

T.C.
ISPARTA UYGULAMALI BİLİMLER ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

HIZ KESİCİ TÜRMEKLERİN YAKIT TÜKETİMİNE ETKİSİNİN
DERİN ÖĞRENME ALGORİTMALARI İLE TAHMİN
EDİLMESİ: ÖRNEK BİR UYGULAMA

Mustafa Fatih TOSUN

Danışman
Dr. Öğr. Üyesi Ali ŞENTÜRK

ISPARTA - 2022



© 2022 [Mustafa Fatih TOSUN]

İÇİNDEKİLER

	Sayfa
İÇİNDEKİLER	i
ÖZET.....	ii
ABSTRACT.....	iii
TEŞEKKÜR.....	iv
ŞEKİLLER DİZİNİ.....	v
ÇİZELGELER DİZİNİ	vi
SİMGELER VE KISALTMALAR DİZİNİ	vii
1. GİRİŞ	1
2. KAYNAK ÖZETLERİ	3
2.1. Veri Toplama Yöntemlerine İlişkin Yapılan Çalışmalar	3
2.2. Yakıt Tüketimine Etki Eden Faktörlere İlişkin Yapılan Çalışmalar	4
2.3. Sürücü Davranışına İlişkin Yapılan Çalışmalar	6
3. MATERYAL VE YÖNTEM	8
3.1. Çalışma Alanı.....	9
3.2. Aracın Özellikleri.....	10
3.3. CAN Veri Yolu	11
3.4. OBD-II	13
3.5. Gömülü Sistem.....	15
3.6. Araçtan Veri Alınması	16
3.7. Yakıt Tüketiminin Anlık Hesaplanması	18
3.8. Veri Setinin Hazırlanması	21
3.9. Sistemin Eğitilmesi için Ön İşlenmesi	23
3.9.1. Pencere kaydırma (window slide) yöntemi.....	23
3.9.2. Veri setinin eğitim ve test setleri için bölünmesi	24
3.9.3. Normalizasyon (minmax scaler)	24
3.10. Yapay Zekâ	25
3.10.1. Makine öğrenmesi.....	26
3.10.2. Derin öğrenme.....	26
3.10.2.1. Aktivasyon fonksiyonu	27
3.10.2.2. Bırakma (dropout) katmanı	28
3.10.2.3. Regression (regresyon) katmanı.....	29
3.10.3. Kullanılan derin öğrenme yöntemleri	29
3.10.3.1. RNN (tekrarlayan sinir ağları - recurrent neural networks).....	30
3.10.3.2. LSTM (uzun kısa süreli bellek - long short term memory)	32
3.10.3.3. GRU (geçitli tekrarlayan birimler - gated recurrent unit)	34
3.10.3.4. Optimizasyon ve performans ölçüm yöntemleri	35
4. BULGULAR VE TARTIŞMA	37
4.1. Önerilen Modellerin Performans Sonuçları	39
5. SONUÇ VE ÖNERİLER	48
KAYNAKLAR	50
ÖZGEÇMİŞ	56

ÖZET

Yüksek Lisans Tezi

HIZ KESİCİ TÜMSEKLERİN YAKIT TÜKETİMİNE ETKİSİNİN DERİN ÖĞRENME ALGORİTMALARI İLE TAHMİN EDİLMESİ: ÖRNEK BİR UYGULAMA

Mustafa Fatih TOSUN

Isparta Uygulamalı Bilimler Üniversitesi
Lisansüstü Eğitim Enstitüsü
Elektrik Elektronik Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Ali ŞENTÜRK

Trafiği düzenlemek için sıklıkla hız kesici tümsekler (kasisler) kullanılmaktadır. Bununla birlikte kasisler sürüş konforunun azalması ve yakıt tüketimini arttırma gibi problemlere de sebep olmaktadır. Günümüzde otomobillerde verilere OBD-II portundan anlık olarak ulaşılabilir. Bu tezde kasislerde araçların yavaşlama ve hızlanmasının yakıt tüketimine etkisinin belirlenmesi amaçlanmıştır. Bunun için kasis bulunan güzergahlarda otomobil kullanılmış ve OBD-II portundan Arduino ile gerçek zamanlı hız ve yakıt tüketimi verileri alınmıştır.

Alınan veriler ön işleme ve normalizasyona tabi tutulmuştur. Yakıt tüketimini tahmin etmek için Tekrarlayan Sinir Ağları (RNN), Uzun Kısa Süreli Bellek (LSTM), Geçitli Tekrarlayan Birim (GRU) yineleyen derin öğrenme modelleri geliştirilmiştir. Ön işlenmiş veriler modelleri eğitmede kullanılmıştır. Veri setinin örnekleme zamanı olarak 300 ms seçilmiştir. Yakıt tüketiminin hesaplanması için geriye doğru birden fazla veriye ihtiyaç duyulduğundan sistem pencere kaydırma yöntemi ile eğitilmiştir. Pencere boyutu ise 5, 10 ve 15 olarak seçilmiştir. Verilerin yüzde atması %60'ı eğitim ve %40'ı doğrulama için kullanılmıştır.

Geliştirilen modellerde hiperparametre optimizasyonu yapılmıştır. Böylece katman sayısı, katmanlardaki hücre sayısı, hücrelerin aktivasyon fonksiyonları ve öğrenme oranı belirlenmiştir. Doğrulama setinde en düşük binde 63 ortalama kare hatası elde edilmiştir. Geliştirilen modeller kullanılarak farklı kasisler ve hız senaryolarının yakıt tüketimine olan etkileri tahminlenmiştir. Kasislerden geçiş için belirlenen hız ve zaman verileri kullanılarak yakıt tüketiminin kasis etkisi boyunca %16.30 ile %31.03 arasında arttırdığı sonucuna ulaşılmıştır.

Anahtar Kelimeler: Yakıt tüketimi, Kasis, CAN veri yolu, Makine öğrenmesi, RNN

2022, 56 sayfa

ABSTRACT

Master's Thesis

ESTIMATING THE EFFECT OF CUSHIONERS ON FUEL CONSUMPTION WITH DEEP LEARNING ALGORITHM: AN EXAMPLE APPLICATION

Mustafa Fatih TOSUN

**Isparta University of Applied Sciences
The Institute of Graduate Education
Department of Electrical Electronics Engineering**

Supervisor: Asst. Prof. Dr. Ali ŞENTÜRK

Speed bumps are used to regulate traffic frequently. However, bumps also cause problems such as decreased driving comfort and increased fuel consumption. Nowadays, data can be accessed instantly from the OBD-II port in vehicles. In this thesis, it is aimed to determine the effect of vehicle's slowing down and accelerating over bumps on fuel consumption. For this purpose, an automobile is used on the routes with bumps and real-time speed and fuel consumption data is obtained from the OBD-II port with Arduino.

The received data is preprocessed and normalized. Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU) deep learning models are developed to predict fuel consumption. The preprocessed data is used to train the models. The sampling time of the data set is 300 ms. The system is trained with the window slide method. The window size is selected as 5, 10 and 15. 60% of the data is used for training and 40% for validation.

Hyperparameter optimization is conducted in the models. Thus, the number of layers and the units in the layers, the activation functions and the learning rate is specified. The lowest mean square error is obtained as per thousand 63 in the validation set. The effects of different speed scenarios on fuel consumption are predicted by using the models. In conclusion, the fuel consumption is increased between 16.30% and 31.03% during the impact of the bumps, by using the speed and time generated for the bumps.

Key Words: Fuel consumption, Bump, CAN bus, Machine learning, RNN

2022, 56 pages

TEŞEKKÜR

Tezimin yürütülmesinde desteğini ve emeğini hiçbir zaman esirgemeyen tez danışman hocam sayın Dr. Öğr. Üyesi Ali ŞENTÜRK'e teşekkürlerimi sunarım.

Teze yönelik yaptığım çalışmaların değerlendirilmesinde bilgileri ile desteklerini esirgemeyen Dr. Öğr. Üyesi Bekir AKSOY ve Prof. Dr. Ecir Uğur KÜÇÜKSİLLE hocalarıma teşekkürlerimi sunarım.

Tez çalışması ve yazımı sırasında bilgi ve tecrübeleriyle çalışmamda açık noktalarda tavsiyeleriyle yardımcı olan Öğr. Gör. Dr. Halit ÇETİNER hocama teşekkürlerimi sunarım.

Tezde yer alan derin öğrenme hesaplamaları TÜBİTAK ULAKBİM, Yüksek Başarım ve Grid Hesaplama Merkezi'nde (TRUBA kaynaklarında) gerçekleştirilmiştir.

Tezimin her aşamasında beni yalnız bırakmayan ve her türlü desteği veren başta annem Feride TOSUN, babam Prof. Dr. İsmail TOSUN ve tüm TOSUN ailesine sonsuz sevgi ve saygılarımı sunarım.

Mustafa Fatih TOSUN
ISPARTA, 2022

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 3.1. Blok diyagram.....	8
Şekil 3.2. Çalışma alanında bulunan kasisler.....	9
Şekil 3.3. CAN veri yolu çalışma Sistemi	12
Şekil 3.4. CAN bus verileri	13
Şekil 3.5. OBD-II konnektörü ve pin çıkışları	14
Şekil 3.6. Gömülü sistem bağlantısı.....	15
Şekil 3.7. Modern araçlardaki elektronik yükler.....	17
Şekil 3.8. Veri toplama akış diyagramı	19
Şekil 3.9. Geliştirilen sistemin basamakları.....	20
Şekil 3.10. Araçtan alınan ham veriler	21
Şekil 3.11. Hazırlanan veri seti	22
Şekil 3.12. Pencere kaydırma metodu.....	23
Şekil 3.13. Veri setinin eğitim ve test verisi olarak ayrılması	24
Şekil 3.14. Bırakma katmanı (a) uygulanmamış, (b) uygulanmış görüntü	29
Şekil 3.15. RNN hücre yapısı.....	30
Şekil 3.16. RNN çeşitleri	31
Şekil 3.17. RNN hücre yapısı.....	31
Şekil 3.18. LSTM mimarisi.....	32
Şekil 3.19. GRU blok diyagramı.....	34
Şekil 4.1. Hiperparametre optimizasyonu.....	38
Şekil 4.2. Bellekli derin öğrenme model blok diyagramı	39
Şekil 4.3. a)5, b)10, c)15 pencere boyutlu RNN eğitim ve doğrulama kayıpları.....	41
Şekil 4.4. a)5, b)10, c)15 pencere boyutlu LSTM eğitim ve doğrulama kayıpları	42
Şekil 4.5. a)5, b)10, c)15 pencere boyutlu GRU eğitim ve doğrulama kayıpları.....	43
Şekil 4.6. RNN-5 tahmin performansı	44
Şekil 4.7. LSTM-5 tahmin performansı	44
Şekil 4.8. GRU-15 tahmin performansı	44
Şekil 4.9. Kasisten geçiş esnasında doğrusal yavaşlama ve hızlanma	45
Şekil 4.10. Kasisten geçiş sırasındaki yakıt tüketim tahmini.....	46

ÇİZELGELER DİZİNİ

	Sayfa
Çizelge 3.1. Kasis geçiş detay bilgileri	10
Çizelge 3.2. Aracın teknik özellikleri ve yakıt bilgisi.....	11
Çizelge 3.3. OBD-II modları.....	12
Çizelge 3.4. İdeal hava/yakıt oranı ve yoğunluk değerleri.....	18
Çizelge 3.5. Aktivasyon fonksiyonları.....	28
Çizelge 4.1 Veri setinin pencere boyutu, eğitim ve test veri sayıları.....	37
Çizelge 4.2. Derin öğrenme modelleri performans sonuçları	39
Çizelge 4.3. Modelde kullanılan hiperparametreler ve parametre sayıları	40
Çizelge 4.4 Gerçek zamanlı kasis geçiş bilgileri	45
Çizelge 4.5. Kasisten kaynaklı yakıt artış oranları.....	46



SİMGELER VE KISALTMALAR DİZİNİ

CAN	Kontrol alan ağı
CARB	California hava kaynakları kurulu
DTC	Teşhis hata kodları
ECU	Elektronik kontrol birimi
EEPROM	Elektriksel olarak silinebilir programlanabilir salt okunur bellek
EPA	Çevre koruma kurumu
GPS	Küresel konumlama sistemi
GRU	Geçitli tekrarlayan birimler
GSM	Mobil iletişim için küresel sistem
HDM	Otoyol geliştirme ve yönetim
ISO	Uluslararası standardizasyon örgütü
LSTM	Uzun kısa süreli bellek
MAF	Kütle hava akışı
OBD	Yerleşik teşhis
OCW	Petrol tüketim ağırlığı
PETFORM	Petrol ve doğal gaz platformu derneği
PID	Parametre kimliği
RNN	Yinelemeli sinir ağları
RPM	Dakikadaki devir sayısı
SAE	Otomotiv mühendisleri birliği
SRAM	Statik rastgele erişimli bellek
YSA	Yapay sinir ağları

1. GİRİŞ

Ülkemiz petrol bölgelerine yakın olmakla birlikte henüz ülkemizden çıkarılan petrol ülkemizin ihtiyacını karşılayacak düzeyde değildir. Bundan dolayı petrol ürünleri yurt dışından ithal edilerek kullanılmaktadır. PETFORM (Petrol ve Doğal Gaz Platformu Derneği) 2020 verilerine göre ülkemizde tüketilen petrolün %7.1'i ülkemizde çıkarılmaktadır. Petrol kullanımının büyük bölümünü özel araçların tüketimi oluşturmaktadır (Anonim, 2021a). Türkiye'de petrol ihtiyacının büyük oranda ihracatla sağlanması, döviz kurlarının yüksek olması ve petrolün vergilendirilmesinden dolayı akaryakıt fiyatları yüksektir. Ayrıca akaryakıt dağıtım şirketlerinin bayi masrafları, lojistik giderleri ve taşıma masrafları da fiyatlara etki etmesinden dolayı tüketilen petrol ürünlerinin verimli kullanılması zaruret halini almıştır.

Ülkemizde araç sayısı ve trafik yoğunluğu her geçen gün artmaktadır. Artan trafiğin ve trafik hız kurallarına riayet etmeyen sürücülerin can ve mal kaybına sebep olmaması için yollarda hız kesici engeller (kasis) kullanılmaktadır. Her ne kadar kasisler trafiği düzenlemeye ve rahatlatmada etkili yöntemler olsa da standartlara uygun yapılmamış kasisler aracın gereğinden çok fazla yavaşlamasına veya zarar görmesine sebep olabilmektedir. Ayrıca çok sık uygulanan kasisler sürüş konforunu olumsuz etkilemekte, trafik yoğunluğunun çok az olduğu saatlerde dahi kasisten geçen bir araç yolun hız sınırlarının çok altına düşürerek yavaşlamasını gerektirmektedir. Ambulans, itfaiye gibi acil durumlarda hızlı gitmesi gereken araçların da yavaşlamasına sebep olmaktadır. Yavaşlama sonrası araçların hızlanması esnasında ise gürültü kirliliği oluşmakta, egzoz salınımı artmaktadır. Ayrıca bu durum yakıt tüketimin de artmasına sebep olmaktadır.

Gelişen çip ve sensör teknolojisi otomotiv sektörünü doğrudan etkilemiştir. Farklı amaca yönelik geliştirilen elektronik sistemler her geçen yıl taşıtlarda daha fazla kullanılmaktadır. İlk olarak 1940'lı yıllarda BMW tarafından kullanılan Elektronik Kontrol Ünitesi (Electronic Control Unit - ECU) günümüzde bütün araçlarda standart olarak bulunmaktadır. Araçlarda ECU ile diğer elektronik cihazlar arasındaki iletişim Kontrol Alan Ağı (Control Area Network - CAN) veri yolu ile sağlanmaktadır.

Yerleşik Teşhis (On-board Diagnostic - OBD-II) CAN veri yoluna bağlı olan arıza teşhisi ve aracın verilerinin raporlanması için kullanılan bir ara yüzdür.

Bu tezde, OBD-II ara yüzünden elde edilen veriler kullanılarak kasislerin yakıt tüketimine etkilerinin makine öğrenmesi ile tahmin edilmesi gerçekleştirilmiştir. Bunun için OBD-II ara yüzünden elde edilen veriler kullanılarak Yinelemeli Sinir Ağları (Recurrent Neural Network- RNN), Uzun Kısa Süreli Bellek (Long Short Term Memory - LSTM) ve Geçitli Tekrarlayan Birimler (Gated Recurrent Unit - GRU) derin öğrenme mimarileri ile eğitilmiştir. Daha sonra derin öğrenme algoritmalarından elde edilen sonuçlara göre kasislerin farklı sürüş senaryolarında yakıt tüketimine etkisinin analizleri yapılmıştır.

2. KAYNAK ÖZETLERİ

Gerçekleştirilen tez çalışması ile ilgili literatür taraması; veri toplama yöntemlerine ilişkin yapılan çalışmalar, yakıt tüketimine etki eden faktörlere ilişkin yapılan çalışmalar, sürücü davranışına ilişkin yapılan çalışmalar olmak üzere üç bölümde ele alınmıştır.

2.1. Veri Toplama Yöntemlerine İlişkin Yapılan Çalışmalar

Coşkun (2008), yaptığı çalışmada, CAN veri yolu protokolünü kullanarak araç takip sistemi uygulaması gerçekleştirilmiştir. Bu uygulamada GPS uydularından alınan konum, hız, yön gibi bilgileri CAN protokolü ile araçtan toplanan yakıt seviyesi, sıcaklık, motor devri, ses, görüntü vb. gibi uygulamanın gerektirdiği daha birçok bilgi ile birleştirip GSM alt yapısını kullanarak bir merkeze gönderilmiş ve kullanıcılar internet üzerinden bu bilgilere ulaşabilmesi sağlanmıştır.

Ko (2010), yapmış olduğu çalışmada, yolun yüzeyinin pürüzlü olması durumu için yakıt tüketimindeki değişimi ve Kore araçlarının çeşitli hızlardaki yakıt tüketimlerini ölçmüştür. Yol yüzeyi pürüzlülüğü için yakıt tüketimindeki değişim oranı Kore'de HDM-4 (Highway Development Manage - Otoyol Geliştirme ve Yönetim) programı ile hesaplanmıştır. Çalışmadan elde edilen sonuçlara göre yolu iyileştirmeyi veya düzeltmeyi belirlemek amaçlanmıştır. Çalışmada en çok pazar payına sahip binek araçlar çeşitli hızlarda test edilmiştir.

Quanqi vd. (2011), OBD-II modül kullanılarak araçlarda hataların gösterilmesi için bir sistem tasarlamışlardır. Bu sistem ile tespit edilebilen 400'e kadar hata LCD (Liquid Crystal Display) ekran üzerine gözlemlenebilmektedir.

Baek vd. (2011) yaptıkları çalışmada, OBD-II, kameralar ve Global Konumlandırma Sistemi (GPS) gibi harici sensörlerden veri göndererek, aracın mevcut durumu hakkında gerçek bir değerlendirme yapan; video, zaman ve konum verilerini toplayan bir multimedya ağı tasarlamışlardır.

Cueva-Fernandez vd. (2014) yaptıkları çalışmada, karayolu taşıtları için bütünleşik bir platform tasarlamışlardır. Tasarlanan platform ile son kullanıcılar için, birbirleri ile iletişim halinde olan araçlardan ve araç sensörlerinden gerçek zamanlı olarak elde edilen bilgiler kullanılarak zengin internet uygulamaları üretilmesini amaçlanmıştır.

Kowalik (2018), yapmış olduğu çalışmada, arıza tespit cihazı (OBD-II) kullanarak bir veri toplama ortamının nasıl oluşturulabileceğini ve bir araçtan nasıl veri elde edileceğini açıklamıştır. Çalışma arızalı termostatı olan bir arabadan toplanan veri analizinin ön sonuçlarını içermektedir.

