



# Kasislerin Yakıt Tüketimine Etkisinin RNN, LSTM, GRU Tekrarlayan Derin Öğrenme Algoritmaları ile Tespiti

Mustafa Fatih Tosun<sup>1\*</sup> , Ali Şentürk<sup>2</sup> 

<sup>1</sup> Isparta Uygulamalı Bilimler Üniversitesi, Elektrik-Elektronik Mühendisliği Bölümü, Isparta, Türkiye

<sup>2</sup> Isparta Uygulamalı Bilimler Üniversitesi, Elektrik-Elektronik Mühendisliği Bölümü, Isparta, Türkiye

mtosun32@gmail.com, alisenturk@isparta.edu.tr

## Öz

Bu çalışmada, trafiği düzenlemek için kullanılan kasislerde, araçların yavaşlama ve hızlanmasının yakıt tüketimine etkisinin belirlenmesi amaçlanmıştır. Bunun için, kasis bulunan güzergâhlarda kullanılan aracın OBD-II portundan Arduino ile gerçek zamanlı hız ve yakıt tüketimi verileri alınmıştır. Alınan veriler ön işleme tabi tutulmuştur. Yakıt tüketimini tahmin etmek için Tekrarlayan Sinir Ağları (RNN), Uzun Kısa Süreli Bellek (LSTM), Geçitli Tekrarlayan Birim (GRU) tekrarlayan derin öğrenme modelleri geliştirilmiştir. Ön işlemden geçen veriler modellerin eğitiminde kullanılmıştır. Geliştirilen modellerde hiperparametre optimizasyonu yapılmıştır. Böylece katman sayısı, katmanlardaki hücre sayısı, hücrelerin aktivasyon fonksiyonları ve öğrenme oranı belirlenmiştir. Doğrulama setinde en düşük %63 ortalama kare hatası elde edilmiştir. Geliştirilen modeller kullanılarak farklı kasisler ve hız senaryolarının yakıt tüketimine olan etkileri tahmin edilmeye çalışılmıştır. Kasislerden geçiş için belirlenen hız ve zaman verileri kullanılarak yakıt tüketiminin kasis etkisi boyunca %16,30 ile %31,03 arasında artırdığı sonucuna ulaşılmıştır.

**Anahtar kelimeler:** Yakıt Tüketimi, Kasis, OBD-II, Derin öğrenme, RNN, LSTM, GRU

## The Detection of the Effect of Bumpers on Fuel Consumption with RNN, LSTM, GRU Recurrent Deep Learning Algorithms

### Abstract

This study is aimed to determine the effect of vehicle deceleration and acceleration on fuel consumption in the bumps which are used to regulate traffic. For this, real-time fuel consumption and speed data are acquired with Arduino from the OBD-II port of the vehicle driven on routes with bumps. Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) deep learning models are developed to predict fuel consumption. The preprocessed data is used to train the models. Hyperparameter optimization is conducted in the developed 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 63% in the validation set. The effects of different speed scenarios on fuel consumption are predicted by using the models. In conclusion, fuel consumption increased between 16.30% and 31.03% during the impact of the bumps, by using the speed and time calculated for the bumps.

**Keywords:** Fuel Consumption, Bump, OBD-II, Deep Learning, RNN, LSTM, GRU

### 1. Giriş (Introduction)

Ülkemizde artan trafiğin düzenlenmesi için yollarda hız kesici engeller (kasis) kullanılmaktadır. Her ne kadar kasisler trafiği düzenlemede etkili yöntemler olsa da standartlara uygun yapılmamış

kasisler araçların gereğinden çok fazla yavaşlamasına veya zarar görmesine, sürüş konforunu olumsuz etkilenmesine sebep olmaktadır. Trafik yoğunluğunun az olduğu saatlerde de kasisten geçen aracın hızının, hız sınırlarının altına düşmeye zorlanmaktadır.

