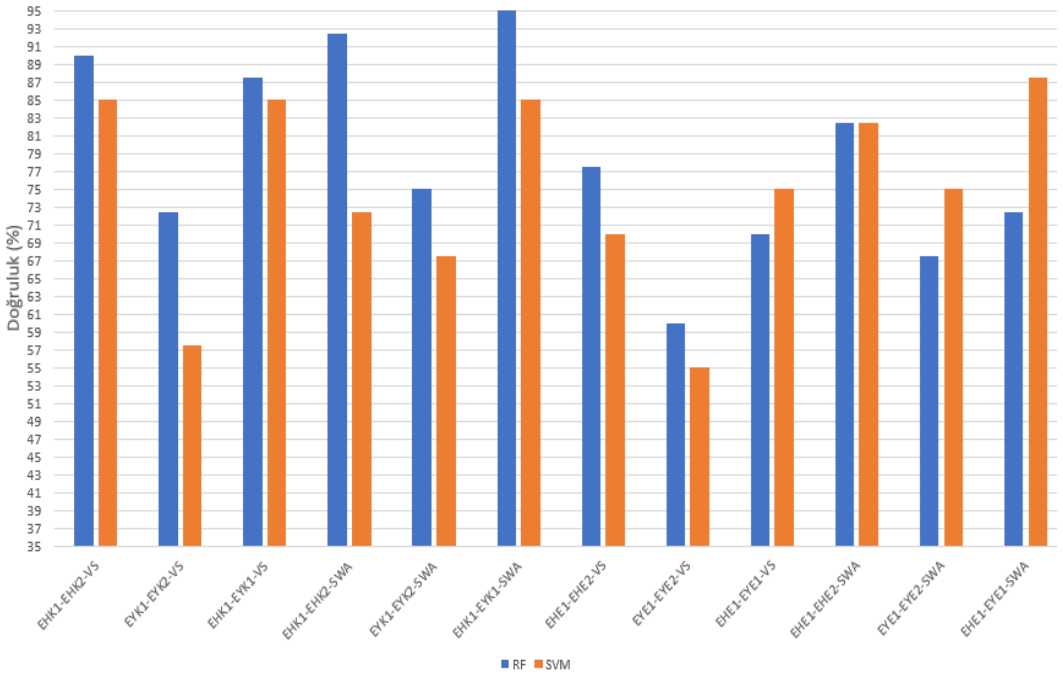
Son deneyimizde ise sürüş sürelerinin sürücü tanımaya bir etkisi olup olmadığı incelenmiştir. Bu deney sonucunda, sürüş süreleri arasındaki farkın sürücü tahmininde etkili olmadığı tespit edilmiştir. Tüm bu deneyler sonucunda, sürücüye ait veri sayısını ve özniteliği arttırmanın sürücü tanıma doğruluk oranını arttırdığı anlaşılmıştır.

Çizelge 5.3: Cinsiyete bağlı olarak rastgele seçilen sürücü çiftlerinin VS CAN hattı verileri ile yapılan deneyin çeşitli metriklere göre en iyi sonuçları.

Veri Tipi, En iyi Alg.	**TP Oranı**	**FP Oranı**	**Precision**	**Recall**	**F−Measure**	**ROC Alanı**
2001-2002-VS , SVM	0.675	0.325	0.679	0.675	0.673	0.675
2003-2018-VS, RF	0.825	0.175	0.826	0.825	0.825	0.911
2001-2018-VS, RF	0.925	0.075	0.926	0.925	0.925	0.975
2032-2048-VS, SVM	0.725	0.275	0.726	0.725	0.725	0.725
2003-2048-VS, RF	0.800	0.200	0.803	0.800	0.799	0.900
1012-1015-VS, SVM	0.625	0.375	0.625	0.625	0.625	0.625
1015-1018-VS, RF	0.650	0.350	0.656	0.650	0.646	0.734
1012-1018-VS, RF	0.600	0.400	0.601	0.600	0.599	0.711
1007-1009-VS, RF	0.925	0.075	0.935	0.925	0.925	0.945
1009-1015-VS, RF	0.625	0.375	0.633	0.625	0.619	0.634
2013-1010-VS, RF	0.875	0.125	0.876	0.875	0.875	0.880
2080-1019-VS, RF	0.675	0.325	0.675	0.675	0.675	0.735
2013-1019-VS, RF	0.850	0.150	0.850	0.850	0.850	0.909
2038-1017-VS, RF	0.500	0.500	0.500	0.500	0.499	0.474
2080-1017-VS, SVM	0.675	0.325	0.675	0.675	0.675	0.675

Çizelge 5.4: Cinsiyete bağlı olarak rastgele seçilen sürücü çiftlerinin VS SWA hattı verileri ile yapılan deneyin çeşitli metriklere göre en iyi sonuçları.