Kowalik ve Szpyrka (2019), yapmış oldukları çalışmada, bir arabadan gerçek zamanlı veri alımı için veri toplama ortamı oluşturmuşlardır. OBD-II kullanılarak arabanın elektronik kontrol birimlerinden (ECU) veriler toplanmıştır. Toplanan veriler Torque PRO uygulamasını çalıştıran akıllı telefon veya internet uygulaması üzerinden gerçek zamanlı olarak sunuculara aktarılabilir. Toplanan veriler hem çevrimiçi hem de çevrimdışı olarak analiz edilebilir.

2.2. Yakıt Tüketimine Etki Eden Faktörlere İlişkin Yapılan Çalışmalar

OBD-II portundan elde edilen veriler kullanılarak aracın yakıt tüketimine etki eden faktörlerin tespitine yönelik gerçekleştirilen çalışmalar aşağıda ele alınmıştır.

Mi-JinKim vd. (2010), OBD-II adaptörü ve navigasyon sistemini içeren bir araç tanı programı geliştirmişlerdir. Geliştirilen sistem meydana gelebilecek çeşitli araç arızalarını teşhis etmek için kullanmıştır. ECU'dan navigasyon sistemine diagnostik verileri aktaran bir OBD-II adaptörü kullanılarak bluetooth protokolü ile veri aktarımı gerçekleştirilerek multimedya ve geometri bilgi hizmetleri ile birlikte gelişmiş servisler kullanıcılara sunulmaktadır.

Zaldivar vd. (2011), yapmış oldukları çalışma ile, OBD-II ara yüzünü kullanarak araç bilgilerini akıllı telefon üzerinden izleyen ve kaza tespiti yapan android tabanlı bir sistem geliştirmişlerdir. Geliştirilen sistem ile önden çarpışma durumunda merkezkaç kuvveti kullanılarak hava yastığı ile birlikte kazalar algılanmış acil durum ekiplerine otomatik telefon aramaları gerçekleştirilmiştir.

Wang vd. (2015) yaptıkları çalışmada, elektrikli taşıtların tasarlanması ve değerlendirilmesinde gerçek zamanlı sürüş döngülerinin geliştirilmesinin ve kullanılmasını incelemişlerdir. Gerçekleştirilen çalışma ile belirli bir bölgedeki optimum güç aktarma sistemi yapılandırmalarını ve gerçek zamanlı sürüş düzenlerinin araçların yakıt tüketimi üzerindeki etkisini incelemişlerdir. İki yıllık inceleme sonucu bir sürüş şekli araştırması ve seyahat verilerine dayanarak, Pekin'deki elektrikli araçların yakıt tüketimini hesaplanarak ABD değerleriyle karşılaştırılmıştır.

Syahputra (2016) yapmış olduğu çalışmada, nöro-bulanık mantık yöntemi kullanarak araç yakıt tüketimini tahmin etmiştir. Elde edilen sonuçlara göre yakıt tüketimine etki eden faktörler silindir sayısı, yer değiştirme, beygir gücü, ağırlık, hızlanma ve model yılı olarak belirlemiştir.

Wickramanayake ve Bandara (2016) yapmış oldukları çalışmada, bir otobüsün yakıt verimliliğini iyileştirmek için mesafe, yük, sürücü davranışları ile birlikte yol koşulları, hava şartları ve trafik durumunu dikkate almışlardır. Yakıt tüketimini tahmin etmek için makine öğrenmesi metodlarından rastgele orman, gradyan azalma ve sinir ağları metotları karşılaştırmalı olarak kullanılmıştır.

Zhang vd. (2016), yapmış oldukları çalışmada, trafik sıkışıklığı altındaki kentsel alanlarda rota planlama stratejisini geliştirmek için bir rota planlama algoritması geliştirmişlerdir. Bu algoritmada Petrol Tüketim Ağırlığı (OCW - Oil Consumption Weight) ve rota planlama yöntemleri göz önüne alınarak, yağ tüketimini azaltan ve sıkışıklığı önleyen böylece enerji verimliliğini arttırmada sıklıkla kullanılan en iyi yolu planlayan OCW-Dijkstra algoritması ile yol planlaması gerçekleştirilmiştir.

Perrotta vd. (2017), yapmış oldukları çalışmada, damperli kamyonların yakıt tüketimi tahmininde Destek Vektör Makinesi (SVM) (Support Vector Machine), Rastgele Orman (RF) (Random Forest) ve Yapay Sinir Ağı (YSA) (Artificial Neural Networks) modellerini geliştirmiş ve performansları karşılaştırmışlardır. Yakıt tüketimini etkileyen sensör ve yol bilgilerini toplamışlanmıştır. Yeni ve daha doğru yakıt tüketiminin hesaplanması için araç sensör verilerinin yanı sıra modele yol bilgileri de eklenmiştir. Bu üç yöntemde yüksek hassasiyetli modeller geliştirmek mümkün

olmasına rağmen, RF'nin SVM ve ANN'den çok az da olsa daha iyi performans gösterdiğini ifade etmişlerdir.

Şen (2020), yapmış olduğu tez çalışmasında, makine öğrenmesi ile yakıt tüketimini iyileştirmeyi amaçlamıştır. Burada aracın ECU'sundan hız, RPM (Revolutions Per Minute - Dakikadaki Devir), fren pedalı konumu, su sıcaklığı ve MAF (Manifold Air Flow- Kütle Hava Akış Sensörü) verileri toplanmıştır. Toplanan veriler makine öğrenmesi algoritmaları kullanarak anlık yakıt tüketimi tahmini yapmayı amaçlamıştır. Ayrıca araçtan alınan sensör verilerinin anlık yakıt tüketimine etkisi gözlemlenmiştir.

Uyanık vd. (2020), yapmış oldukları çalışmada, konteynır yük gemisinin yakıt tüketimini makine öğrenmesi ile tahmin etmişlerdir. Ana motor devri, ana motor silindir değerleri, süpürme havası, şaft göstergeleri gibi parametreler yakıt tüketimini etkileyen faktörler olarak seçilmiştir. Tahmin işlemi için Çoklu Lineer Regresyon, Ridge ve LASSO Regresyon, Destek Vektör Regresyon, Ağaç Tabanlı Algoritmalar, Boosting Algoritmaları gibi farklı yapay zekâ modelleri kurulmuştur. Modellerin doğruluğu, K-kat çapraz doğrulama ile belirlenmiştir. Tahmin modellerinin doğruluğunu değerlendirmek için ortalama karesel hata, ortalama mutlak hata, belirleme katsayısı gibi hata metrikleri kullanılmış ve değişkenler arası korelasyon analiz işlemlerini gerçekleştirmiştir.

2.3. Sürücü Davranışına İlişkin Yapılan Çalışmalar

Choi vd. (2007) tarafından sürücü davranışı, CAN veri yolu bilgisinden elde edilen sürüş özelliklerinin sırasını yakalamak için kullanılan Hidden Markov Modelleri (HMM) ve GMM'ler (Gaussian Mixture Models) aracılığıyla direksiyon açısı, fren durumu, hızlanma durumu ve araç hızından alınan veriler kullanılarak tanımlanmış ve modellenmiştir. Eylem sınıflandırması için %69 ve sürücü tanımlaması için %25 doğruluk elde edilmiştir.

Meseguer vd. (2013) gerçekleştirdikleri çalışmalarında, bir sürüş stili uygulaması, geri yayılma veri madenciliği algoritmasını ve sinir ağlarını kullanarak sürücünün her rotadaki davranışını analiz etmişlerdir. Hız, dakikadaki devir ve hızlanma gibi

parametrelere ve daha sonra aracın o anda hareket ettiđi yolu karakterize eden ve sınıflandıran sinir ađına dayalı algoritma uygulanmıřtır. Ayrıca sürücünün saldırganlık derecesi de analiz edilmiřtir.

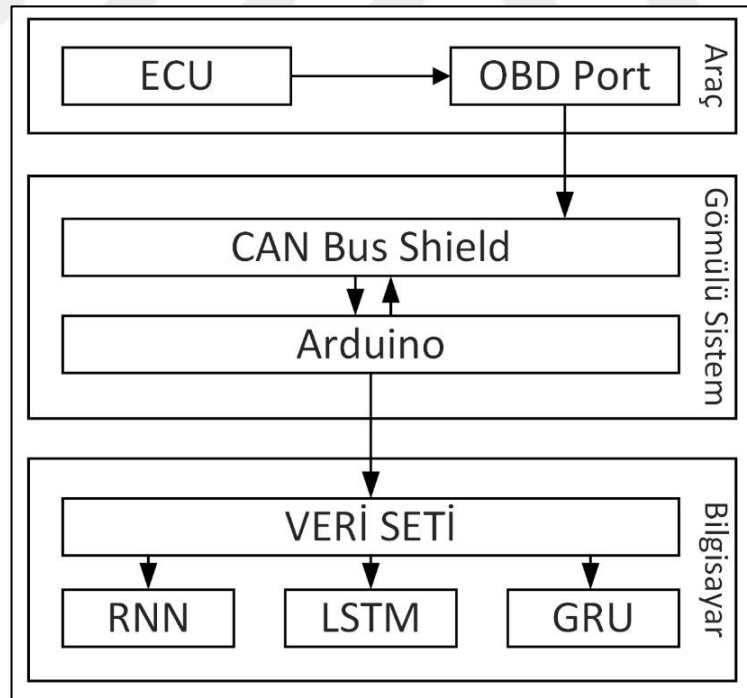
Khanapuri vd. (2015) gerekleřtirdikleri alıřmalarında, araba asistanı uygulaması gerekleřtirmiřlerdir. Bu uygulama iin motor devri, yakıt durumu, yerleřik teřhis (OBD-II) üzerinden gaz kelebeđi konumu gibi parametreleri izleyen ve amatr sürüclere vites deđiřtirmede ve araç arızalarında yardımcı olabilecek bir Android uygulaması önermiřlerdir. Gerekleřtirilen uygulama ile sürüclerin daha iyi bir yakıt ekonomisi elde etmelerine yardımcı olunmuřtur. Mevcut alıřma sonucunda kavřaklarda sürüc davranıřı sınıflandırması sorununa bu teknikleri uygulamanın faydalarını keřfetmektedir.

3. MATERYAL VE YÖNTEM

Bu tezde yapılan çalışmalar araçtan veri toplama ve toplanan verilerin analizi olmak üzere iki başlık altında ele alınmıştır.

Veri toplama işlemi için Ford Fiesta marka araç seçilmiştir. Araçtan veriler Arduino Uno ve CAN veri yolu kartı (CAN Bus shield) ile OBD-II üzerinden alınmıştır. Araç çalıştırıldığında OBD-II portundan veri toplama işlemi başlamaktadır. Elde edilen veriler Seri haberleşme (UART) ile bilgisayar ortamına aktarılmıştır. Bu veriler yakıt tüketimini tahmin etmede kullanılacak olan derin öğrenme modelleri için veri setini oluşturmuştur.

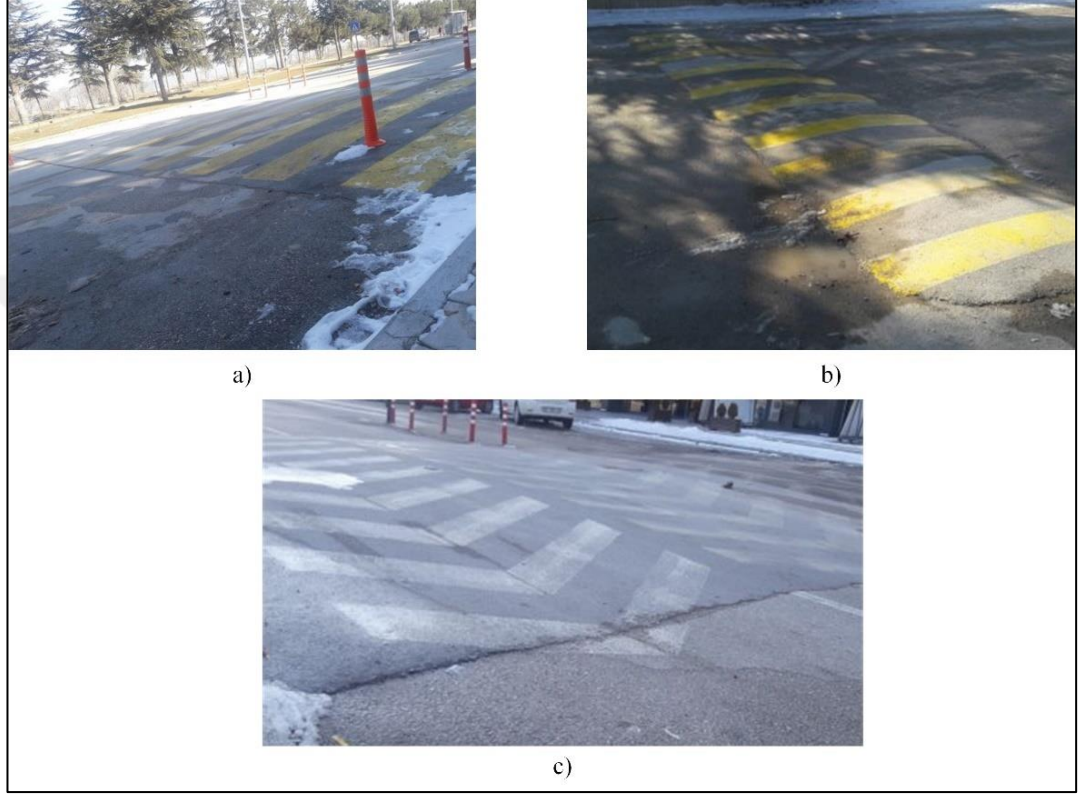
Yapılan çalışmada aracın yakıt tüketimini doğru hesaplamak için yineleyen derin öğrenme metotları olan RNN, LSTM ve GRU mimarileri kullanılmıştır. Tez çalışması için tasarlanan sistemin blok diyagramı Şekil 3.1’de verilmiştir.



Şekil 3.1. Blok diyagram

3.1. Çalışma Alanı

Kasislerin yakıt tüketimine etkisinin analizi için kullanılacak veri setini elde etmek için farklı tipte kasisler içeren güzergahlar seçilmiştir. Şekil 3.2’de farklı kasis boy ve yüksekliklerine göre görseller verilmiştir.



Şekil 3.2. Çalışma alanında bulunan kasisler

Çizelge 3.1’de kasislerden araç geçişi esnasındaki veriler bulunmaktadır. Bu çizelgede kasis fark edildiğinde normal seyir hızı (giriş hızı), kasisten kaynaklı yavaşlamanın ne kadar sürdüğü (yavaşlama süresi) ve yavaşlama mesafesi, kasis sürecinde ulaşılan en düşük hız (min. hız), kasis sonrası normal seyir hızına ulaşmak için geçen süre (hızlanma süresi), bu süredeki mesafe ve çıkış hızı gösterilmiştir

Çizelge 3.1. Kasis geçiş detay bilgileri

Giriş Hızı(km/sa)	Yavaşlama Süresi(sn)	Yavaşlama Mesafesi(m)	min. Hız(km/sa)	Hızlanma Süresi(sn)	Hızlanma Mesafesi(m)	Çıkış Hızı(km/sa)
26	4	21.12	15	4.8	29.06	27
26	5.7	33.86	12	8	41.54	26
36	7.3	55.02	19	4.1	29.77	37
37	8.1	74.09	9	11.4	46.02	40
38	4.4	30.81	15	6.6	48.34	42
40	5.2	43.60	15	6	42.17	36
42	5.3	55.24	32	3.9	38.30	41
45	3.7	33.00	23	8.5	88.49	50
47	6.5	62.67	15	7.7	61.78	45
48	6.7	66.10	20	5.9	50.90	47
50	5.2	54.43	22	9.6	94.01	52
55	7.7	86.44	22	7.6	91.43	58
60	5.2	56.69	16	10.3	107.32	60
63	5.4	65.50	25	11.4	144.58	60
66	6.8	82.26	15	12.5	149.03	67

Çizelge 3.1’de bulunan veriler farklı kasislerden gerçek zamanlı olarak alınmıştır. Buradaki veriler sadece kasis bilgilerini içerecek şekilde düzenlenerek kasise girmeden önceki maksimum hızlar ortalama 30, 40 50 ve 60 olacak şekilde seçilmiştir. Kasisin etkisi aracın maksimum hızından yavaşlayarak kasise girmesi ve kasisten çıkınca eski hızına ulaşması olarak planlanmıştır.

3.2. Aracın Özellikleri

Çalışmada Ford Fiesta marka 2009 model bir araç kullanılmıştır. Araca ait teknik bilgiler Çizelge 3.2’de verilmiştir (Anonim, 2021b). Aracın yakıt tüketimini hesaplamak ve hız verilerine ulaşmak için OBD-II arayüzünden CAN bus veri yoluna istemler gönderilmiş, CAN hattından gönderilen hız ve MAF (Manifold Air Flow) verileri çalışmada kullanılmıştır.

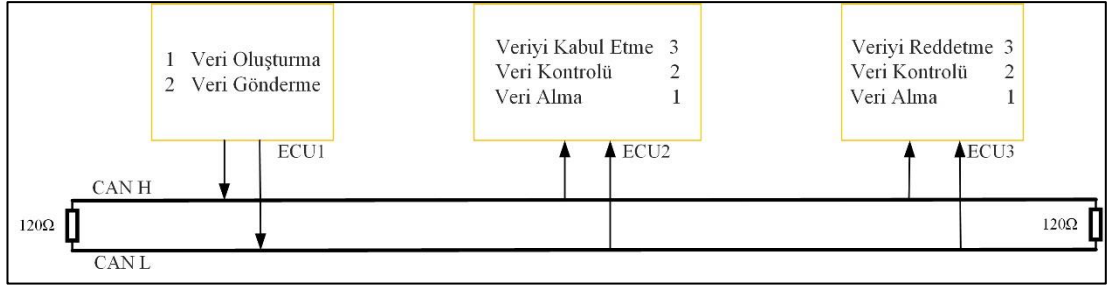
Çizelge 3.2. Aracın teknik özellikleri ve yakıt bilgisi

Teknik özellikler	
Çekiş	Önden çekiş
Ağırlık	1 080 kg
Yük Kapasitesi	1 570 kg
Lastik Boyutları	195/50/15
Motor Hacmi	1.4 L (68hp)
Maksimum Hız	164 km/sa
0-100 Km Hızlanma	15 sn
Silindir Sayısı	4
Valf Sayısı	8
Turbo	Var
Yakıt Bilgileri	
Yakıt Türü	Dizel
Yakıt Tüketim (Şehir İçi)	5.3 L
Yakıt Tüketim (Şehir Dışı)	3.7 L
Yakıt Tüketim (Ortalama)	4.3 L
CO ₂ Salınımı	116 G/Km

3.3. CAN Veri Yolu

CAN, mikrodnetleyiciler ve protokolü destekleyen elektronik cihazlar arasında merkezi sisteme ihtiyaç duymadan çift yönlü seri haberleşme protokolüdür. CAN otomotiv sektöründe araç içi iletişimi sağlamak için Robert Bosch tarafından 1980’li yılların sonunda geliştirilmiştir. Günümüzde endüstriyel otomasyon, otomotiv ve mobil cihaz sektöründe yaygın olarak kullanılmaktadır (Bozda vd., 2020).

CAN veri yolu ile ilgili standartlar Uluslararası Standardizasyon Örgütü (ISO - International Organization for Standardization) tarafından belirlenmiştir. CAN sistemlerinde tek bir noktadan veriyi gönderip, araçta bulunan dağıtık sistemlerinin her birisinin ayrı ayrı algılayabileceği şekilde çoklu algılama gerçekleştirebilmektedir. Aynı zamanda iletim hızını ayarlayarak yüksek iletim hızına ulaşabilmektedir. CAN sistemlerini ön plana çıkaran özellikler gürültü frekanslarını engelleyebilmesi, 1 Mbps’ye kadar sinyal aktarım hızına erişmesi, mesaj içeriklerinin tek bir noktadan birçok noktaya kayıpsız aktarılabilmesi ve otomatik hata teşhis ve onarma özelliğine sahip olmasıdır (Güçlü, 2018). CAN sisteminin yapısı Şekil 3.3’te verilmiştir.