Günümüzde bütün araçlarda standart olarak Elektronik Kontrol Ünitesi (Electronic Control Unit-

\* Sorumlu yazar.  
E-posta adresi: mtosun32@gmail.com

Alındı : 6 Temmuz 2022  
Revizyon : 26 Eylül 2022  
Kabul : 24 Aralık 2022

ECU) bulunmaktadır. Araçlarda ECU ile diğer elektronik cihazlar arasındaki iletişim Kontrol Alan Ağı (Control Area Network - CAN) ile sağlanmaktadır. Yerleşik Teşhis (On-board Diagnostic - OBD-II) CAN veri yoluna bağlı olan, arıza teşhisi ve aracın verilerinin raporlanması için kullanılan bir ara yüzüdür.

Bu çalışmada, 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. Derin öğrenme algoritmalarından elde edilen sonuçlara göre kasislerin farklı sürüş senaryolarında yakıt tüketimine etkisinin analizleri yapılmıştır.

Yakıt tüketimi tahmini ile ilgili çalışmalar aşağıda verilmiştir. 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 birçok bilgi ile birleştirip GSM alt yapısını kullanarak bir merkeze gönderilmiş ve kullanıcılar internet üzerinden bu bilgilere ulaşabilmesi sağlanmıştır. 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) yaptığı çalışmada arıza tespit cihazı (OBD-II) kullanarak bir veri toplama ortamının nasıl oluşturulabileceğini ve bir araçtan nasıl veri elde edileceğini açıklamıştır. Çalışma arızalı termostatu olan bir arabadan toplanan veri analizinin ön sonuçlarını içermektedir. Kowalik ve Szpyrka (2019) yaptığı çalışmada bir arabadan gerçek zamanlı veri alımı için veri toplama ortamı oluşturmuşlardır. OBD-II kullanılarak aracın 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.

Wickramanayake ve Bandara (2016) yaptığı 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

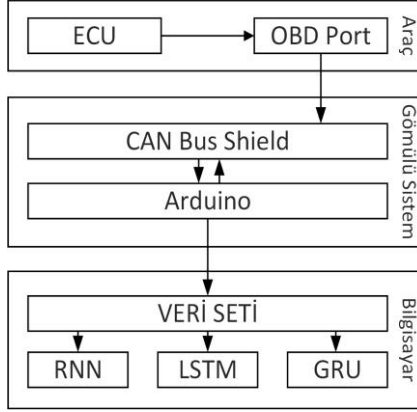
rastgele orman, gradyan azalma ve sinir ağları metodları karşılaştırmalı olarak kullanılmıştır. Syahputra (2016) yaptığı çalışmada, nöro-bulanik mantık yöntemi kullanarak araç yakıt tüketimini tahmin etmiştir. Elde edilen sonuçlara göre yakıt tüketimine etki eden faktörler silindir sayısı, yer değiştirme, beygir gücü, ağırlık, hızlanma ve model yılı olarak belirlemiştir. Perrotta vd. (2017) yaptığı çalışmada damperli kamyonların yakıt tüketimi tahmininde Destek Vektör Makinesi (SVM) (Support Vector Machine), Rastgele Orman (RF) (Random Forest) ve Yapay Sinir Ağı (ANN) (Artificial Neural Networks) modellerini geliştirmiş ve performansları karşılaştırmışlardır. Yakıt tüketimini etkileyen sensör ve yol bilgilerini toplamışlardır. 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) yaptığı çalışmada 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) yaptığı çalışmada konteyner yük gemisinin yakıt tüketimini makine öğrenmesi ile tahmin etmişlerdir. Ana motor devri, ana motor silindiri 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 işlemleri gerçekleştirilmiştir.