Veri Tipi, En iyi Alg.	TP Oranı	FP Oranı	Precision	Recall	F−Measure	ROC Alanı
2001-2002-SWA , SVM	0.700	0.300	0.700	0.700	0.700	0.640
2003-2018-SWA, RF	0.925	0.075	0.926	0.925	0.925	0.931
2001-2018-SWA, RF	0.900	0.100	0.900	0.900	0.900	0.929
2032-2048-SWA, RF	0.625	0.375	0.628	0.625	0.623	0.685
2003-2048-SWA, RF	0.850	0.150	0.850	0.850	0.850	0.911
1012-1015-SWA, RF	0.775	0.225	0.776	0.775	0.775	0.749
1015-1018-SWA, RF	0.550	0.450	0.555	0.550	0.540	0.504
1012-1018-SWA, RF	0.675	0.325	0.675	0.675	0.675	0.666
1007-1009-SWA, RF	0.925	0.075	0.935	0.925	0.925	0.978
1009-1015-SWA, RF	0.550	0.450	0.555	0.550	0.540	0.530
2013-1010-SWA, RF	0.900	0.100	0.900	0.900	0.900	0.970
2080-1019-SWA, SVM	0.725	0.275	0.756	0.725	0.716	0.725
2013-1019-SWA, RF	0.975	0.025	0.976	0.975	0.975	0.998
2038-1017-SWA, SVM	0.725	0.275	0.730	0.725	0.723	0.725
2080-1017-SWA, RF	0.625	0.375	0.628	0.625	0.623	0.664



Şekil 5.7: Hıza bağlı olarak seçilen sürücü çiftlerinin VS ve SWA CAN hattı verileri ile yapılan deneyin doğruluk oranları.

Çizelge 5.5: Hıza bağlı olarak seçilen sürücü çiftlerinin VS ve SWA CAN hattı verileri ile yapılan deneyin çeşitli metriklere göre en iyi sonuçları.

Veri Tipi, En iyi Alg.	**TP Oranı**	**FP Oranı**	**Precision**	**Recall**	**F−Measure**	**ROC Alanı**
EHK1-EHK2-VS , RF	0.900	0.100	0.900	0.900	0.900	0.963
EYK1-EYK2-VS, RF	0.725	0.275	0.726	0.725	0.725	0.819
EHK1-EYK1-VS, RF	0.875	0.125	0.876	0.875	0.875	0.975
EHK1-EHK2-SWA , RF	0.925	0.075	0.926	0.925	0.925	0.980
EYK1-EYK2-SWA, RF	0.750	0.250	0.750	0.750	0.750	0.716
EHK1-EYK1-SWA, RF	0.950	0.050	0.950	0.950	0.950	0.993
EHE1-EHE2-VS , RF	0.775	0.225	0.776	0.775	0.775	0.889
EYE1-EYE2-VS, RF	0.600	0.400	0.601	0.600	0.599	0.651
EHE1-EYE1-VS, SVM	0.750	0.250	0.753	0.750	0.749	0.750
EHE1-EHE2-SWA , RF	0.825	0.175	0.826	0.825	0.825	0.892
EYE1-EYE2-SWA, SVM	0.825	0.175	0.826	0.825	0.825	0.892
EHE1-EYE1-SWA, SVM	0.875	0.125	0.884	0.875	0.874	0.875