Şekil 3.3. CAN veri yolu çalışma sistemi

Şekil 3.3’de CAN haberleşmesi CAN-H ve CAN-L olmak üzere iki kablodan oluşmaktadır. Araçta bulunan bütün sistemler bu iki kabloya bağlanarak birbirleri ile haberleşmektedir. Araçtaki sistemler hattı sürekli dinlemektedir. Kendilerini ilgilendiren mesaj hatta bulunduğu zaman bu mesajı alıp; hatta cevabı göndermektedir. Gönderilen mesaj tek bir sistemi ilgilendirebildiği gibi birden çok sistemi de ilgilendirebilmektedir (Güçlü, 2018).

Araçtan veri alınırken kullanılan parametreler tanımlayıcı (Identifier), OBD-II modları, PID ve Veri Baytları olarak sıralanabilir. Bu parametreler aşağıda açıklanmıştır.

Tanımlayıcı (Identifier): Tanımlayıcı biti istek ve cevap mesajlarını ayırt etmek için kullanılır. Araçtan veri okuyabilmek için; araca 0x7DF ID’si ile istek yapılır. Buna karşılık olarak ise araçtan 0x7E8 ID ile cevap alınır.

OBD-II Modları: OBD-II ye ait 10 farklı mod türü vardır (Rimpas vd., 2020). Bu mod türleri Çizelge 3.3’te gösterilmiştir. Tez çalışmasında gerçek zamanlı verilerin kullanılması için mod 01 kullanılmıştır.

Çizelge 3.3. OBD-II modları

Mod(hex)	Açıklama
0x01	Mevcut veriler gösterilir.
0x02	Dondurulmuş çerçeve verileri gösterilmektedir.
0x03	Saklanan tanılama hata kodları gösterilmektedir.
0x04	Diagnostik hata kodları ve saklanan değerler temizlenir.
0x05	Test sonuçları ve oksijen sensörü gözlemlenmektedir.
0x06	Test sonuçları ve diğer bileşenler izlenmektedir.
0x07	Bekleyen diagnostik hata kodları gösterilmektedir
0x08	Yerleşik sistemin kontrol çalışması gösterilmektedir
0x09	Araç bilgisi istenmektedir
0x0A	Kalıcı diagnostik hata kodları

PID: OBD-II PID değerleri SAE J1979 tarafından tanımlanmıştır (Chen vd., 2015). Her ECU biriminin kendine özgü bir PID değeri vardır. Bu PID değerleri heksadesimal olarak kodlanmıştır. Aracın OBD-II portuna araç için anlamlı olan verileri dinlemek için bağlantı yapılabilir. İstem yapılan her PID değerine aracın CAN veri yolundan, yanıt ID'si ve yapılan istemle ilgili verilerin elde edilebileceği faydalı bilgilerle cevap verilir. Bu tez kapsamında hız için 0x0D ve MAF 0x10 PID değerleri kullanılmıştır.

Veri Baytları: Gerçek zamanlı olarak verileri dağıtmak ve yönetmek için veri baytları kullanılır. Araçtan istek yapmak için 8 bayt veri gönderilir. Bu verinin ikinci baytına istek yapılan PID yazılmalıdır. Buna karşılık araçta sekiz bayt uzunluğunda veriyle cevap alınmaktadır. Sekiz baytlık verinin üçüncü (A), dördüncü (B), beşinci (C) ve altıncı (D) baytları cevap baytlarıdır. Bu veriler Şekil 3.4'te gösterilmiştir.

İstek ID	CAN İstek Verisi							
0x7DF	0 0x02	1 0x01 mod	2 0x0D PID	3 0x55	4 0x55	5 0x55	6 0x55	7 0x55
Cevap ID	CAN Cevap Verisi							
0x7E8	0 0x03	1 0x41	2 0x04	3 A	4 B	5 C	6 D	7 0xAA

Şekil 3.4. CAN bus verileri

3.4. OBD-II

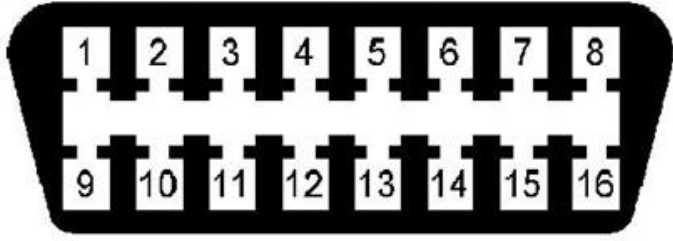
OBD (On-Board Diagnostics - yerleşik teşhis) araçta bulunan bütün elektronik aksamaların birbiriyle haberleşmesini sağlayan sisteme erişmek için kullanılan standarttır. Bu standart, araçta bulunan statik durumlar ve elektriksel (sensör değerleri) değişimleri teşhis ve raporlama işlemlerini gerçekleştirmektedir. Böylece araç hakkında ve araçtaki birçok alt sistemlerin durumuna erişerek araç durumu hakkında tespitler yapmamıza olanak sağlamaktadır. Bu erişim sayesinde araç kullanıcıları ve teknik servis çalışanları araç hakkında daha fazla bilgi sahibi olmaktadır (Amarasinghe vd., 2015).

OBD sistemleri ilk olarak 1980'lerde emisyon düzenlemeleri için kullanılmaya başlanmıştır. Otomotiv üreticileri, araç sistemlerini izleme ihtiyacının farkına

varmışlar ve motorun çalışma parametrelerini emisyon kontrolünü sağlamak için optimize etmişlerdir.

Yerleşik teşhis sistemlerinin ilk nesli OBD-I'dir. OBD-I sistemleri, üreticiler arasında nispeten ilkel ve tutarsız olarak değerlendirilmiştir. OBD-II 1990'ların ortalarında tanıtılmıştır. OBD-II sistemleri, öncekilerden daha karmaşıktır ve motor izleme ve kontrol yetenekleri daha gelişmiştir. 1996 yılından itibaren Amerika'da üretilen araçlarda zorunlu hale getirilmiştir. OBD-II standartları ilk olarak SAE (Society of Automotive Engineers) tarafından geliştirilmiş ve daha sonra EPA (Environmental Protection Agency) ve CARB (California Hava Kaynakları Kurulu- California Air Resources Board) tarafından kabul edilmiştir (Karanfil, 2020).

OBD-II sistemleri standart bir teşhis konektörüne sahiptir ve standart elektrik sinyali protokollerini ve mesajlaşma formatlarını kullanmaktadırlar. OBD-II uyumlu bir araç, SAE J1850 PWM, SAE J1850 VPW, ISO9141-2, ISO14230-4 (KWP2000) veya ISO 15765-4/SAE J2480 (CAN) olan beş iletişim protokolünden herhangi birini kullanabilmektedir. Konektörde aracın ECU'suna erişim için iletişim pinlerinin yanı sıra güç kaynağı pinleri bulunur (Vilgenoğlu, 2019). SAE J1962 standardına göre konektörün pin çıkışları Şekil 3.5'te verilmiştir (Anonim, 2021c).



PIN	DESCRIPTION	PIN	DESCRIPTION
1	Vendor Option	9	Vendor Option
2	J1850 Bus +	10	j1850 BUS
3	Vendor Option	11	Vendor Option
4	Chassis Ground	12	Vendor Option
5	Signal Ground	13	Vendor Option
6	CAN (J-2234) High	14	CAN (J-2234) Low
7	ISO 9141-2 K-Line	15	ISO 9141-2 Low
8	Vendor Option	16	Battery Power

Şekil 3.5. OBD-II konektörü ve pin çıkışları

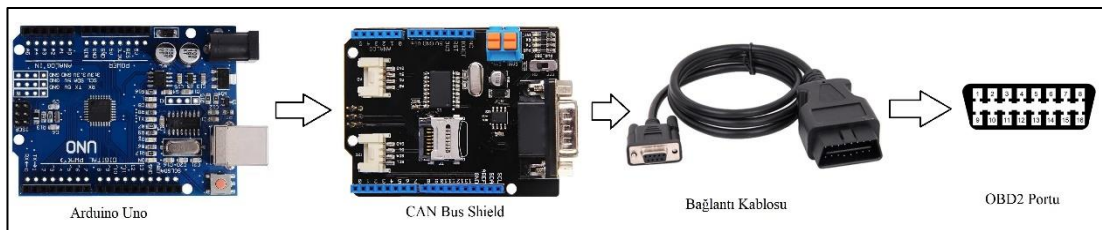
OBD-II'de arıza teşhis kodları tüm araç üreticileri için aynıdır, ancak her üretici kendi elektronik sistemleri hakkında ek bilgi sağlamak için kendi gelişmiş kodlarını ekleyebilmektedir. Ayrıca üreticilerin uymaları gereken ECU ile haberleşme, Teşhis hata kodları (DTC - Diagnostic Trouble Code) ve Parametre Kimlikleri (PID-Parameter ID) olmak üzere üç zorunlu kural kullanılmaktadır (Sim vd., 2014).

ECU'lar CAN paketleri göndererek birbirleriyle iletişim kurar. Bu tür paketlere, veri yolu üzerindeki tüm bileşenlere ulaşır ve her bir bileşen kendilerine yönelik olup olmadığına karar verebilmektedir (Martinelli vd., 2020).

Teşhis hata kodları (DTC): Araçtaki bilgisayar teşhis sisteminin sahip olduğu kodlardır. Sistemin bir aracın içinden ne tür bir sorun algılayabildiğine bağlı olarak sistem belirli bir kod görüntülemektedir. Araç sahiplerinin araçla ilgili sorunları ve aracın sorununun nerede olduğunu anlamalarına yardımcı olmak için kullanılmaktadır.

3.5. Gömülü Sistem

Yerleşik gömülü sistem CAN bus shield modülünden oluşmaktadır. Bu modül, ATmega328 mikrodenetleyici üzerine kurulu bir Arduino Uno kartı tarafından programlanıp ve kontrol edilebilir. Arduino Uno 16MHz saat frekansında çalışır ve 2 Kb Statik Rastgele Erişim Belleğine (SRAM - Static Random Access Memory) ve 1Kb Elektriksel Olarak Silinebilir Programlanabilir Salt Okunur Belleğe (EEPROM - Electronically Erasable Programmable Read-Only Memory) sahiptir. Arduino Uno hem CAN bus shield ile RX ve TX pinleri üzerinden ile haberleşmektedir (Banzi vd., 2022). Şekil 3.6'da gösterilen gömülü sistemin kullanılan devre elemanlarını göstermektedir.



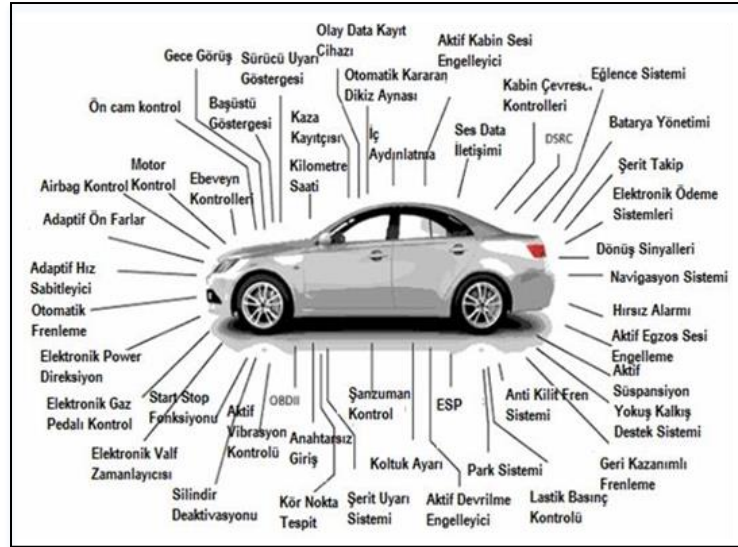
Şekil 3.6. Gömülü sistem bağlantısı

Tez çalışmasında araç hızı, MAF verileri almak için Şekil 3.6’da gösterilen Arduino ve CAN bus shield kullanılmıştır.

CAN bus Shield; araçlardan veri almak için kullanılan, orta hızlı ve güvenilir bir haberleşme kartıdır. Kart üzerinde MCP2515 CAN bus kontrolcü entegresi ve MCP2551 CAN alıcı/verici entegresi bulunmaktadır (Amarasinghe vd., 2015). CAN bus shield Arduino ile aracın ECU'sunun haberleşmesini sağlar. CAN bus Shield ile aracın motor devri, MAF, gaz pedalı pozisyonu vb. gibi birçok veriye anlık olarak ulaşabilmek mümkündür. Bu veriler Arduino ile alınıp çeşitli hesaplamalarla anlamlı hale getirilir. CAN bağlantısı DB9’dan OBD-II’ye dönüştürücü kablosu ile sağlanmaktadır. Kart üzerinde ayrıca seri LCD bağlantısı, joystick, mikro SD kart yuvası ve EM506 GPS modülü bağlantıları bulunmaktadır. Arduino ile SPI (Serial Peripheral Interface) haberleşme arayüzünü kullanarak haberleşmektedir.

3.6. Araçtan Veri Alınması

Modern araçlarda ölçüm ve kontrol için birçok elektronik sensör bulunmakta ve bu sensörler yardımıyla birçok veri üretilmektedir. Bu araçlarda bulunan elektronik yükler Şekil 3.7’de gösterilmiştir (Derse, 2017). Araçlardaki sensörler ECU tarafından kontrol edilir. Araç ve bilgisayar performansındaki gelişmeler sonucunda ECU, otomatik şanzıman, tahrik sistemi, fren sistemi ve direksiyon sistemi gibi aracın tüm parçalarını kontrol edebilmektedir. Bu elektronik teşhis sistemleri gelişmiş ve yakın zamanda OBD-II adı verilen standart bir teşhis sistemi olarak kurulmuştur (Baek ve Jang, 2015).



Şekil 3.7. Modern araçlardaki elektronik yükler

Tez çalışmasında araçtaki yakıt tüketiminin belirlenmesi amacıyla hız ve MAF değerleri kullanılmıştır. Bu veriler heksadesimal formatta araçta üretilir. OBD mod1 kullanılarak istek yapılması neticesinde anlık değerler okunabilir. İstek yapılan verilerin her birisinin kendisine ait PID numarası vardır.

Araçtan anlık olarak hız verisine ulaşmak için OBD-II üzerinden 0x0D PID'si ile istek yapılır. Araçtan bu isteğe cevap olarak gönderilen 8 baytlık verinin dördüncü baytı (A) aracın km/sa biriminde anlık hız verisidir. Verilen cevap 0 ile 255 arasında bir değer alabilir.

Araçta bulunan kütle hava akış sensöründen elde edilen verilerle MAF değerini hesaplamak mümkündür. Bunun için OBD-II üzerinden 0x10 PID'si ile istek yapılır. Aracın bu isteğe sekiz baytlık veri ile cevap verir. Döndürülen bu verinin dördüncü ve beşinci baytları (A ve B) ile Denklem (3.1)'de gösterildiği gibi MAF değeri hesaplanır. Hesaplanan değerin birimi gram/sn olup 0 ile 655 arasındadır.

$$MAF = \frac{256A + B}{100} (gram/sn) \quad (3.1)$$

3.7. Yakıt Tüketiminin Anlık Hesaplanması

Yakıt tüketimi genellikle, 100 kilometrelik sürüş boyunca tüketilen yakıtın litre cinsinden ifade edilmesi ile bulunmaktadır. Tez çalışmasında yakıt tüketiminin anlık hesaplanmasında dizel motorlu araç kullanılmıştır. 100 kilometrede yakıt tüketimini hesaplamak için kullanılan formül Denklem (3.2)'de verilmiştir (Meseguer, 2015).

$$Yakıt\ Tüketimi\left(\frac{l}{100km}\right) = \frac{Yakıt\ Debisi\left(\frac{l}{sa}\right)}{Hız\left(\frac{km}{sa}\right)} \quad (3.2)$$

Anlık yakıt tüketimi ise, sadece araç hareket halindeyken ve motor çalışır durumdayken anlık yakıt debisi ve aracın hızından hesaplanmaktadır. Yakıt debisi ve aracın anlık hızını hesaplamak için aracın OBD-II ara yüzünden yararlanılmaktadır. OBD-II yakıt tüketimi verisini doğrudan göndermemektedir. Bunun yerine yakıt tüketimi hesaplanmasında kullanılabilecek diğer parametreler OBD-II'den alınabilir. Bu parametreler Motor Yakıt Oranı (0x5E) ile birlikte aracın hız bilgisi veya MAF (0x10) değeri veya emme manifoldu basıncı (0x0B) ve emme havası sıcaklığıdır (0x0F) (Meseguer, 2015). Bu çalışmada yakıt tüketimini hesaplamak için denklem (3.3)'te gösterildiği gibi MAF parametresi kullanılmıştır.

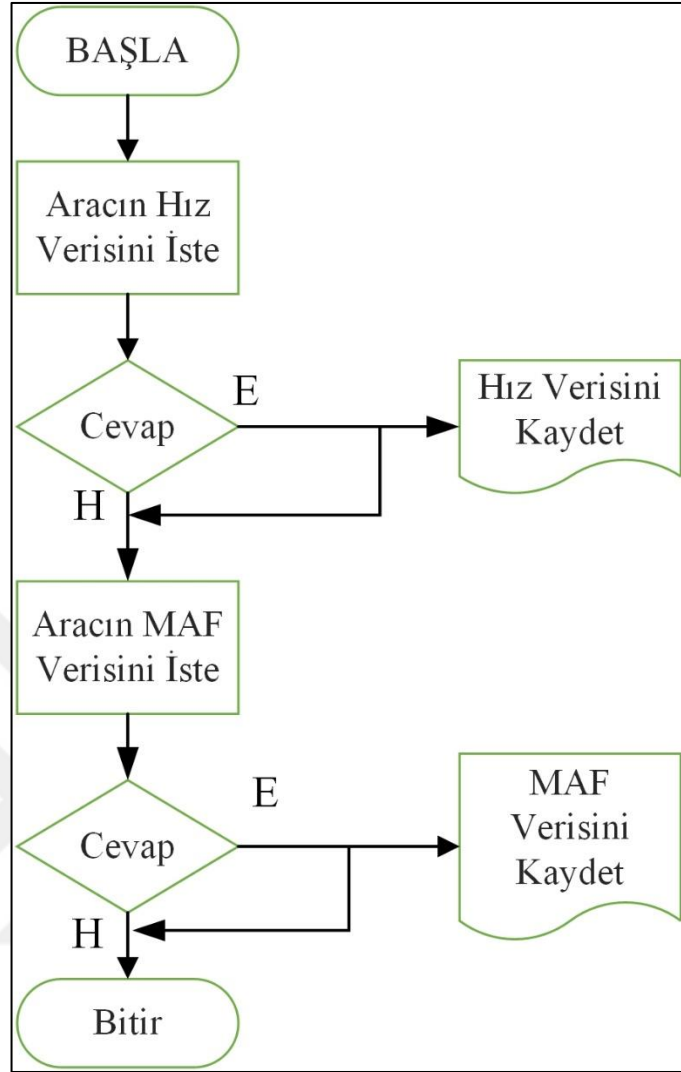
$$Yakıt\ Tüketimi\left(\frac{l}{sa}\right) = \frac{MAF \cdot 3600}{AFR_A \cdot FD} \quad (3.3)$$

Denklem (3.3)'te MAF: kütle hava akışı (g/s), **AFR_A**: gerçek hava-yakıt oranı (1 gram yakıtı yakmak için gerekli hava miktarı) ve FD: yakıt yoğunluğunu (g/l) temsil etmektedir. **AFR_A** ve FD değerleri Çizelge 3.4'te verilmiştir.