## ***2. Materyal ve Metodoloji (Material and Methodology)***

Bu çalışmada kasislerin yakıt tüketimine etkisinin belirlenmesi amaçlanmıştır. Veri toplama işlemi için 2009 model Ford Fiesta marka araç kullanılmıştır. Araçtan veriler Arduino Uno ve CAN veri yolu kartı (CAN Bus shield) ile OBD-II üzerinden alınmıştır. Elde edilen veriler Seri haberleşme (UART) ile bilgisayara aktarılmıştır. Bu veriler yakıt tüketimini tahmin etmede kullanılacak

olan derin öğrenme modelleri için veri setini oluşturmuştur.

Yapılan çalışmada aracın yakıt tüketimini hesaplamada, tekrarlayan derin öğrenme algoritmaları olan RNN, LSTM ve GRU kullanılmıştır. Tasarlanan sistemin blok diyagramı Şekil 1’de verilmiştir.



Şekil 1. Blok diyagram (Block diagram)

### 2.1. Yerleşik teşhis - OBD-II (On board diagnostic - OBD-II)

CAN, mikrodenetleyiciler ve protokolü destekleyen elektronik cihazlar arasında merkezi sisteme ihtiyaç duymadan haberleşmelerini sağlayan çift yönlü seri haberleşme protokolüdür. CAN otomotiv sektöründe araç içi iletişimi sağlamaktadır (Lokman vd., 2019).

OBD (on-board diagnostics - yerleşik teşhis) araçta bulunan bütün elektronik aşşamların birbiriyle haberleşmesini sağlayan CAN veri yoluna 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-II sistemleri standart bir teşhis konektörüne sahiptir ve standart elektrik sinyali protokollerini ve mesajlaşma formatlarını kullanmaktadır. Konektörde aracın ECU’suna erişim için iletişim pinlerinin yanı sıra güç kaynağı pinleri bulunmaktadır (Vilgenoğlu, 2019).

### 2.2. Araçtan veri alınması ve yakıt tüketiminin hesaplanması (Acquiring data from the vehicle and calculating the fuel consumption)

Araçtan verilerin alınması için CAN bus shield modülü kullanılmıştır. Bu modül, ATmega328 mikrodenetleyici üzerine kurulu bir Arduino Uno kartı tarafından programlanıp ve kontrol edilebilir.

Kart üzerinde MCP2515 CAN bus kontrol entegresi ve MCP2551 CAN alıcı/verici entegresi bulunmaktadır (Amarasinghe vd., 2015).

Araçtaki yakıt tüketiminin belirlenmesi amacıyla hız ve MAF değerleri kullanılmıştır. Bu veriler heksadesimal formatta araçta üretilir. Araçtaki anlık değerleri okumak için OBD Mod-1 kullanılır. İstek yapılan verilerin her birisinin kendisine ait PID numarası vardır. Örneğin aracın anlık olarak hız verisine ulaşmak için OBD-II üzerinden 0x0D PID’si ile istek yapılır. Araçtan bu isteğe cevap olarak gönderilen 8 bayt verinin dördüncü baytı (A) aracın km/saat biriminde anlık hız verisidir. Verilen cevap 0 ile 255 arasında bir değerdir.

Araçta bulunan kütle hava akış sensöründen elde edilen verilerle MAF değerini hesaplamak mümkündür. Bunun için OBD-II üzerinden 0x10 PID’si ile istek yapılır. Araç bu isteğe sekiz bayt veri ile cevap verir. Döndürülen bu verinin dördüncü ve beşinci baytları (A ve B) ile Eşitlik 1’de gösterildiği gibi MAF değeri hesaplanır. Hesaplanan değer birimi gram/sn olup 0 ile 655 arasındadır.

$$MAF = \frac{256A + B}{100} \quad (1)$$

Yakıt tüketimi, 100 kilometrelik sürüş boyunca tüketilen yakıtın litre cinsinden ifade edilebilir. Bu çalışmada yakıt tüketiminin anlık hesaplanmasında dizel motorlu araç kullanılmıştır. 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 kullanılacak diğer parametreler OBD-II’den alınabilir. Bu parametreler Motor Yakıt Oranı (0x5E) ile aracın hız bilgisi veya MAF (0x10) değeri veya emme manifoldu basıncı (0x0B) ve emme havası sıcaklığıdır (0x0F) (Meseguer, 2015). Bu çalışmada yakıt tüketimini hesaplamada Eşitlik 2 kullanılmıştır.