6. SONUÇ VE ÖNERİLER

Araştırmamızda Uyanık veri kümesi CAN hattı verilerini kullanarak sürücü kümeleme, sürücü cinsiyet sınıflandırma ve sürücü tanıma ile ilgili deneyler gerçekleştirdik. Sürücü kümeleme deneylerinde DTW ve kendi geliştirdiğimiz DDW veri dönüşüm metodlarını uygulayarak farklı mesafe metriklerine göre hiyerarşik sürücü kümeleme yaptık. Bu deneydeki amacımız sürüş davranışlarına göre gruplamanın ne ölçüde başarılabildiğini test etmekti. VS ve ERPM CAN verileri üzerinde yapılan deneylerin sonuçlarına göre sürüş süreleri bazında tutarlı sürücü gruplamaları elde ettik. Fakat sürücü cinsiyetlerine göre bir gruplandırmanın yapılmadığını tespit ettik. Bunun üzerine bu ayrımı gerçekleştirmek için daha gelişmiş makine öğrenme tekniklerini kullanmaya karar verdik ve sürücü cinsiyet sınıflandırma deneylerini gerçekleştirdik. Sürücü cinsiyet sınıflandırma deneylerinde veri çoğaltma, öznitelik çıkarımı, öznitelik eleme ve ayrıştırma veri ön işleme metodlarını uygulayarak sınıflandırma doğruluk oranında 0.97 gibi yüksek bir doğruluk oranına ulaştık. Yüksek başarımlı bu sonuç ile CAN sürüş sinyallerinden sürücü özellik çıkarımının yapılabileceğini doğrulamış olduk. Elde edilen sonuçlar üzerine, bu verilerden direk sürücünün kendisini tanımanın ne ölçüde başarılabildiğini test etmek amacıyla sürücü tanıma deneylerini gerçekleştirdik. Sürücü tanıma deneylerinde kendi geliştirdiğimiz bir örüntü parçalama tekniği ve öznitelik çıkarımı veri ön işleme metodlarını uygulayarak 105 adet sürücü arasından 0.1 doğruluk oranında sürücü sınıflandırma işlemini gerçekleştirmeyi başardık. Cinsiyet ve sürüş sürelerinin bu sınıflandırma doğruluk oranına geliştirici bir etkisi olmadığını gözlemledik. Literatür araştırması kısmında sürücü sınıflandırma ile ilgili birçok çalışma mevcuttur [4] [5] [6]. Fakat bu çalışmalar, kullanılan temel teknikler bakımından birbirine benzemektedir. Araştırmamızda ise bu yöntemlerden tamamen farklı bir metodoloji kullanılmıştır. Ayrıca, diğer çalışmalardan farklı olarak deneylerimize veri kümesinde bulunan tüm sürücüler dahil edilmiştir. Sınıflandırılacak sürücü sayısının deney doğruluk oranlarına kritik bir etkisi bulunmaktadır. Bunların haricinde, literatür araştırmalarında CAN verilerini birleştirmenin sürücü sınıflandırmasını olumlu etkilediğinden bahsedilmiştir [14]. Deney sonuçlarımız bu çıkarımı doğrulamaktadır. Dönme ve durma manevralarının daha karakteristik yapıda olduğu belirtilmiştir [13]. Deney sonuçlarımız incelendiğinde, bu sürüş davranışları gerçekleştirilirken kullanılan CAN hattı verilerinin sınıflandırma doğruluk oranlarının yüksek olduğu görülmektedir.

Araştırmamızda ki temel amaç, sürücüyü tanımayı gerçekleştirmek ve ona yardımcı faktörleri tespit ederek literatüre kazandırmaktır. Aynı zamanda bu amaç, CAN verilerinden sürücü ile ilgili çıkarımların yapılabildiğini ispatlayarak bu verilerin hukuki boyutta hassas kişisel veri olarak değerlendirilmesi gerektiğine de

hizmet etmektedir. Elde ettiğimiz deney sonuçları da sürüş örüntülerinin kişilerin özelliklerini yansıttıklarını ve bu verilerin paylaşılırken hassas bir veri olarak ele alınması gerektiğini doğrulamaktadır. Bunların yanı sıra çalışmamızın, gelecekte kullanımının giderek artacağı öngörülen kişiye özel sürüş çözümleri konseptine bir teknolojik alt yapı olarak fayda sağlayacağını düşünmekteyiz.

Gelecekteki araştırmalarımızda, sürüş sinyallerinden sürücü yaş grubu, eğitim seviyesi gibi diğer kişisel özelliklerin çıkarımı ve sürücü tanıma doğruluk oranının geliştirilmesi ile ilgili çalışmalar yapmayı planlamaktayız. Ayrıca farklı veri kümelerini de kullanarak çalışmamızın kapsamını genişletmeyi ve mevcut araştırma yöntemlerimizi bu veri kaynakları için de doğrulamayı hedefliyoruz.

KAYNAKLAR

[1] **H. Abut, H. Erdogan, A. Ercil, et al.**, Corpus and Signal Processing for Driver Behavior, chapter Data collection with UYANIK: too much pain; but gains are coming Springer Business-Science, 2008.

[2] <http://www.endustri40.com/>, alındığıtarih: 11.07.2017.

[3] **W. Lun, C. K. Ng, M. A. Borhanuddin, et al.**, Review of Researches in Controller Area Networks Evolution and Applications Proceedings of the Asia-Pacific Advanced Network. 30. 10.7125/APAN.30.3.

[4] **C. Miyajima, Y. Nishiwaki, K. Ozawa, T. Wakita, K. Itou, K. Takeda**, Cepstral Analysis of Driving Behavioral Signals for Driver Identification. 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, Toulouse, 2006, pp.

[5] **S. Choi, J. Kim, D. Kwak, P. Angkititrakul, J. Hansen**, Analysis and Classification of Driver Behavior Using in-Vehicle CAN-BUS Information Biennale Workshop DSP In-Vehicle Mobile System.