Çizelge 3.4. İdeal hava/yakıt oranı ve yoğunluk değerleri

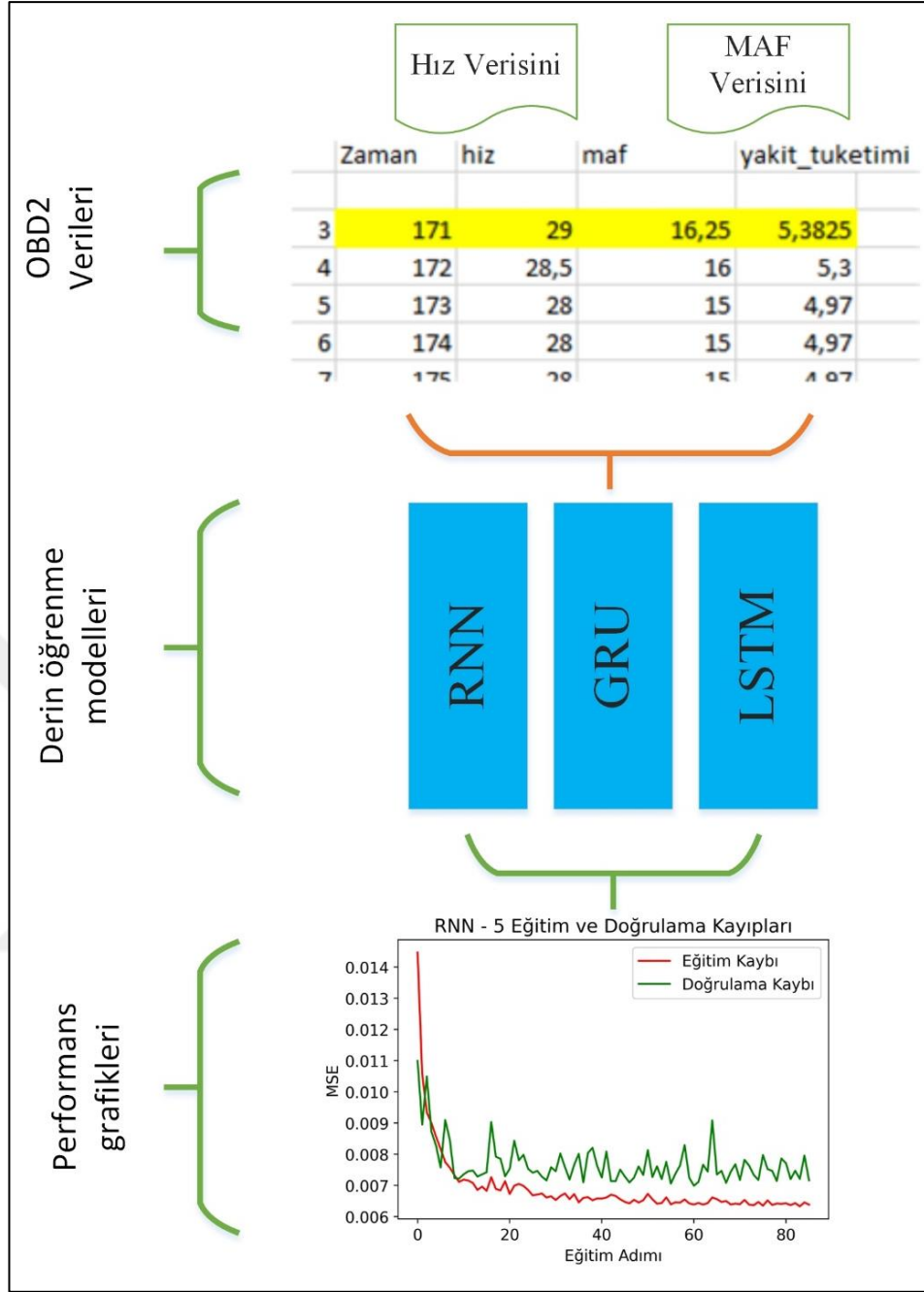
Yakıt Tipi	Gerçek Hava-Yakıt Oranı	Yakıt Yoğunluğu (g/l)
Benzin	14.7	820
Dizel	14.5	750

Arduino ve Can bus shield ile hareket halindeki araçtan gerçek zamanlı olarak hız ve MAF değerleri alınmaktadır. Yapılan işlemin akış diyagramı Şekil 3.8'de gösterilmiştir.



Şekil 3.8. Veri toplama akış diyagramı

Şekil 3.9’da elde edilen veriler ve kullanılan derin öğrenme modelleri gösterilmiştir. Öncelikle OBD-II’den elde edilen veriler düzenlenir ve kullanılacak derin öğrenme yöntemleri için zaman serisi formatına çevrilir. Mevcut adım ve öncesindeki adımlar arasındaki ilişkiyi doğru kurgulayabilmek için yenilemeli sinir ağlarından RNN, GRU ve LSTM modelleri kullanılmıştır. Eğitilen modeller farklı senaryoların tahmin edilmesinde kullanılır.



Şekil 3.9. Geliştirilen sistemin basamakları

3.8. Veri Setinin Hazırlanması

Araçtan alınan veriler Şekil 3.10’da gösterilmektedir. Burada m: milisaniye, M: Kütle hava akış sensörü (MAF), y: Yakıt tüketimi, h: aracın hız bilgisine karşılık gelmektedir.

```
m:7819 h:14
---
m:7844 M:14.10 y:4.67
---
m:7936 h:14
---
m:7961 M:14.22 y:4.71
---
m:7999 M:14.33 y:4.74
---
m:8115 h:14
---
m:8141 M:13.76 y:4.55
---
m:8195 h:14
```

Şekil 3.10. Araçtan alınan ham veriler

Şekil 3.10’da da görüldüğü gibi veriler düzenli zaman aralıklarında alınamamaktadır. Dolayısıyla ardışık iki veri arasındaki süreler farklı olabilmektedir. Makine öğrenmesinde kullanılacak verilerin eşit gecikmelerle elde edilmesi gerektiğinden veri setinde olmayan ms’lerdeki hız ve yakıt tüketimleri için uygun değerlerin belirlenmesi gerekmektedir. Bu işlem için doğrusal interpolasyon yöntemi kullanılmıştır.

Doğrusal interpolasyon yönteminde veri setinde bilinen iki değer arasında kalan üçüncü bir değeri bulmak için veriler koordinat düzlemine aktarılır. Dolayısıyla iki noktası bilinen doğru denklemi elde edilir. Bu değerler arasında kalan üçüncü değerin bu doğru üzerinde olduğu var sayılarak üçüncü değer için uygun bir değer belirlenir.

(x_0, y_0) ve (x_1, y_1) doğrunun bilinen iki noktası ise bu doğru denklemi Denklem (3.4) kullanılarak bulunabilir.

$$\frac{y - y_0}{x - x_0} = \frac{y_1 - y_0}{x_1 - x_0} \quad (3.4)$$

Denklem (3.4) y değeri için çözülürse Denklem (3.5)'de gösterilen ifade elde edilmiş olur.

$$y = y_0 + (x - x_0) \frac{y_1 - y_0}{x_1 - x_0} \quad (3.5)$$

Bu çalışmada Denklem (3.5)'de verilen x milisaniyedir. y ise hız veya yakıt tüketimi için kullanılır. Denklem (3.5) kullanılarak boş milisaniyeler orantısal olarak doldurulur.

Doldurulan verilerden veri seti elde etmek için belirli aralıklarla örnekleme yapılması gerekmektedir. Bu tezde örnekleme zamanı 300 milisaniyede olarak belirlenmiştir. Her 300 ms'de yakıt tüketimi ve hız verisinin belirlenmesi için veri setinde araçtan alınan ve interpolasyon ile elde edilen 149 önce ve 150 sonraki verinin ortalaması alınarak veri seti hazırlama işlemi tamamlanmıştır.

yakit	hiz	dosya	millis
2,3	8,0	0,0	1001,0
3,0	8,0	0,0	1001,3
5,4	8,4	0,0	1001,6
4,5	9,8	0,0	1001,9
5,2	11,5	0,0	1002,2
6,4	12,6	0,0	1002,5
6,4	14,1	0,0	1002,8
6,4	15,1	0,0	1003,1
6,5	16,0	0,0	1003,4
6,5	16,4	0,0	1003,7

Şekil 3.11. Hazırlanan veri seti

Şekil 3.11'de hazırlan veri setinin son hali gösterilmiştir. Veri setinde her 300 ms'de bir adet hız ve yakıt tüketimi değeri bulunmaktadır.

3.9. Sistemin Eğitilmesi için Ön İşlenmesi

Veriler düzenlendikten sonra verilerin ardışıl halde getirilmesi, eğitim ve test için bölünmesi ve normalizasyon işlemleri yapılmıştır.

3.9.1. Pencere kaydırma (window slide) yöntemi

X n adet hız verisini Y ise bunlara karşılık gelen n adet yakıt tüketimini göstermek üzere $X = \{x_0, x_1, \dots, x_{n-1}\}$, $Y = \{y_0, y_1, \dots, y_{n-1}\}$ şeklinde gösterilebilir. t zamanında yakıt verisi sadece x_t verisine değil t'den önceki verilere de bağlı olduğundan verilerin eğitilmesinde yineleyen derin öğrenme algoritmaları kullanılmıştır. Veriler ise pencere kaydırma yöntemi ile düzenlenmiştir. m pencere boyutu olmak üzere y_t 'yi tahmin etmek için $X^{(t)} = \{x_{t-m+1}, \dots, x_{t-1}, x_t\}$ kullanılır.

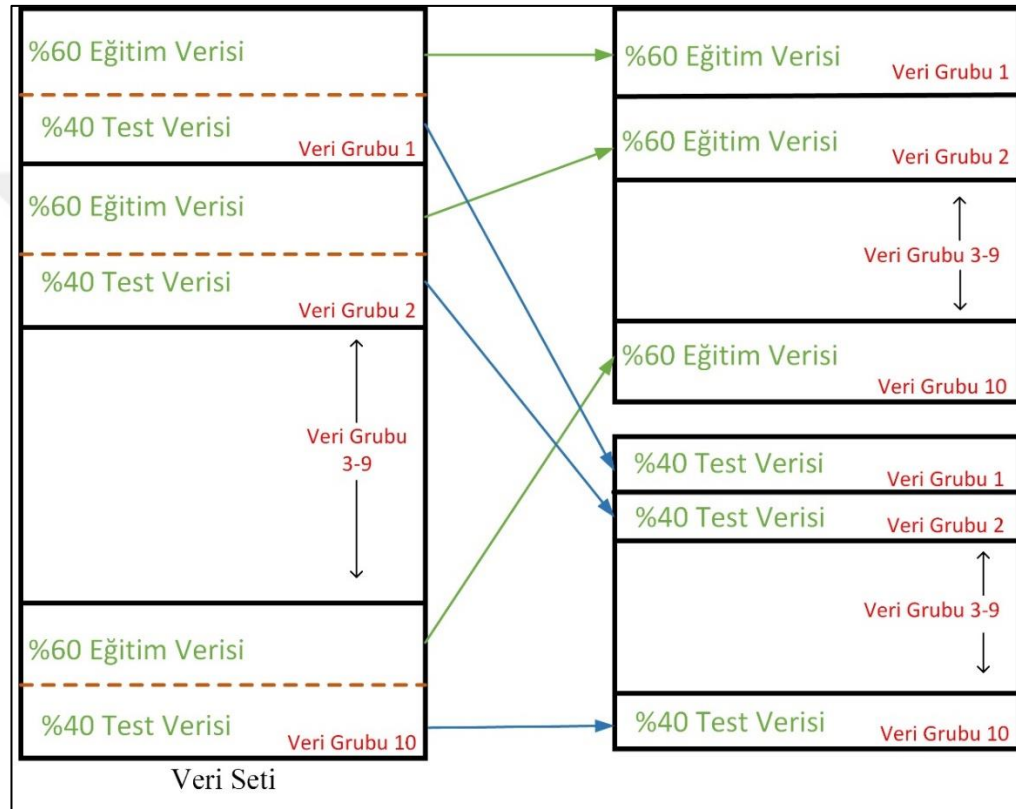
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Hız (X)	19,0	18,4	18,0	18,0	17,6	16,6	15,4	14,9	14,1	14,0	14,6	15,1	16,0	16,2	17,0
Yakıt Tüketimi (Y)	3,4	3,2	3,0	2,6	3,8	4,5	5,8	7,1	6,6	5,7	5,4	5,7	5,6	5,8	5,9
Hız (X)	19,0	18,4	18,0	18,0	17,6	16,6	15,4	14,9	14,1	14,0	14,6	15,1	16,0	16,2	17,0
Yakıt Tüketimi (Y)	3,4	3,2	3,0	2,6	3,8	4,5	5,8	7,1	6,6	5,7	5,4	5,7	5,6	5,8	5,9
Hız (X)	19,0	18,4	18,0	18,0	17,6	16,6	15,4	14,9	14,1	14,0	14,6	15,1	16,0	16,2	17,0
Yakıt Tüketimi (Y)	3,4	3,2	3,0	2,6	3,8	4,5	5,8	7,1	6,6	5,7	5,4	5,7	5,6	5,8	5,9

Şekil 3.12. Pencere kaydırma metodu

Şekil 3.12’de m=10 için pencere kaydırma yöntemi gösterilmiştir. $X^{(9)}$ ilk on hız verisi için kullanılan vektördür ve y_9 $X^{(9)}$ ’un çıktısı olarak düşünülür. Sonraki X vektörlerini ve karşılık gelen y değerini elde etmek için pencere her defasında bir basamak kaydırılır. Dolayısı n tane veri için m pencere boyutu olmak üzere input matrisinin boyutu $(n - m + 1) \times m$ olur. Tez de m pencere boyutları 5, 10 ve 15 olarak seçilmiştir.

3.9.2. Veri setinin eğitim ve test setleri için bölünmesi

Araçtan OBD-II arayüzünden veriler 3 farklı sürüşte alınmıştır. Bu veri grupları Bölüm 3.9.1’de ifade edildiği gibi ön işleme tabi tutulmuştur. Her 3 veri grubu kendi içinde 10 eşit kısma ayrılmıştır. Her kısım ise kendi içinde %60’ı eğitim ve %40’ı test için bölünmüştür. Şekil 3.13’te veri setinin on kısma bölünmesi ve her kısmın eğitim ve test verisi olarak ayrılması gösterilmiştir.



Şekil 3.13. Veri setinin eğitim ve test verisi olarak ayrılması

3.9.3. Normalizasyon (minmax scaler)

Makine öğrenmesi algoritmalarında kullanılan verilerin aralığının farklı olması algoritmanın başarısının olumsuz etkilediği için makine öğrenmesi algoritmalarında veriler normalize edilir. Bu çalışmada da hız ve yakıt tüketimi verileri arasındaki ölçek farklılıkları bulunmaktadır. Hız verileri 0 ile 59 arasında değişirken yakıt tüketimi ise; 2.1 ile 21.7 arasında değişmektedir. Veriler normlize edilerek 0 ile 1 aralığına indirgenir.

Eğitim setine Denklem (3.6)'da verilen denklem kullanılarak normalizasyon işlemi uygulanmıştır.

$$x_{i_norm} = \frac{x_i - \min(X)}{\max(X) - \min(X)} \quad (3.6)$$

Denklem (3.6)'da tanımlanan x_{i_norm} x_i 'nin normalize edilmiş değerini, $\min(X)$, $\max(X)$ X verilerindeki en büyük ve en küçük değerleri göstermektedir.

3.10. Yapay Zekâ

Yapay zekâ insanlara ait öğrenme, problem çözme ve tahminde bulunma gibi öz davranışların bilgisayarlar tarafından gerçekleştirilmesi olarak tanımlanmaktadır. Bilgisayarların insanlar gibi öğrenmesi, algılayabilmesi ve hareket etmesi çalışmaları 1950'li yıllardan başlayarak bilim insanlarının önemli araştırma konuları arasında yer almıştır. Tez kapsamında makine öğrenme algoritmasının bir alt dalı olan derin öğrenme yöntemleri sıklıkla kullanılmaktadır (Chollet, 2018). İnsanlar tarafından rutin olarak yapılan iş ve görevlerin bilgisayarlara öğretilerek daha hızlı ve standart bir şekilde gerçekleştirilmesi makine öğrenmesi yöntemleri kullanılarak sağlanmaktadır.

Makine öğrenmesi özellikle mühendislik ve istatistik gibi birçok bilim dalında çok yaygın bir şekilde kullanılmaktadır (Kumar vd., 2016). Benzer şekilde meteorolojiden (McGovern vd., 2017; Kosovic vd., 2020) tarıma (Çetiner, 2021a), askeri çalışmalara (Maas, 2019; Wang vd., 2020) tıpa (Yıldız vd., 2015; Jiang vd., 2017; Macrae, 2019; Rong vd., 2020), ormandan (Cetiner vd., 2016) ekonomiye (Huang vd., 2019) kadar çok farklı bilim dallarında kullanıldığı görülmüştür.

Makine öğrenmesinin doğal dil işleme, karar destek sistemleri, bilgisayarlı görü ve veri madenciliği gibi farklı yapay zekâ alt alanlarında kullanılmaktadır. Tez çalışmasında ise makine öğrenmesinin bir alt alanı olan derin öğrenme modelleri kullanılmıştır.

3.10.1. Makine öğrenmesi

Makine öğrenmesinde bilgisayar doğrudan amaca yönelik programlanmaz fakat algoritmadaki parametreler veri seti doğrultusunda güncellenerek bilgisayarın öğrenmesi sağlanır. Dolayısıyla makine öğrenmesi, algoritmaların verileri kullanarak tahmin doğruluğunu kademeli bir şekilde artırır. Yapılan bu işlem insanların öğrenmesi ile benzerlik göstermektedir (Ongsulee, 2017; Qiu vd., 2016).

Makine öğrenmesi yöntemleri dört kategoriye ayrılmaktadır (Ongsulee, 2017). Bunlar denetimli (supervised), denetimsiz (unsupervised), yarı denetimli (semi-supervised) ve pekiştirmeli (reinforcement) öğrenme yöntemleridir. Denetimli öğrenmede algoritma giriş (Input) ve çıkış (Output) değerleri belirlenmiş verilerle eğitilir. Böylece algoritmaya giriş olarak girilen veri sonucu elde edilen değer gerçek değer ile karşılaştırılır. Hatanın minimuma indirilmesi için gerekli güncellemeler yapılarak sistemin eğitilmesi sağlanır. Denetimsiz öğrenme ise sınıf etiketlerinin belirli olmadığı veri setleri üzerinde yapılan öğrenme tekniğidir (Lalonde vd., 2001). Daha çok sınıflandırma ve kümeleme işlemlerinde kullanılır. Veri setinde etiketleme işlemi maliyeti fazla olması durumunda, veri setinin bir kısmının etiketlenmesi söz konusu olabilir. Bu durumda yarı denetimli öğrenme yöntemleri kullanılarak etiketli ve etiketsiz verilerle sistem eğitilir (Ongsulee, 2017). Pekiştirmeli öğrenme genelde robotik ve oyun gibi uygulamalarda kullanılmaktadır. Deneme yanılma yöntemi ile elde edilen kazanım miktarına göre öğrenme gerçekleştirilmektedir.

Tez çalışmasında hız ve yakıt tüketim değerleri araçtan alınarak veri seti oluşturulmuştur. Hız giriş, yakıt tüketimi ise çıkış verisi olarak kullanılmıştır. Böylece yakıt tüketim tahminleme sonuçları elde edilmiştir.