$$Yakıt\ Tüketimi = \frac{MAF \times 3600}{AFR_A \times FD} \quad (2)$$

Eşitlik 2’de 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. Gerçek hava-yakıt oranı dizel araçlar için 14,5 benzinli araçlar için 14,7’dir. Yakıt yoğunluğu ise dizel araçlarda 750 g/l, benzinli araçlar için 820 g/l’dir.

### 2.3. Veri setinin ön işlenmesi (Preprocessing the data set)

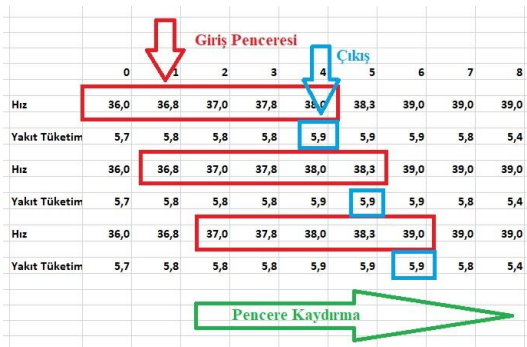
Araçtan veriler düzenli zaman aralıklarında alınamamaktadır. Dolayısıyla ardışık iki veri

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.

Doldurulan verilerden veri seti elde etmek için belirli aralıklarla örnekleme yapılmıştır. Örnekleme zamanı 300 milisaniye 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. Dolayısıyla hazırlanan veri setinde 300 ms aralıklı zamanlarda hız ve yakıt verileri bulunmaktadır.

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

Sistemi eğitmek için pencere kaydırma yöntemi kullanılmıştır. Pencere boyutu 5, 10 ve 15 veriden oluşmaktadır. Şekil 2'de 5'li pencere gösterilmektedir.



Şekil 2. Pencere kaydırma metodu (Sliding window method)

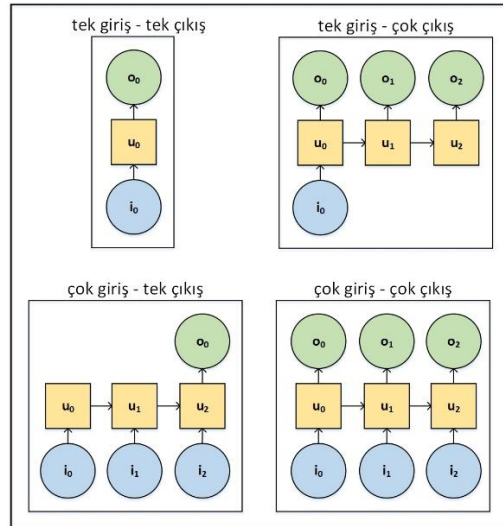
Araçtan OBD-II arayüzünden veriler 3 farklı sürüşte alınmıştır. Bu sürüşler aracın farklı tipte kasislerdeki 25 adet geçişten oluşmuştur. Bununla birlikte aracın farklı hızlarda sürülmesi, bu hızlara ulaşma ve yavaşlama verileri de mevcuttur. Her 3 veri grubu kendi içinde 10 eşit kısma ayrılmıştır. Her kısım ise kendi içinde %60'ı eğitim ve %40'ı test için bölünmüştür.

Makine öğrenmesi algoritmalarında kullanılan verilerin aralığının farklı olması algoritmanın başarısını olumsuz yönde etkilediği için 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 normalize edilerek 0 ile 1 aralığında olması sağlanmıştır.