[6] **E. Öztürk**, Driver status identification from driving behavior signals. Yüksek Lisans Tezi, Koç Üniversitesi, Türkiye, 2010.

[7] **K. Igarashi, C. Miyajima, K. Itou, K. Takeda, F. Itakura, H. Abut**, Biometric identification using driving behavioral signals. 2004 IEEE International Conference on Multimedia and Expo (ICME). 1. 65 - 68 Vol.1. 10.1109/ICME.2004.1394126.

[8] **Y. Zheng, X. Shi, A. Sathyanarayana, N. Shokouhi, J. H. L. Hansen**, In-vehicle speech recognition and tutorial keywords spotting for novice drivers' performance evaluation. 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, 2015, pp. 168-173.

[9] **K. Özaçmak**, Analysis of experimental data collected by drivesafe vehicle Uyanık Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Türkiye, 2011.

[10] **J. R. Annam, S. S. Mittapalli, R. S. Bapi**, Time series Clustering and Analysis of ECG heart-beats using Dynamic Time Warping, 2011 Annual IEEE India Conference, Hyderabad, 2011, pp. 1-3.

[11] **A. D. Calin**, Gesture Recognition on Kinect Time Series Data Using Dynamic Time Warping and Hidden Markov Models, 2016 18th

International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), Timisoara, 2016, pp. 264-271.

[12] **D. Hallac et al.**, Driver identification using automobile sensor data from a single turn, 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, 2016, pp. 953-958.

[13] **N. C. Fung et al.**, Driver identification using vehicle acceleration and deceleration events from naturalistic driving of older drivers, 2017 IEEE International Symposium on Medical Measurements and Applications (MeMeA), Rochester, MN, 2017, pp. 33-38.

[14] **M. Van Ly, S. Martin and M. M. Trivedi**, Driver classification and driving style recognition using inertial sensors, 2013 IEEE Intelligent Vehicles Symposium, Gold Coast, QLD, 2013, pp. 1040-1045.

[15] `<http://vpa.sabanciuniv.edu/>`, alındığıtarih: 18.06.2017.

[16] **S. Salvador, P. Chan**, Toward accurate dynamic time warping in linear time and space, Intell. Data Anal. 11, 5 (October 2007), 561-580.

[17] `<https://tsfresh.readthedocs.io/en/latest/>`, alındığıtarih: 08.05.2017.

[18] **S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, K. R. K. Murthy**, Improvements to Platt's SMO Algorithm for SVM Classifier Design, in Neural Computation, vol. 13, no. 3, pp. 637-649, March 1 2001.

[19] **J. Ross Quinlan**, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[20] **Leo Breiman**, Random Forests, Mach. Learn. 45, 1 (October 2001), 5-32.

[21] **Y. Freund, R. E. Schapire**, Experiments with a new boosting algorithm, In Proceedings of the Thirteenth International Conference on International Conference on Machine Learning (ICML'96), Lorenza Saitta (Ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 148-156.

[22] **Y. Freund, R. E. Schapire**, Large Margin Classification Using the Perceptron Algorithm, Machine Learning, 37, 3 (December 1999), 277-296.

[23] **R. R. Bouckaert**, Bayesian networks in Weka, Technical Report 14/2004. Computer Science Department, University of Waikato, 2004.

[24] **N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer**, SMOTE: synthetic minority over-sampling technique, J. Artif. Int. Res. 16, 1 (June 2002), 321-357.

[25] <https://www.kisiselverilerinkorunmasi.org/>, alındığıtarih: 21.02.2018.

ÖZGEÇMİŞ

Ad-Soyad : Batuhan Karataş

Uyruğu : Türkiye Cumhuriyeti

Doğum Tarihi ve Yeri : 08.08.1992 – Keçiören

E-posta : batuhankaratas@etu.edu.tr

ÖĞRENİM DURUMU:

- **Lisans** : 2015, TOBB ETÜ, Mühendislik Fakültesi, Bilgisayar Mühendisliği
- **Yüksek Lisans** : 2018, TOBB ETÜ, Mühendislik Fakültesi, Bilgisayar Mühendisliği, Araştırma Burslu Yüksek Lisans Öğrencisi

MESLEKİ DENEYİM VE ÖDÜLLER:

Yıl	**Yer**	**Görev**
2016–	ASELSAN A.Ş.	Sistem Tasarım Mühendisi
2015–2016	İnnova Bilişim Çözümleri A.Ş.	Yazılım Uzmanı

YABANCI DİL: İngilizce, Almanca

TEZDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER:

- **Karatas, B., Abul, O.** 2018. Sürüş örüntülerinden cinsiyet tahmin edilebilir mi?, 2018. Akademik Bilişim Konferansı, Ocak 31- Şubat 2, Karabük, Türkiye.