3.10.2. Derin öğrenme

Yapay sinir ağları (YSA) insan nöronlarının yapısı ve bağlantılarından esinlenilerek geliştirilmiş bir denetimli öğrenme modelidir. YSA 1950’li yıllarda bilgisayarın ortaya çıkmasından kısa bir süre sonra geliştirilmeye başlanmıştır (Aggarwal, 2018). Bununla birlikte bu yıllarda bilgisayarların veri işleme ve saklama kapasitelerinin yetersiz olmasından dolayı çok fazla ilerleme kaydedilememiştir. 1990’lı yıllardan itibaren

artan hesaplama gücü sayesinde yapay sinir ağları ile ilgili çalışmalar artış göstermiştir. Özellikle 2000’li yıllardan itibaren veri miktarı işlem gücü ve saklama kapasitesindeki artış neticesinde YSA ile ilgili yapılan çalışmaların sayısında önemli oranda bir artış gerçekleşmiştir. Günümüzde mobil telefonların insanların günlük yaşantısında çok fazla yer alması ve nesnelerin interneti gibi kavramların yaygınlaşması neticesinde çok ciddi miktarda üretilen verilerde artış yaşanmıştır. Çok büyük miktarda YSA’lar ile işlenmesinde doğruluğu arttırmak için YSA’ların katmanlarının sayıları da arttırılmıştır. Böylece çok sayıda katmana sahip bu yapılar Derin Öğrenme ile ifade edilmektedir.

Son zamanlarda problemin çeşidine göre farklı derin öğrenme yöntemleri geliştirilmiş olmakla birlikte, zaman serileri gibi ardışıl verilerin işlenmesi için genelde yineleyen derin öğrenme yapıları kullanılmaktadır. Tez kapsamında yakıt tüketim tahmini gerçekleştirmek için RNN, LSTM ve GRU derin öğrenme modelleri kullanılmıştır. Önerilen bu modelleri oluşturmak için aşağıdaki bölümlerde anlatılan derin öğrenme kavramları kullanılmıştır.

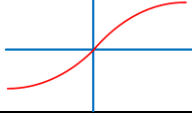
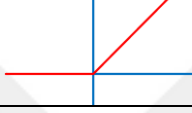
3.10.2.1. Aktivasyon fonksiyonu

Aktivasyon fonksiyonu, kendisine girdi olarak verilen değerleri belirli bir aralık değerlerine indirgenmesini sağlamaktadır. Derin öğrenme mimarileri ile eğitim sürecinde amaç hata değerini en aza indirmek olduğu için geri yayılım algoritması kullanılır. Algoritmanın hızlı çalışabilmesi için türevi kolay alınan bir aktivasyon fonksiyonu seçilmesi önemlidir (Arora vd., 2016). Literatür çalışmaları incelendiğinde Sigmoid, TanH ve ReLU gibi farklı aktivasyon fonksiyonları bulunmaktadır. Bunlardan ReLU aktivasyon fonksiyonu kolay türevlenebilmesi, hız ve doğruluk açısından diğer aktivasyonlarına göre avantaj sağladığı görülmüştür (Çetiner, 2021b).

Çizelge 3.5’te tez çalışmasında kullanılan aktivasyon fonksiyonları denklem ve türevleriyle birlikte gösterilmektedir. Sigmoid fonksiyonu girdi değerlerinin tamamı için 0 ile 1 arasında değerler üretir. Bu fonksiyonun dezavantajı ise çıkış değerleri sıfırdan uzaklaştıkça girişe daha az tepki vermeye başlar ve türevleri sıfıra yaklaşmaktadır. TanH fonksiyonu sigmoid fonksiyonundan farklı olarak -1 ile 1 aralığında değerler almaktadır. Bu fonksiyonun dezavantajı sigmoid ile aynıdır. ReLU

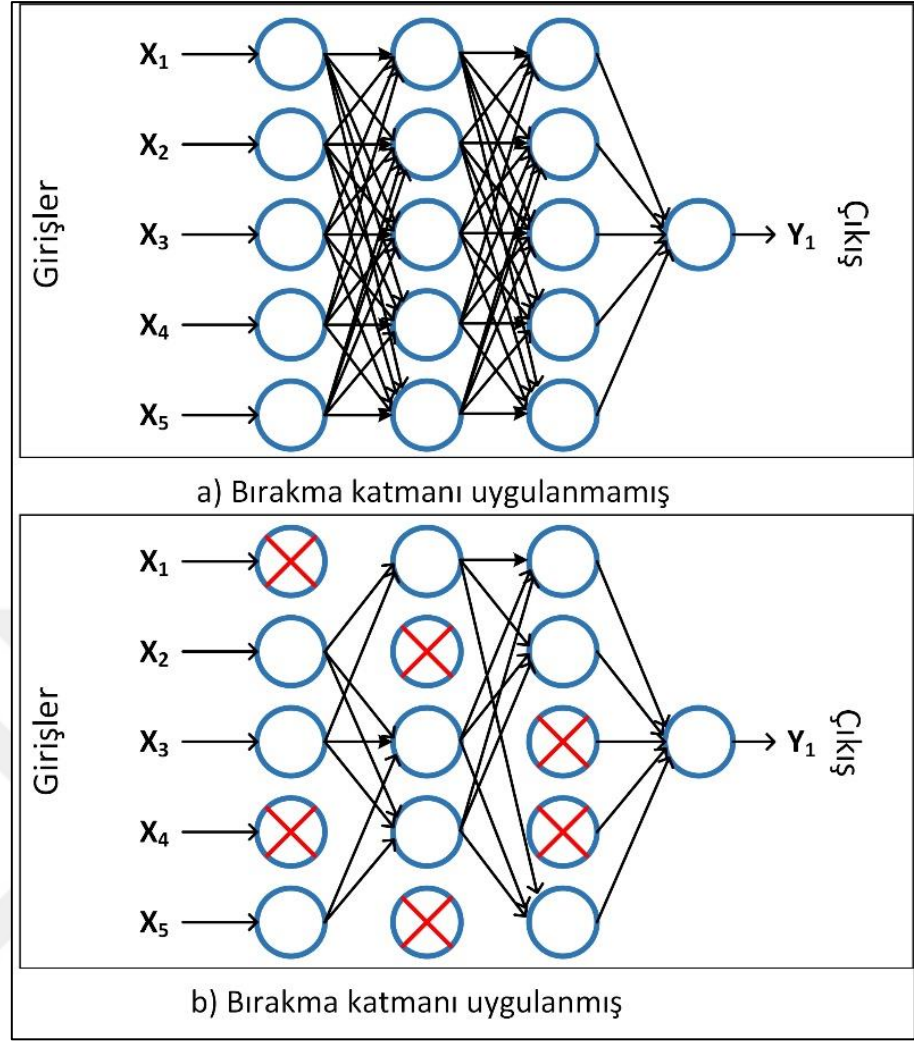
aktivasyon fonksiyonunda negatif deęerler 0 olarak döndürölmektedir. Yani kullanılan modelde negatif deęerler eęitilmedięinden hesaplama yökü daha azdır ve sistem daha hızlı cevap vermektedir.

Çizelge 3.5. Aktivasyon fonksiyonları

İsim	Grafik	Eşitlik $f(x)$	Türev $f'(x)$
Sigmoid		$\sigma = \frac{1}{1 + e^{-x}}$	$\sigma' = f(x)(1 - f(x))$
TanH		$TanH(x) = \frac{2}{1 + e^{-2x}} - 1$	$TanH'(x) = 1 - f(x)^2$
ReLU		$ReLU(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$	$ReLU'(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$

3.10.2.2. Bırakma (dropout) katmanı

Derin öęrenme tabanlı mimarilerin eęitim sürecinde model verileri ezberleyebilmektedir (Overfitting). Bu durumda modele bazı verilerin unutturulması gerekmektedir (Gal ve Ghahramani, 2016). Bu işlemleri gerçekleştirebilmek için model oluşumunda bazı katmanların arasına farklı oranlarda bırakma işleminin yapılması zorunluluk haline gelmektedir. Tez kapsamında önerilen modellere ait bazı katmanlarında bırakma katmanı eklenmiştir. Bunun sonucunda modelin yakıt tüketim verilerini ezberlemesinin önüne geçilmiştir. Şekil 3.14b’de bırakma işlemi uygulanmış sinir ağı gösterilirken, Şekil 3.14a’da ise bırakma işlemi uygulanmamış sinir ağı gösterilmektedir.



Şekil 3.14. Bırakma katmanı a) uygulanmamış, b) uygulanmış görüntü

3.10.2.3. Regression (regresyon) katmanı

Derin öğrenme algoritmalarındaki son katman olarak tanımlanan tahmin işlemini gerçekleştiren katman regresyon katmanıdır. Bu katmanda, kendisinden önceki katman olan tam bağlantı katmanından gelen veriler analiz edilerek çıkış tahmin değerleri elde edilmektedir.

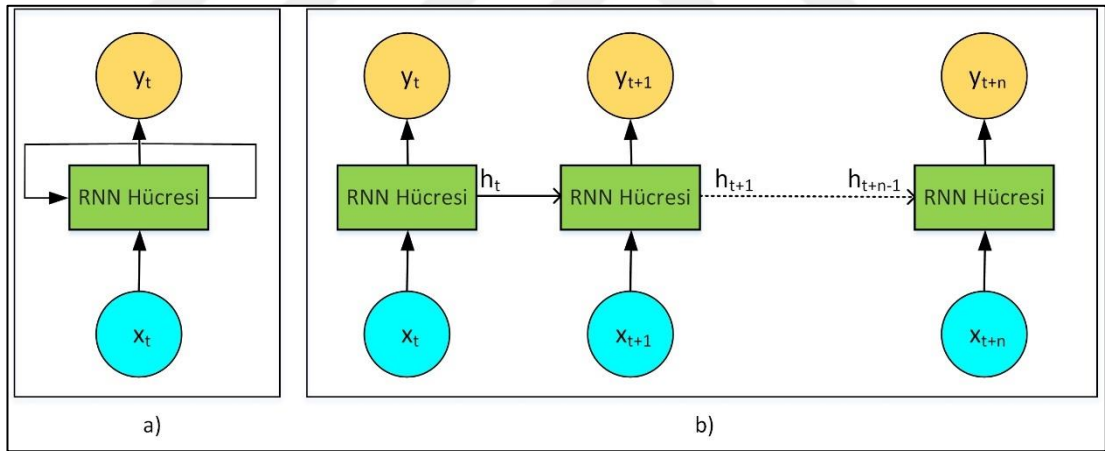
3.10.3. Kullanılan derin öğrenme yöntemleri

Tez kapsamında aracın hızına göre yakıt tüketiminin tahmin edilmesi gerçekleştirilmiştir. Aracın hızlanırken belli bir hızdaki tükettiği yakıt ile yavaşlarken aynı hızda tükettiği yakıt farklıdır. Bu durumda aracın hem hızlanırken hem de yavaşlarken yakıt tüketimini doğru değerlendirebilmek için daha önceki hızlarının

unutulmaması gerekmektedir. Bu durumda bir ya da birden fazla önceki adımları hatırlayabilen derin öğrenme yapılarına ihtiyaç duyulmaktadır. Bundan dolayı aracın hızlanıp yavaşlama durumuna göre yakıt tüketimini doğru bir şekilde tahmin edebilmek için RNN, LSTM, GRU yineleyen derin öğrenme algoritmaları kullanılmıştır. Bu algoritmalar aşağıdaki bölümlerde ayrıntılı bir şekilde ifade edilmiştir.

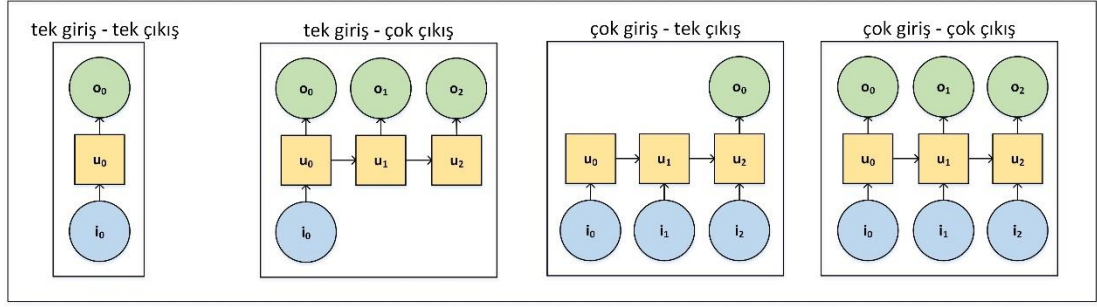
3.10.3.1. RNN (tekrarlayan sinir ağları - recurrent neural networks)

RNN'ler ardışıl verilerin veya zaman serilerinin tahmin edilmesinde kullanılan bir çeşit yapay sinir ağıdır (Salouhou, 2019). Bir RNN hücresinin yapısı Şekil 3.15a'da gösterilmiştir. Bu yapının ardışıl zamanlarda kullanılması sonucu açılmış hali Şekil 3.15b'deki gibi gösterilebilir. Bu ağa girdi olan x_t verisinden elde edilen h_t çıktısı bir sonraki (t+1) zamanında x_{t+1} ile birlikte hücreye girdi olur. Bu şekilde daha önceki zamanlardaki n ardışıl veri kullanılarak t+n zamanındaki çıktı tahmin edilir.



Şekil 3.15. RNN hücre yapısı

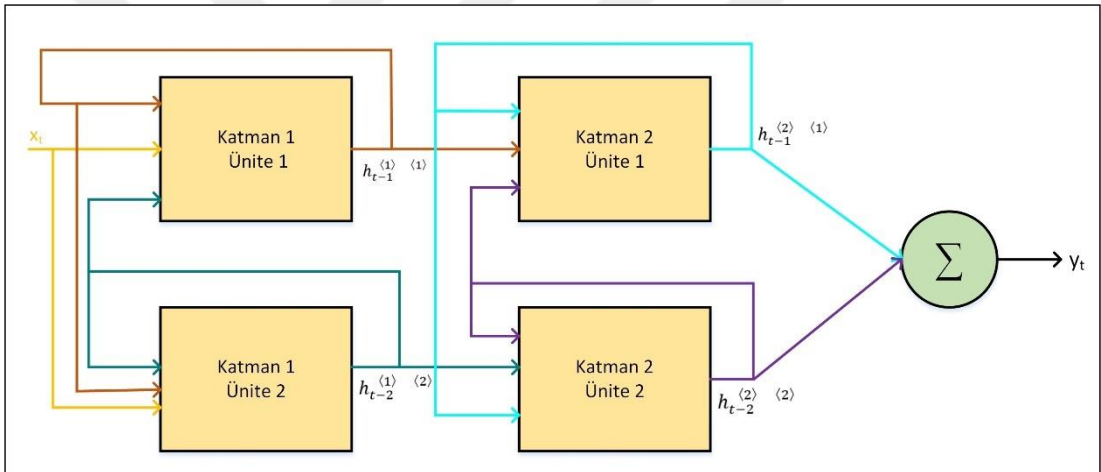
RNN'lerin farklı kullanım şekilleri bulunmaktadır. Bunları Şekil 3.16'da olduğu gibi tek giriş - tek çıkış, tek giriş - çok çıkış, çok giriş - tek çıkış, çok giriş - çok çıkış olarak sınıflandırılabilir. Yapılan tez çalışmasında çok giriş - tek çıkış türü RNN kullanılmıştır. Giriş verisi 5, 10 veya 15 adet 300 ms aralıklı ardışıl hız ve çıkış verisi ise yakıt tüketimidir.



Şekil 3.16. RNN çeşitleri

RNN hücresinde yapılan işlemin ifadesi Denklem (3.7)'de gösterilmiştir. W ağırlık, b bias, h_t hücrenin t zamanındaki çıktı verisi, x_t ise t zamanındaki girdi verisi için kullanılmıştır.

$$h_t = f(W_h h_{t-1} + W_x x_t + b) \quad (3.7)$$



Şekil 3.17. RNN hücre yapısı

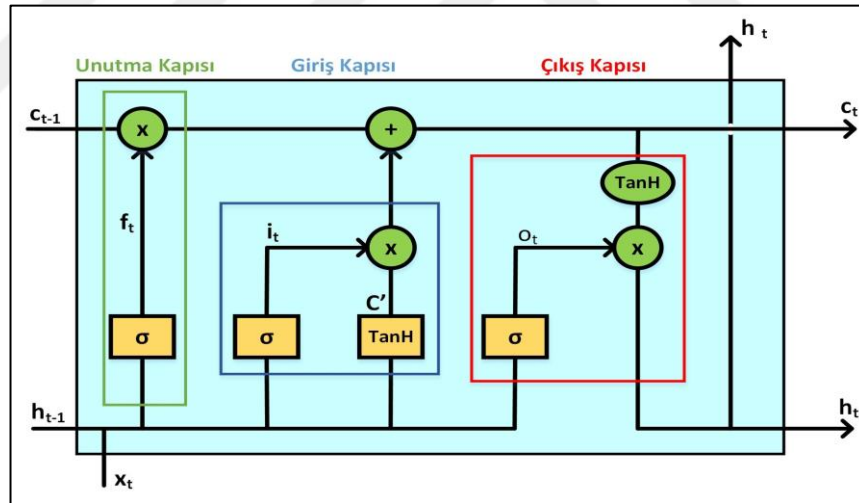
RNN sisteminde her birisinde birden fazla ünite olan RNN grupları da olabilir. Şekil 3.17'de RNN 1 1. grup, RNN 2 ise 2. Gruptur. Her bir grupta 2 ünite bulunmaktadır. Bu RNN yapısında x_t t zamanındaki girdiyi, y_t t zamanındaki çıktıyı göstermektedir. h çıktıların birinci üs indisi grup numarasını, ikinci üs indisi ise ünite numarasını belirtir. Dolayısıyla örneğin $h_{t-1}^{(2)} (1)$ $t-1$ zamanının 2. grubunun 1. ünitesinin çıktısıdır.

Yapay sinir ağlarında olduğu gibi RNN sistemlerini eğitmek için de gradyan inişi (gradient descent) ve geri yayılım algoritması kullanılır. RNN hücrelerinde zamanda geri yayılım algoritması kullanıldığında, ünite sayısının fazla olması durumunda

kaybolan veya patlayan gradyan problemlerine sebep olmaktadır (Mao vd., 2014). Bundan dolayı önceki zamanlardaki verinin sonraki zamanlara aktarılması için RNN modellerini temel alan LSTM ve GRU modelleri geliştirilmiştir.

3.10.3.2. LSTM (uzun kısa süreli bellek - long short term memory)

LSTM algoritması, RNN algoritmasının ağırlık güncellemelerinde yaşanan gradyan değerinin sıfıra yaklaşması sonucunda eğitimin kaybolması durumunu çözen bir algoritmadır. RNN ve LSTM algoritmalarının her ikisi de YSA modellerinin bir versiyonu olarak düşünülebilir. LSTM algoritması eğitim işlemi sırasında gradyan değerlerinin sıfırlanması ya da aşırı yükselmesi sonucunda oluşan hata değerlerini çözebilmek için hafıza blokları adı verilen özel bir birime sahiptir. Hafıza blokları dahil olmak üzere RNN yöntemindeki eksiklikleri gideren LSTM algoritmasının üç farklı kapısı bulunmaktadır. Bu kapılar Şekil 3.18’de gösterildiği gibi sırasıyla unutma, giriş ve çıkış kapılarıdır.



Şekil 3.18. LSTM mimarisi

Unutma kapısında veri setindeki hangi bilgilerin korunacağı ya da korunmayacağı belirlenmektedir. Girdi olan verilere, derin öğrenmenin giriş katmanında normalleştirme işlemi uygulanır. Normalleştirme işlemi sonunda veri sıfır değerine yakınsa verinin değerinin unutulması gerektiği düşünülür. Normalleştirilmiş veri bir değerine yakın ise verinin korunması sağlanmalıdır.