## 2.4. Tekrarlayan derin öğrenme algoritmaları (Recurrent deep learning algorithms)

Bu çalışmada 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 belirleyebilmek için daha önceki hızlarının unutulmaması gerekmektedir. Dolayısıyla önceki adımları hatırlayabilen derin öğrenme yapılarına ihtiyaç duyulmaktadır. Bu çalışmada belirtilen adımların işlendiği birimler ünite olarak isimlendirilmiştir.

RNN, LSTM, GRU tekrarlayan derin öğrenme algoritmalarının farklı kullanım şekilleri bulunmaktadır. Bunlar tek giriş - tek çıkış, tek giriş - çok çıkış, çok giriş - tek çıkış, çok giriş - çok çıkıştır. Bu kullanım şekilleri Şekil 3'te gösterilmiştir. Şekilde  $i$  giriş,  $o$  çıkış,  $u$  ise üniteleri temsil etmektedir. Yapılan çalışmada çok giriş - tek çıkış türü tekrarlayan derin öğrenme yapısı kullanılmıştır. Giriş verisi 5, 10 veya 15 adet 300 ms aralıklı ardışıl hız, çıkış verisi ise yakıt tüketimidir.



Şekil 3. Tekrarlayan derin öğrenme modelleri (Recurrent deep learning models)

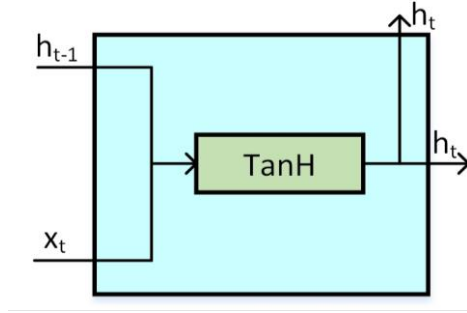
Yapılan çalışmada tahminler ortalama kare hata (mean square error - MSE) metriği ile ölçülmüştür. MSE derin öğrenme modellerinin tahmini ile gerçek değerler arasındaki farkın karelerinin toplamının tahmin sayısına bölünmesi ile elde edilir. MSE'nin matematik ifadesi Eşitlik 3'te gösterilmiştir.

$$MSE = \frac{1}{m} \sum_{i=1}^m (Y_i - \hat{Y}_i)^2 \quad (3)$$

## 2.5. Tekrarlayan sinir ağları - RNN (Recurrent Neural Networks - RNN)

RNN'ler ardışıl verilerin veya zaman serilerinin tahmin edilmesinde kullanılan bir çeşit yapay sinir ağıdır (Somuncu vd., 2021). Bir RNN ünitesinin yapısı Şekil 4'te gösterilmiştir.

RNN ünitesinde yapılan işlemin ifadesi Eşitlik 4'te gösterilmiştir.  $W$  ağırlık,  $b$  bias,  $h_t$  hücrenin  $t$  zamanındaki çıktı verisi,  $x_t$  ise  $t$  zamanındaki girdi verisi için kullanılmıştır.



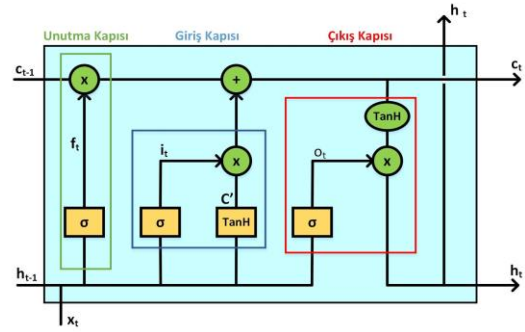
Şekil 4. RNN ünitesi (RNN unit)

$$h_t = f(W_h h_{t-1} + W_x x_t + b) \quad (4)$$

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. Bundan dolayı önceki zamanlardaki verinin sonraki zamanlara aktarılması için RNN modellerini temel alan LSTM ve GRU modelleri geliştirilmiştir (Fu vd., 2018).