Bir diğer LSTM kapısı olan giriş kapısında ise önceki gizli durumdan gelen veri ile girdi verisi sigmoid aktivasyon fonksiyonuna ile işleme tabi tutulmaktadır. Unutma kapısında olduğu gibi aktivasyon işlemi sonucunda elde edilen değer bir değerine yakınsa verinin hedef çıktıyı tahminde önemli olduğu anlamına gelmektedir. Bu adım sonrasında yapılan işlemlerin daha düzenli olabilmesi için giriş kapısından ve gizli durumdan gelen veriler TanH aktivasyon fonksiyonuna iletilir. TanH ile sigmoid aktivasyon fonksiyonlarından elde edilen çıktılar hangi bilgilerin tutulup hangi bilgilerin unutulacağına karar verilebilmesi için çarpılmaktadır. Bu çarpma işlemi sonrasında elde edilen değer gizli durum olarak tanımlanmaktadır (Wang vd., 2020). Şekil 3.18’de belirtilen kapılarda kullanılan sigmoid aktivasyon fonksiyonu Denklem (3.8)’de gösterilmiştir.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (3.8)$$

Denklem (3.9), (3.10) ve (3.11)’de sırasıyla gösterilen giriş, unutma ve çıkış kapılarına ait matematiksel ifadeler gösterilmiştir. Bu kapılara ait eşitlikler sırasıyla i_t, f_t, o_t eşitlikleri ile ifade edilmiştir.

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i) \quad (3.9)$$

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f) \quad (3.10)$$

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o) \quad (3.11)$$

$$c'_t = \text{TanH}(W_{cx}x_t + W_{ch}h_{t-1} + b_c) \quad (3.12)$$

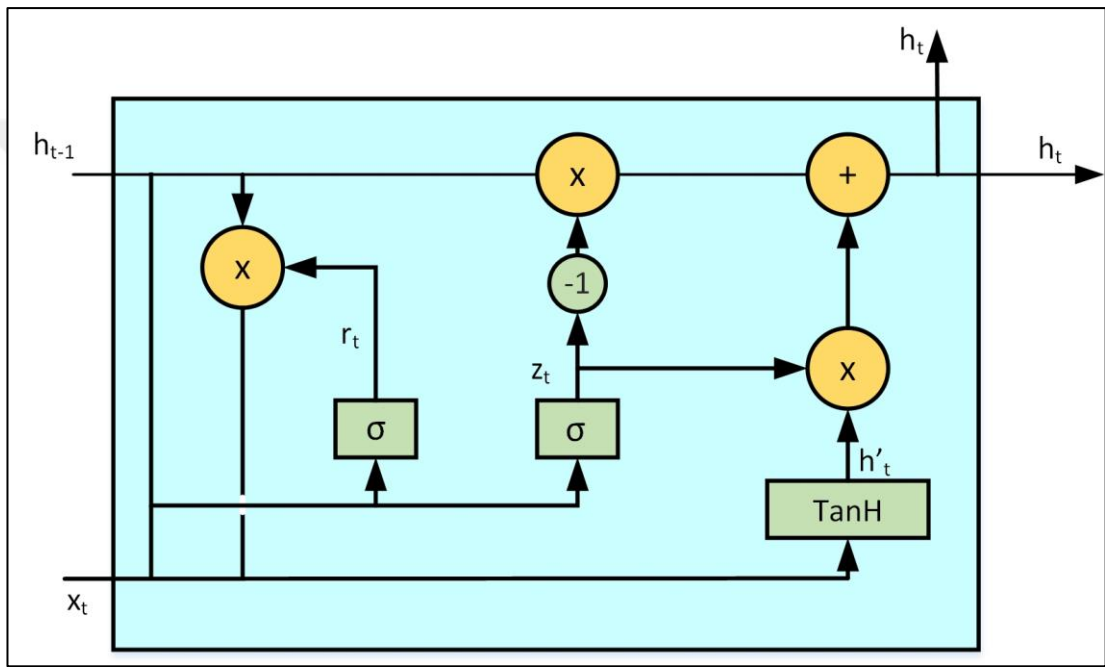
$$c_t = i_t c'_t + f_t c_{t-1} \quad (3.13)$$

$$h_t = \text{TanH} o_t c_t \quad (3.14)$$

Denklem (3.9) ile (3.14) arasında W ağırlık matrislerini, b bias parametrelerini ve Denklem (3.14)’te tanımlı olan h_t değeri t zamanındaki çıktı değerlerini göstermektedir. Denklem (3.10) ve (3.11)’deki h_{t-1} ise $t - 1$ zamanındaki çıktı değerini göstermektedir. TanH aktivasyon fonksiyonu Denklem (3.15)’te gösterilmiştir.

3.10.3.3. GRU (geçitli tekrarlayan birimler - gated recurrent unit)

GRU mimarisinde, LSTM yapısına göre kapıların sayısı azaltılmıştır. Böylece yapıda meydana gelen karmaşıklığı azaltarak, hesaplama performansının artırılması sağlanmıştır. GRU, bir LSTM ünitesindeki unutma ve giriş kapısını güncelleme kapısı olarak birleştirmektedir. Bunun sonucunda gizli durum ve hücre durumunun birleştirilmesi ile LSTM modelinin daha hafif bir versiyonu elde edilmektedir (Zang ve Kabuka, 2018). Bu elde edilen versiyon Şekil 3.19'da gösterilmiştir.



Şekil 3.19. GRU blok diyagramı

GRU mimarisinin kullanılan matematik ifadeleri Denklem (3.16) ile Denklem (3.19) arasında gösterilmiştir.

$$h_t = (1 - z_t)h'_t + z_th_{t-1} \quad (3.16)$$

$$z_t = \sigma(W_z h_{t-1} + W_z x_t + b_z) \quad (3.17)$$

$$h'_t = \text{TanH}(W_{hh} r_t h_{t-1} + W_{hx} x_t + b_h) \quad (3.18)$$

$$r_t = \sigma(W_{hr} h_{t-1} + W_{xr} x_t + b_r) \quad (3.19)$$

Verilen Eşitliklerde W ağırlık matrisleri, b bias değerleri, TanH ve σ aktivasyon fonksiyonu için kullanılmıştır. h_t gizli durum yapısı, r_t sıfırlama kapısı, z_t güncelleme kapısı ve h'_t aday gizli duruma karşılık gelmektedir. Sıfırlama kapısı veri setindeki kısa

sürekli bağımlılıkların tutulmasını sağlarken güncelleme kapısı uzun süreli bağımlılıkların tutulmasını sağlar.

3.10.3.4. Optimizasyon ve performans ölçüm yöntemleri

RNN, LSTM ve GRU tabanlı derin öğrenme modellerinin eğitiminde çok fazla zamana ve belleğe ihtiyaç duyulmasından dolayı Adam optimizasyon yöntemi kullanılmıştır. Adam optimizasyon algoritması önerilen modellerin daha hızlı yakınsamasını sağlayarak daha az kaynak tüketimi sağlamaktadır (Yaguchi vd., 2018). Bunun sonucunda da öğrenme hızı arttırılmaktadır. Adam algoritması birinci dereceden bir optimizasyon algoritmasıdır. Momentumun birinci mertebeden moment tahmini hesaplamasının temelinde ikinci moment tahminine Adadelata momentinin eklenmesidir. Adam optimizasyon yönteminde her parametrenin öğrenme oranı birinci ve ikinci moment tahmini kullanılarak yapılmaktadır. Bias düzeltmeleri ile de parametreler nispeten kararlı hale getirilmektedir. Denklem (3.20) ile Denklem (3.25) arasında Adam optimizasyon yöntemine ait matematik ifadeleri verilmiştir.

$$g = \left(h_{\theta}(x^i - y^i) \right) x^i \quad (3.20)$$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)g \quad (3.21)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)g^2 \quad (3.22)$$

$$m'_t = \frac{m_t}{1 - \beta_1^t} \quad (3.23)$$

$$v'_t = \frac{v_t}{1 - \beta_2^t} \quad (3.24)$$

$$\theta_j = \theta_{j-1} - m'_t = \frac{\alpha}{\sqrt{v'_t} + \epsilon} \quad (3.25)$$

Denklem (3.21)'deki g değeri hesaplanmış gradyanı göstermektedir. Denklem (3.22)'deki m_t ise g değerinin ilk momentidir. Denklem (3.23)'deki v_t ise g değerinin ikinci momentidir. β_1 birinci dereceden moment düzeltme katsayısıdır. β_2 ise ikinci dereceden moment düzeltme katsayısıdır. θ güncellenmesi gereken parametreleri temsil etmektedir. m'_t ve v'_t ise ofset düzeltme katsayılarıdır (Jiang vd., 2020).

Tezde tahminler Ortalama Kare Hata (Mean Squared Error - MSE) metriği ile ölçülmüştür. MSE derin öğrenme modellerinin tahmini ile gerçek değerler arasındaki

farkın karelerinin toplamının tahmin saysına bölünmesi ile elde edilir. MSE'nin matematik ifadesi Denklem (3.27)'de gösterilmiştir.

$$MSE = \frac{1}{m} \sum_{i=1}^m (Y_i - \hat{Y}_i)^2 \quad (3.26)$$



4. BULGULAR VE TARTIŞMA

Tezde yapılan verilerin ön işlenmesi ve derin öğrenme modellerinin geliştirilmesinde Python 3.8 kullanılmıştır. RNN, LSTM ve GRU yineleyen derin öğrenme yöntemleri Tensorflow 2.7.0 ile geliştirilmiştir. Hiperparametre optimizasyonu için KerasTuner 1.1.0 kütüphanesi kullanılmıştır (Pang vd., 2020; Hertel vd., 2020).

Veri setinde her bir veri 300 ms'deki hız ve yakıtı göstermek üzere toplam 8553 veri bulunmaktadır. Veri setinde pencere boyutu 5, 10 ve 15 olarak ayarlandığında %60 eğitim, %40 test için ayrıldığında kullanılan veri sayıları Çizelge 4.1'de gösterilmiştir. Dolayısıyla RNN, LSTM ve GRU modellerinde 3 pencere boyutu kullanılmıştır.

Çizelge 4.1. Veri setinin pencere boyutu, eğitim ve test veri sayıları

Pencere Boyutu	Eğitim Veri Sayısı	Test Veri Sayısı
5	5 111	3 430
10	5 099	3 427
15	5 094	3 417

RNN, LSTM, GRU yineleyen derin öğrenme modelleri için hiperparametre optimizasyonu yapılmıştır. Yineleyen modeller 1-3 grup halinde kullanılmıştır. Her bir grupta 8-256 arasında ünite bulunmaktadır. Yineleyen modellerden sonra 1-3 arasında tam bağlantı katmanı mevcuttur. Düğüm ve ünitelerde Sigmoid, TanH, ReLU aktivasyon fonksiyonları kullanılmıştır. Eğitim esnasında aşırı öğrenmenin önüne geçmek için %0-%40 oranında bırakma yapılmıştır. Öğrenme oranları 10^{-1} ile 10^{-4} . Hiperparametre optimizasyonunda kullanılan parametreler Şekil 4.1'de gösterilmiştir.

Hiperparametre Optimizasyonu						
Model	RNN, LSTM, GRU					
Pencere	5	10	15			
Tekrarlayan Katman Sayısı	1	2	3			
Tam Bağlantı Katmanı	1	2	3			
Ünite - Düğüm Sayısı	8	16	32	64	128	256
Aktivasyon Fonksiyonu	ReLU	TanH	Sigmoid			
Bırakma Yüzdeleri	0	10	20	30	40	
Öğrenme Oranı	10^{-1}	10^{-2}	10^{-3}	10^{-4}		

Şekil 4.1. Hiperparametre optimizasyonu

Üç pencere boyutunda her bir model rastgele arama ile her birisinde 2 yürütme olmak üzere 300 denemeye tabi tutulmuştur. Dolayısıyla toplam 2 700 farklı modelde 5 400 yürütme gerçekleştirilmiştir. Kullanılan hiperparametrelerele elde edilen sistemin eğitim performansı 5 adımda iyileşmeyi bıraktığında sistemin eğitimi durdurulmuştur.

4.1. Önerilen Modellerin Performans Sonuçları

Verilerin eğitilmesi ve test işleminde kullanılan yineleyen derin öğrenme modellerinin temel yapısı Şekil 4.2’de gösterilmiştir. Bu şekilde, yineleyen derin öğrenme katmanında RNN, LSTM veya GRU ünitelerden bir tanesi bulunmaktadır.



Hiperparametre optimizasyonunda tahmin doğrulukları en yüksek RNN, LSTM, GRU modelleri için elde edilen eğitim ve doğrulama kayıpları Çizelge 4.2’de verilmiştir.

Çizelge 4.2. Derin öğrenme modelleri performans sonuçları

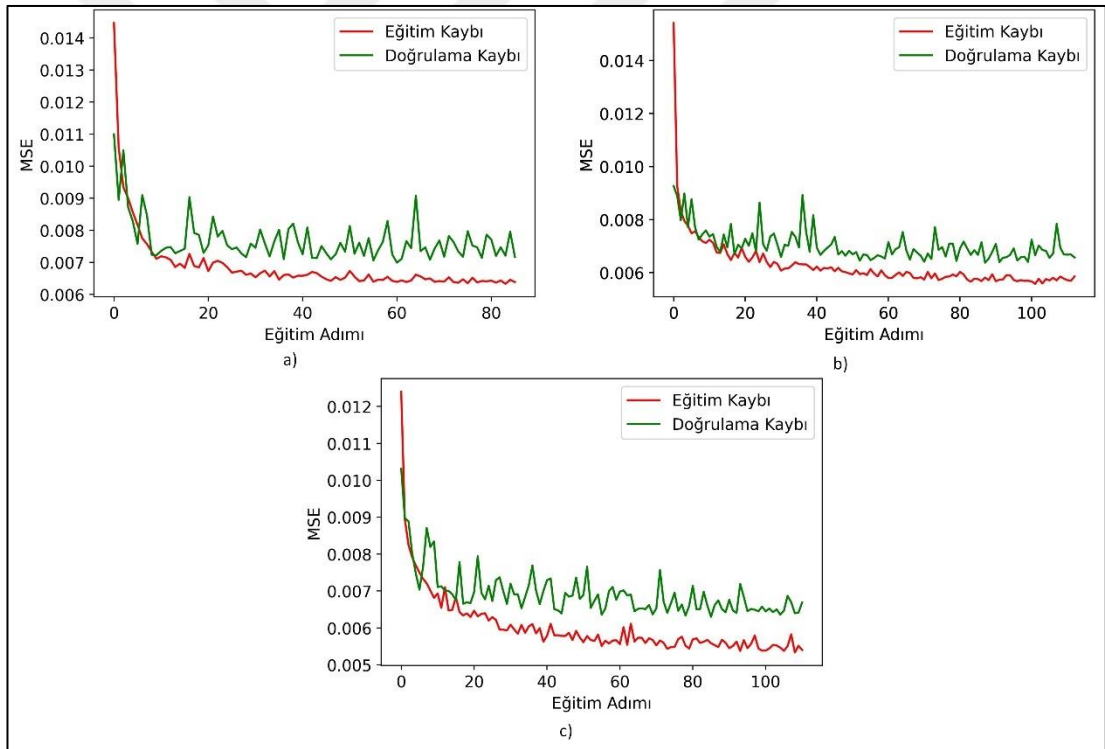
Yöntem	Pencere Sayısı	Eğitim Kayıpları	Doğrulama Kayıpları
RNN	5	0.0064	0.0071
RNN	10	0.0057	0.0064
RNN	15	0.0056	0.0063
LSTM	5	0.0070	0.0075
LSTM	10	0.0127	0.0110
LSTM	15	0.0058	0.0063
GRU	5	0.0063	0.0070
GRU	10	0.0077	0.0072
GRU	15	0.0060	0.0064

Çizelge 4.2’de eğitim ve doğrulama kaybını en aza indiren modellerin pencere boyutu 15 olan RNN, LSTM ve GRU modelleri olduğu görülmektedir. En performanslı çalışan model LSTM-15’de eğitim kaybının 0.0058’e ve doğrulama kaybının 0.0063’e kadar düştüğü görülmüştür. Bununla birlikte farklı denemelerde RNN, LSTM ve GRU modellerinin kayıpları birbirine çok yakın olmakla birlikte en yüksek performansın diğer pencere ve modellerde elde edildiği de gözlenmiştir.

Çizelge 4.3. Modelde kullanılan hiperparametreler ve parametre sayıları

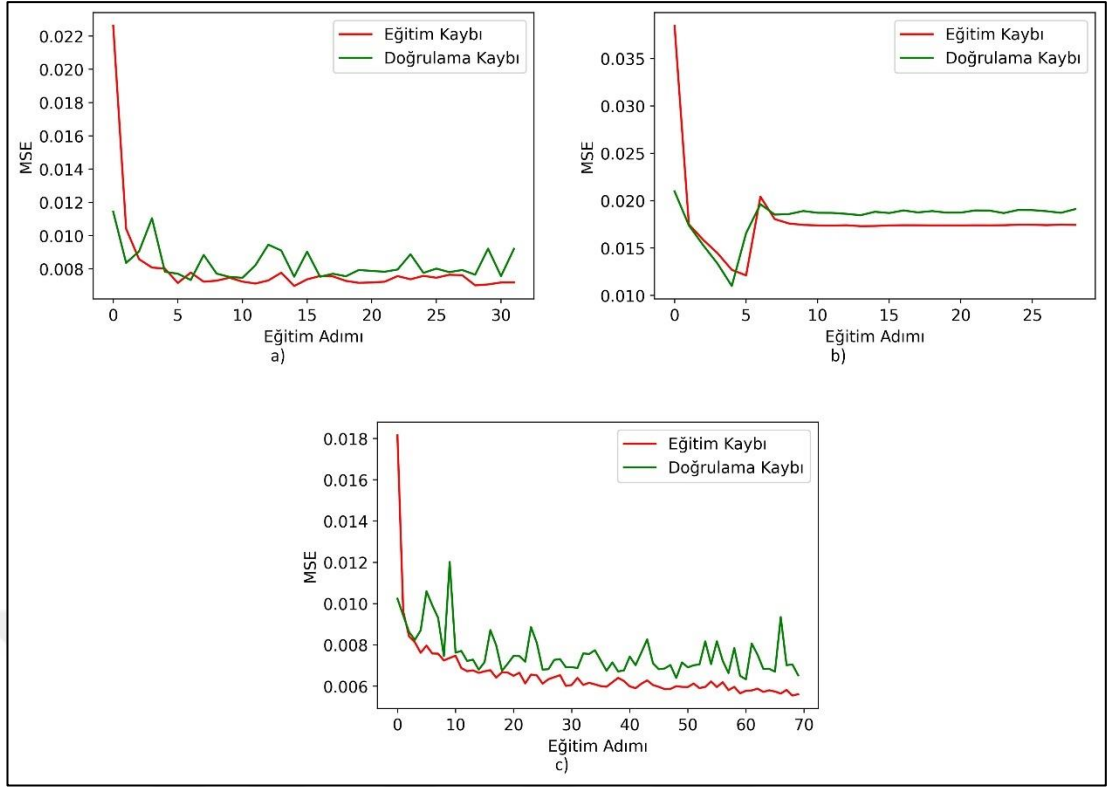
Model		Aktivasyon	Ünite-Düğüm Sayısı	Bırakma Yüzdeleri	Eğitilen Parametre Sayısı	Toplam Eğitilen Parametre Sayısı
RNN 5	RNN Katmanı	TanH	128	30	16 640	49 921
	Tam Bağlantı Katmanı	ReLU	256	10	33 024	
	Regresyon Katmanı		1		257	
RNN 10	RNN Katmanı	TanH	128	20	16 640	33 281
	Tam Bağlantı Katmanı	ReLU	128	10	16 512	
	Regresyon Katmanı		1		129	
RNN 15	RNN Katmanı	TanH	64	20	4 224	12 673
	Tam Bağlantı Katmanı	ReLU	128	10	8 320	
	Regresyon Katmanı		1		129	
LSTM 5	LSTM Katmanı	TanH	32	10	1 088	5 441
	Tam Bağlantı Katmanı	ReLU	128	20	4 224	
	Regresyon Katmanı		1		129	
LSTM 10	1. LSTM Katmanı	TanH	32	40	1 088	8 113
	2. LSTM Katmanı	TanH	16	0	784	
	3. LSTM Katmanı	TanH	64	0	5 184	
	Tam Bağlantı Katmanı	Sigmoid	16	10	1 040	
	Regresyon Katmanı		1		17	
LSTM 15	LSTM Katmanı	TanH	256	10	6 6048	99 073
	Tam Bağlantı Katmanı	ReLU	128	10	32 896	
	Regresyon Katmanı		1		129	
GRU 5	GRU Katmanı	ReLU	256	20	6 6048	99 073
	Tam Bağlantı Katmanı	ReLU	128	20	32 896	
	Regresyon Katmanı		1		129	
GRU 10	GRU Katmanı	TanH	64	30	4 224	21 121
	Tam Bağlantı Katmanı	ReLU	256	20	16 640	
	Regresyon Katmanı		1		257	
GRU 15	1. GRU Katmanı	TanH	128	20	16 640	82 817
	1. GRU Katmanı	TanH	128	20	32 896	
	Tam Bağlantı Katmanı	TanH	256	10	33 024	
	Regresyon Katmanı		1		257	

Çizelge 4.3'te hiperparametre optimizasyonu sonucu elde edile derin öğrenme modelleri, aktivasyon fonksiyonları, ünite-düğüm sayıları ve eğitilen parametre sayıları verilmiştir. Hiperparametre optimizasyonu sonucu en performanslı çalışan modellerin LSTM-10 ve GRU-15 dışında tek katmalı modeller olduğu Çizelge 4.3'te görülmektedir. LSTM-10'da üç tane LSTM katmanı ve GRU-15'de iki tane GRU katmanı kullanıldığı görülmüştür. Kullanılan aktivasyon fonksiyonları TanH, ReLU ve Sigmoid olmasına rağmen TanH ve ReLU sigmoid'e göre daha performanslı sonuç vermektedir. Eğitilen parametre sayısı en fazla olan modeller 99 073 ile LSTM-15 ve GRU-15 modelleridir. En az olan model ise 5 441 parametre ile LSTM-5 modelidir. Bütün modellerde 1 tane tam bağlantı katmanı kullanılmıştır ve 1 regresyon katmanı bulunmaktadır. Ayrıca hiperparametre optimizasyonunda kullanılan bırakma katmanının %0 ile %40 arasındaki oranları yine Çizelge 4.3'te görülmektedir.



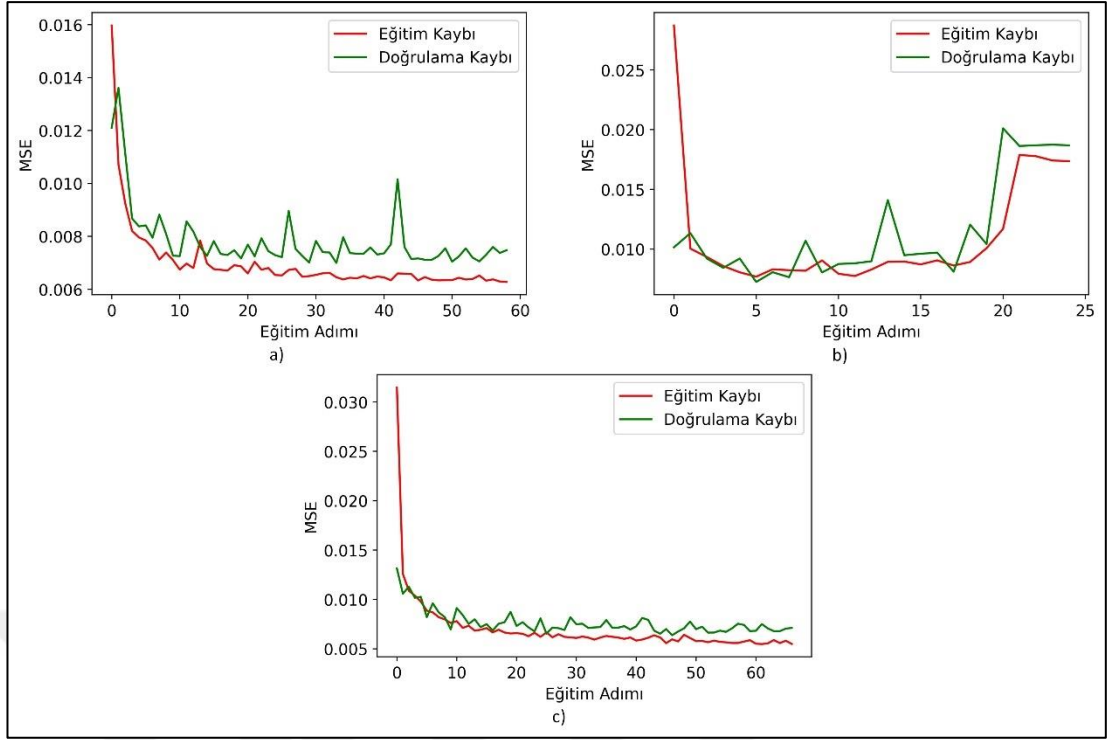
Şekil 4.3. a)5, b)10, c)15 pencere boyutlu RNN eğitim ve doğrulama kayıpları

Şekil 4.3'de RNN modellerin eğitim ve doğrulama kayıpları verilmiştir. Şekil 4.3 a'da pencere boyutu 5 olan RNN, b'de pencere boyutu 10 olan RNN ve c'de pencere boyutu 15 olan RNN modelinin eğitim ve doğrulama kayıpları gösterilmiştir. RNN-5 86. adımda, RNN-10 113. adımda ve RNN-15 111. adımda eğitim işlemini sonlandırmıştır.



Şekil 4.4. a)5, b)10, c)15 pencere boyutlu LSTM eğitim ve doğrulama kayıpları

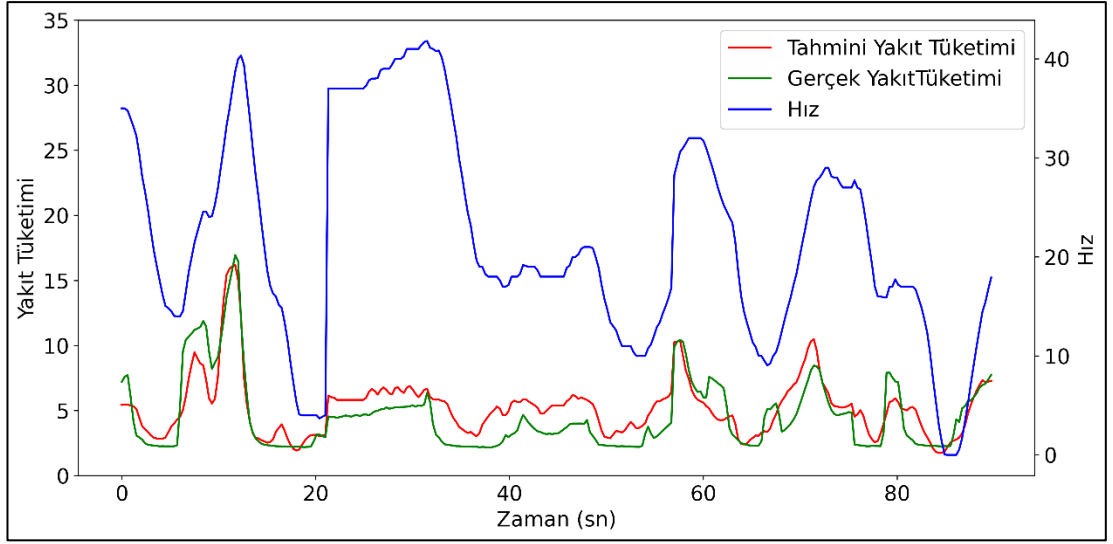
Şekil 4.4’de LSTM modellerin eğitim ve doğrulama kayıpları verilmiştir. Şekil 4.4 a’da pencere boyutu 5 olan LSTM, b’de pencere boyutu 10 olan LSTM ve c’de pencere boyutu 15 olan LSTM modelinin eğitim ve doğrulama kayıpları gösterilmiştir. LSTM-5 32. adımda, LSTM-10 30. adımda ve LSTM-15 86. adımda eğitim işlemini sonlandırmıştır.



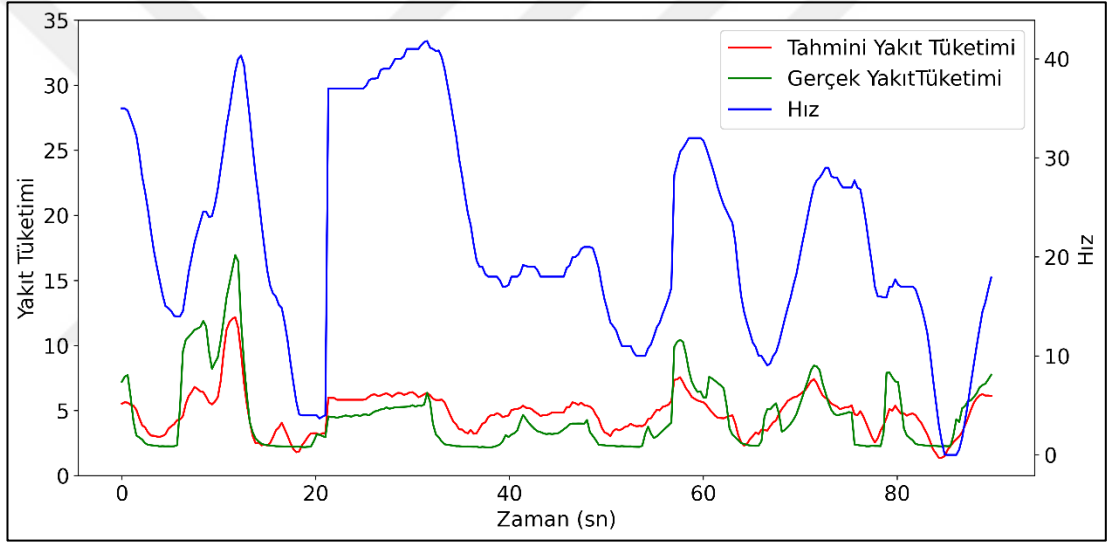
Şekil 4.5. a)5, b)10, c)15 pencere boyutlu GRU eğitim ve doğrulama kayıpları

Şekil 4.5’de GRU modellerin eğitim ve doğrulama kayıpları verilmiştir. Şekil 4.5a’da pencere boyutu 5 olan GRU, b’de pencere boyutu 10 olan GRU ve c’de pencere boyutu 15 olan GRU modelinin eğitim ve doğrulama kayıpları gösterilmiştir. GRU-5 67. adımda, GRU-10 59. adımda ve GRU-15 67. adımda eğitim işlemini sonlandırmıştır.

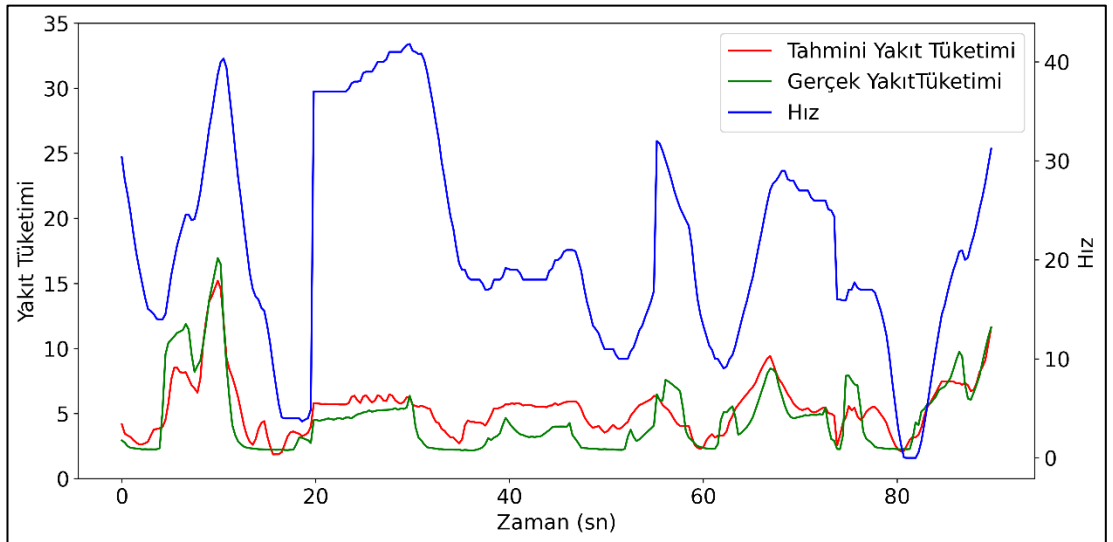
Şekil 4.6, Şekil 4.7 ve Şekil 4.8’de en yüksek tahmin performansı olan modellerin yakıt tüketimi tahminleri gösterilmiştir. Sol y eksenini yakıt tüketimini, sağ y eksenini ise hızı göstermektedir. Mavi çizgi ölçülen hızı, yeşil çizgi ölçülen yakıt tüketimini, kırmızı çizgi ise modellerin tahmini sonucu elde edilen yakıt tüketimini göstermektedir.



Şekil 4.6. RNN-5 tahmin performansı



Şekil 4.7. LSTM-5 tahmin performansı



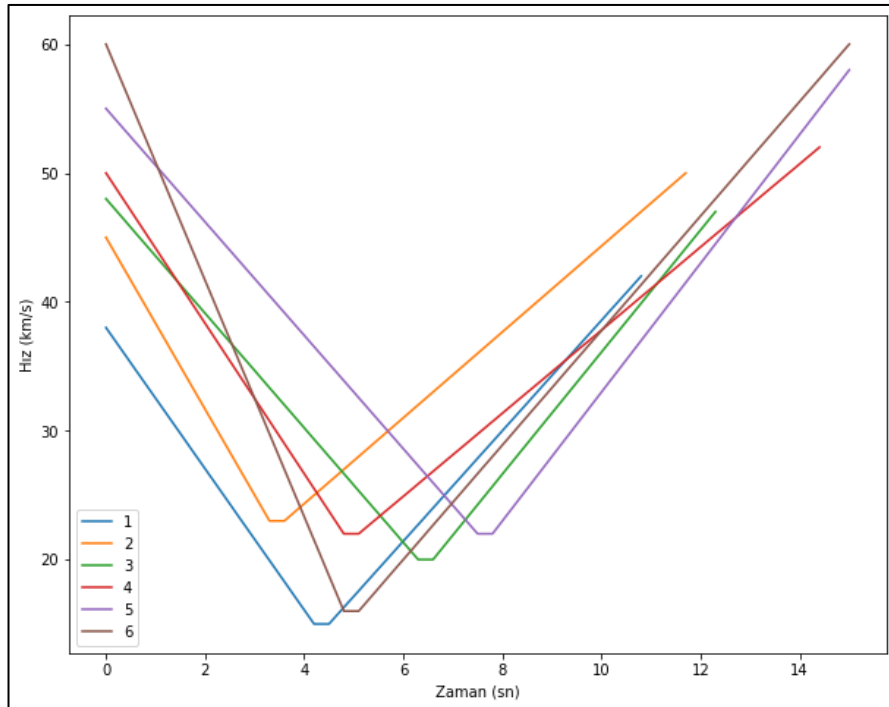
Şekil 4.8. GRU-15 tahmin performansı

Çizelge 4.4’de gerçek sürüş esnasında kasislerden geçişlerdeki hız ve zaman bilgileri gösterilmektedir. Bu çizelgede “Giriş Hızı” kasis için yavaşlamaya başlamadan önceki hızı, “En Düşük Hız” kasisten kaynaklı en düşük hızı, “Yavaşlama Süresi” Giriş Hızı ile En Düşük Hız arasında geçen süreyi, “Çıkış Hızı” kasisin etkisinin geçtiği ve kasisten önceki benzer sabit hıza geçildiği hızı, “Hızlanma Süresi” En Düşük Hızla Çıkış Hızı arasındaki geçen süreyi ifade etmektedir.

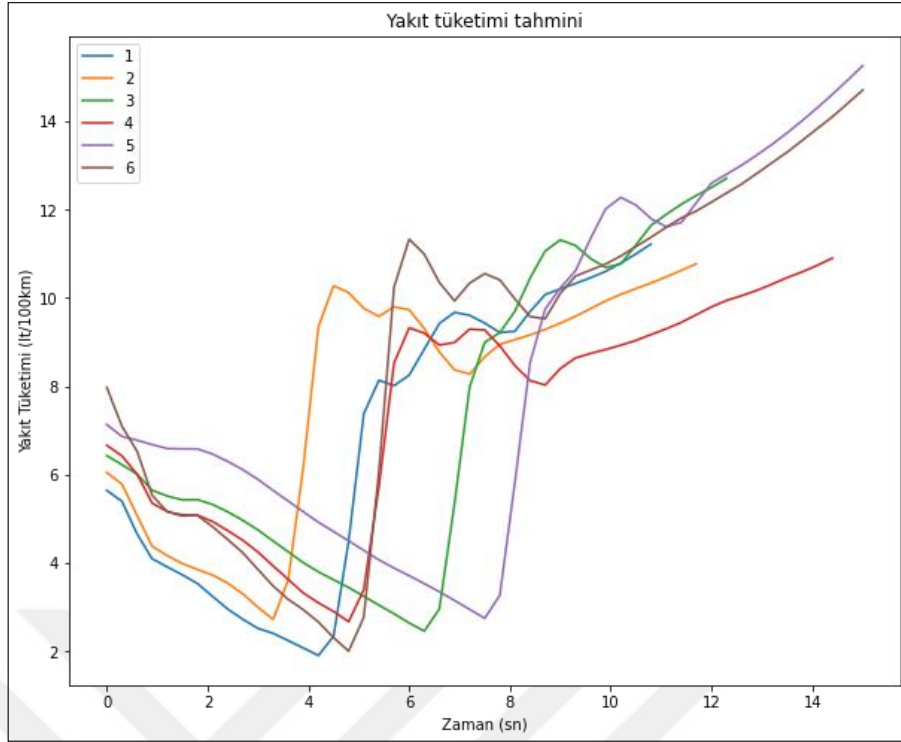
Çizelge 4.4. Gerçek zamanlı kasis geçiş bilgileri

Giriş Hızı (km/sa)	Yavaşlama Süresi (sn)	En Düşük Hız (km/sa)	Hızlanma Süresi (sn)	Çıkış Hızı (km/sa)
38	4.4	15	6.6	42
45	3.7	23	8.5	50
48	6.7	20	5.9	47
50	5.2	22	9.6	52
55	7.7	22	7.6	58
60	5.2	16	10.3	60

Çizelge 4.4’deki veriler kullanılarak Hızın belirtilen zamanlardaki doğrusal olarak azalma ve artmaları Şekil 4.9’da gösterilmiştir. Burada y eksenini kasisten geçiş esnasında oluşan hızı, x eksenini ise kasisten geçme süresini göstermektedir.



Şekil 4.9. Kasisten geçiş esnasında doğrusal yavaşlama ve hızlanma



Şekil 4.10. Kasisten geçiş sırasındaki yakıt tüketim tahmini

Şekil 4.10’da bulunan grafik Şekil 4.9’da hesaplanan kasis yaşvaşlama ve hızlanmalar için tahmin edilen yakıt tüketimlerini göstermektedir. Yakıt tüketim tahmini hiperparametre optimizasyonu sonucu en performanslı çalışan model olan RNN-15 kullanılarak yapılmıştır.

Çizelge 4.5’de kasisten geçiş sırasında harcanan yakıt miktarı (lt), kasisin etkisi boyunca gidilen mesafe verilmektedir. Ayrıca aynı mesafe kasis olmadan sabit “Giriş Hızında” gidilseydi harcanan yakıt miktarı (lt) ve kasisli yolda sabit hıza göre fazladan harcanan yakıt miktarı yüzdesel olarak verilmiştir.

Çizelge 4.5. Kasisten kaynaklı yakıt artış oranları

Kasis Tahmin Yakıt (lt/100km)	Sabit Hız Yakıt (lt/100km)	Fazladan Harcanan Yakıt Yüzdesi	Gidilen Mesafe (m)
0.0062	0.0048	28.38	85.38
0.0095	0.0073	31.03	119.17
0.0090	0.0077	16.30	118.17
0.0116	0.0100	16.37	149.67
0.0150	0.0118	27.11	166.75
0.0154	0.0128	20.98	161.50

Çizelge 4.5’de görüldüğü gibi aynı mesafedeki yolda kasis olmadığı durumda tasarruf edilecek yakıt miktarı %16.3 ile %31.03 arasında değişmektedir. Kasisteki hızlanma süresi ve kasis çıkış hızının yüksek olması kasisten kaynaklı yakıt tüketiminin artmasına sebep olduğu görülmektedir.



5. SONUÇ VE ÖNERİLER

Tez çalışmasında trafik akışını yavaşlatmak için çok yaygın olarak kullanılan hız kesici tümseklerin (kasis) yakıt tüketimine etkisi araştırılmıştır. Bunun için aracın CAN veri yoluna OBD-II portu ile ulaşarak, aracın hız ve yakıt tüketimi ile ilgili veriler elde edilerek veri seti oluşturulmuştur. Veri setinde ön işleme ile eksik veriler interpolasyon yöntemi ile doldurulmuş, bir zaman aralığındaki verilerin ise ortalaması bulunarak o zaman birimi için değer tespit edilmiştir. Kullanılan derin öğrenme yöntemlerinde veriler %60-%40 eğitim ve doğrulama seti olarak bölünmüştür. %60 eğitim setindeki en büyük ve en küçük değerler göz önüne alınarak normalizasyon yapılmıştır. Elde edilen normalizasyon parametreleri ile doğrulama seti de normalize edilmiştir.

Ardışık verilerin kullanımının yakıt tüketimini hesaplamada daha doğru sonuç vermesinden dolayı Tekrarlayan Sinir Ağları (RNN), Uzun Kısa Süreli Bellek (LSTM), Geçitli Tekrarlayan Birim (GRU) yineleyen derin öğrenme modelleri tasarlanmıştır. Araçtan gerçek zamanlı elde edilen veriler farklı katman sayısı ve hiperparametre optimizasyonu ile en yüksek doğruluk değerini veren model belirlenmiştir. Ayrıca en yüksek tahmin performansı sağlayan öğrenme katsayısı, TanH, ReLU ve sigmoid aktivasyon fonksiyonları da hiperparametre optimizasyonunda araştırılmıştır.

Eğitilen modellerde en düşük binde 63 ortalama hata karesi elde edilmiştir. En az tahmin hatasını veren model olan 15 ardışıl verili RNN modeli ile yapılan yakıt tahmini ve hesaplamalar neticesinde bir kasisin %16.30 ile %31.03 arasında daha fazla yakıt sarfiyatına sebep olduğu tespit edilmiştir. Bu tespit her ne kadar kullanılan araç için yapılmış olsa da genel olarak araçlardan elde edilecek hız-yakıt verileri ile benzer tahminlerin yapılabilmesi mümkündür. Aynı zamanda yolun eğimi, aracın taşıdığı ağırlık, motor hacmi, yakıt türü gibi pek çok parametreler kasislerdeki yakıt sarfiyatını etkilemesi kaçınılmazdır. Bununla birlikte kasislerin ortalama bir araçtaki sebep olduğu fazladan yakıt sarfiyatı yapılan çalışma ile anlaşılabilir.

Şehirlerde hızı yavaşlatmak için çok fazla sayıda bulunan kasislerin üzerinde geçen araçların sayısı da göz önüne alındığında ciddi miktarda yakıt tüketimine sebep olabileceği anlaşılmaktadır. Ayrıca kasislerin standartlara uygun olarak yapılmamış olması veya kasisin sebep olacağı asgari hızın belirli olmaması da ya fazladan yakıt sarfiyatına sebep olmaktadır. Kasislerden geçiş esnasında aracın zarar görmesi veya sürüş konforunun azalmasına sebep olduğu da aşıkardır. Dolayısıyla sadece yoğun saatlerde kullanımının zaruri olduğu kasislerin mevcut durumda günün her saatinde etki etmesi söz konusudur. Bu durumun önüne geçilmesi için hızların elektronik olarak tespit edilmesi gibi daha modern yöntemlerin kullanılmasının daha uygun olduğu düşünülmektedir.



KAYNAKLAR

- Aggarwal, C. C. (2018). Neural networks and deep learning. *Springer*. (pp. 973–978)
- Amarasinghe, M., Kottegoda, S., Arachchi, A. L., Muramudalige, S., Bandara, H. M. N. D. & Azeez, A. (2015). Cloud-based driver monitoring and vehicle diagnostic with OBD-II telematics. *2015 Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTer)*, 243–249.
- Anonim (2021a). *Türkiye'de Petrol Üretimi*. <https://www.petform.org.tr/arama-uretim-sektoru/turkiyede-petrol-uretimi/> (Son erişim tarihi: 30.07.2021)
- Anonim (2021b). *Ford Fiesta Collection Özellikleri*. <https://oto.net/ford/fiesta/fiesta-collection-1.4-tdci-5-kapi/bilgi/> (Son erişim tarihi: 30.08.2021)
- Anonim (2021c). *On-board Diagnostics*. https://en.wikipedia.org/wiki/On-board_diagnostics#OBD-II_diagnostic_connector (Son erişim tarihi: 02.08.2021)
- Arora, R., Basu, A., Mianjy, P. & Mukherjee, A. (2016). *Sixth International Conference on Learning Representations*. Understanding deep neural networks with rectified linear units, April 30-May 3, Vancouver, 1-17.
- Baek, S. & Jang, J. (2015). Implementation of integrated OBD-II connector with external network. *Information Systems*, 50, 69-75. doi:10.1016/j.is.2014.06.011.
- Baek, S. H., Kim, H.S., Jeong, D. W., Kim, M. J., Park, Y. S. & Jang, J. W. (2011). Implementation vehicle driving state system with OBD-II, MOST network. *7th Asia-Pacific Conference on Communications (APCC)*, 709–714.
- Banzi, M. & Shiloh, M. (2011). *Getting started with Arduino*. Maker Media, Inc.
- Bozdal, M., Samie, M., Aslam, S. & Jennions, I. (2020). *Evaluation of CAN Bus Security Challenges, and Potential Solutions*. 16–17 August, International Conference on Computing, Electronics & Communications Engineering (iCCECE), Sensors, 2364 - 2383.
- Cetiner, I., Var, A. A. & Cetiner, H. (2016). Classification of knot defect types using wavelets and KNN. *Elektronika Ir Elektrotechnika*, 22(6), 67–72. doi:10.5755/j01.eie.22.6.17227
- Chen, S. H., Pan, J. S. & Lu, K. (2015). *Driving behavior analysis based on vehicle OBD information and adaboost algorithms*. In Proceedings of the international multiconference of engineers and computer scientists. March 18 – 20, Hong Kong, 18-20.

- Choi, S., Kim, J., Kwak, D., Angkititrakul, P. & Hansen, J. (2007). Analysis and Classification of Driver Behavior Using in-Vehicle Can-Bus Information. 3. Biennial Workshop on DSP for Mobile and Vehicular Systems. https://www.researchgate.net/publication/228936619_Analysis_and_Classification_of_Driver_Behavior_Using_in-Vehicle_CAN-BUS_Information (Son erişim tarihi: 01.07.2022)
- Chollet, F. (2018). *Deep learning with Python*. New York, Manning.
- Coşkun, U. (2008). *Controller area network ve uygulaması*. (Yüksek Lisans Tezi, Gebze Teknik Üniversitesi, Yüksek Teknoloji Enstitüsü)
- Cueva-Fernandez, G., Espada, J. P., García-Díaz, V., García, C. G. & Garcia-Fernandez, N. (2014). Vitruvius: An expert system for vehicle sensor tracking and managing application generation. *Journal of Network and Computer Applications*, 42, 178–188. doi:<https://doi.org/10.1016/j.jnca.2014.02.013>
- Çetiner, H. (2021a). Classification of apple leaf diseases using the proposed convolution neural network approach. *Journal of Engineering Sciences and Design*, 9(4), 1130–1140. doi:[10.21923/jesd.980629](https://doi.org/10.21923/jesd.980629)
- Çetiner, H. (2021b). *Python Ortamında Derin Öğrenme Uygulamaları*. Anı Yayıncılık.
- Derse, C. (2017). *Yeni Bir Can Bus İletişim Tabanlı Otomotiv Güç Dağıtım Modülü Uygulaması ve Sistemin Güvenilirlik Analizi: “eaPDM”*. (Yüksek Lisans Tezi, Celal Bayar Üniversitesi Fen Bilimleri Enstitüsü)
- Gal, Y. & Ghahramani, Z. (2016). *Dropout as a bayesian approximation: Representing model uncertainty in deep learning*. In international conference on machine learning. June-06, PMLR, 1050-1059.
- Güçlü, Ö. (2018). *Hata Simülasyonlu Can-Bus Eğitim Seti Geliştirilmesi*. (Yüksek Lisans Tezi, Kocaeli Üniversitesi Fen Bilimleri Enstitüsü)
- Hertel, L., Collado, J., Sadowski, P., Ott, J. & Baldi, P. (2020). Sherpa: Robust hyperparameter optimization for machine learning. *SoftwareX*, 12, 100591. doi:[10.1016/j.softx.2020.100591](https://doi.org/10.1016/j.softx.2020.100591)
- Huang, M. H., Rust, R. & Maksimovic, V. (2019). The feeling economy: Managing in the next generation of artificial intelligence (AI). *California Management Review*, 61(4), 43–65. doi:[0.1177/0008125619863436](https://doi.org/10.1177/0008125619863436)
- Jiang, F., Jiang, Y., Zhi, H., Dong, Y., Li, H., Ma, S., Wang, Y., Dong, Q., Shen, H. & Wang, Y. (2017). Artificial intelligence in healthcare: Past, present and future. *Stroke and Vascular Neurology*, 2(4). 230-243. doi:[10.1136/svn-2017-000101](https://doi.org/10.1136/svn-2017-000101)

- Jiang, X., Hu, B., Chandra S. S., Wang, S. H. & Zhang, Y. D. (2020). Fingerspelling identification for Chinese sign language via AlexNet-based transfer learning and Adam optimizer. *Scientific Programming*, 2020,1-13. doi:10.1155/2020/3291426
- Karanfil, G. (2020). *OBD-II Protokollerini Kullanan Bir Arayüz Yazılımı Tasarımı*. (Yüksek Lisans Tezi, Bursa Uludağ Üniversitesi Fen Bilimleri Enstitüsü)
- Khanapuri, A. V., Shastri, A., D'souza, G. & D'souza, S. (2015). *On road: A car assistant application*. International Conference on Technologies for Sustainable Development (ICTSD). February 04 – 06, Mumbai - India 1–7.
- Ko, K. H. (2010). The change rate of fuel consumption for different IRI of paved roads. *International journal of highway engineering*, 12(1), 55-59.
- Kosovic, B., Haupt, S. E., Adriaansen, D., Alessandrini, S., Wiener, G., Delle-Monache, L., Liu, Y., Linden, S., Jensen, T. & Cheng, W. (2020). A comprehensive wind power forecasting system integrating artificial intelligence and numerical weather prediction. *Energies*, 13(6), 1372-1386. doi:10.3390/en13061372
- Kowalik, B. & Szpyrka, M. (2019). Online environment for data acquisition from car sensors. *Automatyka/Automatics*, 23(1), 7-7. doi:10.7494/automat.2019.23.1.7.
- Kowalik, B. (2018). *Introduction to car failure detection system based on diagnostic interface*. In 2018 International Interdisciplinary PhD Workshop. June 21-18, IIPhDW, 4-7.
- Kumar, N., Kharkwal, N., Kohli, R. & Choudhary, S. (2016). Ethical aspects and future of artificial intelligence. *International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH)*, 111–114. doi:10.1109/ICICCS.2016.7542339
- Lalonde, M., Beaulieu, M. & Gagnon, L. (2001). Fast and robust optic disc detection using pyramidal decomposition and hausdorff-based template matching. *Medical Imaging, IEEE*, 20(11), 1193–1200. doi:10.1109/42.963823
- Maas, M. M. (2019). How viable is international arms control for military artificial intelligence? Three lessons from nuclear weapons. *Contemporary Security Policy*, 40(3), 285–311. doi:10.1080/13523260.2019.1576464
- Macrae, C. (2019). Governing the safety of artificial intelligence in healthcare. *BMJ Quality and Safety*, 28(6), 495–498. doi:10.1136/bmjqs-2019-009484
- Mao, J., Xu, W., Yang, Y., Wang, J., Huang, Z. & Yuille, A. (2014). Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv preprint arXiv:1412.6632*. doi:/10.48550/arXiv.1412.6632

- Martinelli, F., Mercaldo, F., Orlando, A., Nardone, V., Santone, A. & Sangaiah, A.K. (2020). Human behavior characterization for driving style recognition in vehicle system. *Computers and Electrical Engineering*, 83, 1-16. doi:10.1016/j.compeleceng.2017.12.050.
- McGovern, A., Elmore, K. L., Gagne, D. J., Haupt, S. E., Karstens, C. D., Lagerquist, R., Smith, T. & Williams, J. K. (2017). Using artificial intelligence to improve real-time decision-making for high-impact weather. *Bulletin of the American Meteorological Society*. (pp. 2073–2090)
- Meseguer, J. E., Calafate, C. T., Cano, J. C. & Manzoni, P. (2015). *Assessing the Impact of Driving Behavior on Instantaneous Fuel Consumption*. 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), January 09-12, Vegas, 443–448.
- Mi-JinKim, Jong-Wook, J. & Yun-Sik, Y. (2010). A Study on In-Vehicle Diagnosis System using OBD-II with Navigation. *IJCSNS International Journal of Computer Science and Network Security*, 10(9), 136-140.
- Ongsulee, P. (2017). *Artificial intelligence, machine learning and deep learning*. 2017 15th International Conference on ICT and Knowledge Engineering (ICT&KE), November 22-24, Bangkok, Thailand, 1–6.
- Pang, Y., Sun, M., Jiang, X. & Li, X. (2017). Convolution in convolution for network in network. *IEEE transactions on neural networks and learning systems*, 29(5), 1587-1597. doi:10.1109/tnnls.2017.2676130
- Perrotta, F., Parry, T. & Neves, L. C. (2017). Application of machine learning for fuel consumption modelling of trucks. *IEEE International Conference on Big Data (Big Data)*, December 07, 3810–3815.
- Qiu, J., Wu, Q., Ding, G., Xu, Y. & Feng, S. (2016). A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing*. (pp. 1–16)
- Quanqi, W., Jian, W. & Yanyan, W. (2011). *Design of vehicle bus data acquisition and fault diagnosis system*. In 2011 International Conference on Consumer Electronics, Communications and Networks (CECNet). April 16-18, Beijing, 245-248.
- Rimpas, D., Papadakis, A. & Samarakou, M. (2020). OBD-II sensor diagnostics for monitoring vehicle operation and consumption. *Energy Reports*, 6, 55–63. doi:10.1016/j.egyr.2019.10.018
- Rong, G., Mendez, A., Assi, E. B., Zhao, B. & Sawan, M. (2020). Artificial intelligence in healthcare: review and prediction case studies. *Engineering*, 6(3), 291–301. doi:10.1016/j.eng.2019.08.015

- Salouhou, A. (2019). *El yazısı karakter tanıma ve resim sınıflandırmada derin öğrenme yaklaşımları*. (Yüksek Lisans Tezi, Fatih Sultan Mehmet Vakıf Üniversitesi Lisansüstü Eğitim Enstitüsü)
- Sim, A. X. A. & Sitohang, B. (2014). *OBD-II Standard Car Engine Diagnostic Software Development*. In 2014 International Conference on Data and Software Engineering (ICODSE). November 26-28, Bandung, 1-5.
- Syahputra, R. (2016). Application of neuro-fuzzy method for prediction of vehicle fuel consumption. *Journal of Theoretical and Applied Information Technology*, 86(1), 138-150.
- Şen, B. (2020). *Estimating instant fuel consumption by machine learning and improving fuel consumption*. (Yüksek Lisans Tezi, Galatasaray Üniversitesi Fen Bilimleri Enstitüsü)
- Uyanık, T., Karatuğ, Ç. & Arslanoğlu, Y. (2020). Machine learning approach to ship fuel consumption: A case of container vessel. *Transportation Research Part D*, 84 (102389), 1-14. doi:10.1016/j.trd.2020.102389
- Vilgenoğlu, E. (2019). *Real-time vehicle monitoring and on-board diagnostic system*. (Yüksek Lisans Tezi, Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü)
- Wang, H., Zhang, X. & Ouyang, M. (2015). Energy consumption of electric vehicles based on real-world driving patterns: A case study of Beijing. *Applied Energy*, 157, 710–719. doi:10.1016/j.apenergy.2015.05.057
- Wang, J., Du, Y., & Wang, J. (2020). LSTM based long-term energy consumption prediction with periodicity. *Energy*, 197, 1-12. doi:10.1016/j.energy.2020.117197
- Wang, W., Liu, H., Lin, W., Chen, Y. & Yang, J. A. (2020). Investigation on Works and Military Applications of Artificial Intelligence. *IEEE Access*, 8, 131614-131625.
- Wickramanayake, S. & Bandara, D. H. M. N. (2016). *Fuel Consumption Prediction of Fleet Vehicles Using Machine Learning: A Comparative Study*. 2nd International Moratuwa Engineering Research Conference (MERCon), April 5-6, Moratuwa, 90–95.
- Yaguchi, A., Suzuki, T., Asano, W., Nitta, S., Sakata, Y. & Tanizawa, A. (2018). *Adam Induces Implicit Weight Sparsity in Rectifier Neural Networks*. 17th IEEE International Conference on Machine Learning and Applications (ICMLA), December 17-20, Florida, 318–325.
- Yıldız, U., Sezik, M., Cetiner, H. & Aydoğan, T. (2015). Evaluation of fetal lung ultrasound images by digital texture analysis methods. *Journal of Clinical and Analytical Medicine*, 6, 479-484.

- Zaldivar, J., Calafate, C. T., Cano, J. C. & Manzoni, P. (2011). *Providing Accident Detection in Vehicular Networks through OBD-II Devices and Android-Based Smartphones*. IEEE 36th Conference on Local Computer Networks. October 4–7, Bonn, 813–819.
- Zhang, D. & Kabuka, M. R. (2018). Combining weather condition data to predict traffic flow: a GRU-based deep learning approach. *The Institution of Engineering and Technology Intelligent Transport Systems*, 12(7), 578-585.
- Zhang, J. D., Feng, Y. J., Shi, F. F., Wang, G., Ma, B., Li, R. S. & Jia, X. Y. (2016). Vehicle routing in urban areas based on the oil consumption weight-dijkstra algorithm. *IET Intelligent Transport Systems*, 10(7), 495-502.



ÖZGEÇMİŞ