## 2.6. Uzun kısa süreli bellek - LSTM (Long Short Term Memory - LSTM)

LSTM algoritması, RNN algoritmasının önceki ünitelerindeki ağırlık güncellemelerinde yaşanan gradyan değerinin sifıra yaklaşması sonucunda eğitimin kaybolması durumunu çözen bir algoritmadır. Bu algorithmada RNN yöntemindeki eksiklikleri gideren hafıza blokları adı verilen birimler bulunmaktadır. Hafıza bloklarında üç farklı kapı mevcuttur. Bu kapılar 5'te gösterildiği gibi sırasıyla unutma, giriş ve çıkış kapılarıdır (Wang vd., 2020).

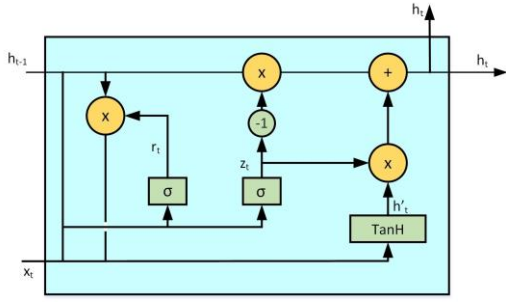


Şekil 5. LSTM ünitesi (LSTM unit)

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. Giriş kapısında ise önceki gizli durumdan gelen veri ile girdi verisi sigmoid aktivasyon fonksiyonunda işleme tabi tutulmaktadır. Unutma kapısında olduğu gibi aktivasyon işlemi sonucunda elde edilen değer bir değerine yakınsa verinin hedef çıktığı 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).

## 2.7. Geçitli tekrarlayan birimler - GRU (Gated Recurrent Units - GRU)

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). GRU ünitesi Şekil 6'da gösterilmiştir.



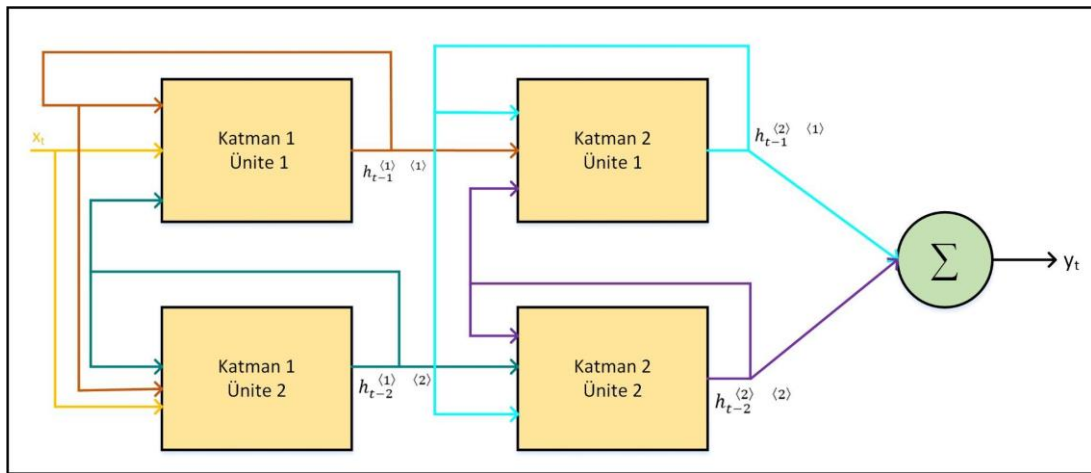
Şekil 6. GRU ünitesi (GRU unit)

### 3. Bulgular ve Tartışma (Results and Discussion)

Araç hızı ile yakıt tüketimini tahmin etmek için farklı derin öğrenme modelleri geliştirilmiştir. Bu modellerde kullanılmak üzere veri setinde her bir veri 300 ms'deki hız ve yakıtı göstermek üzere toplam 8553 veri bulunmaktadır. RNN, LSTM ve GRU modellerinde 5, 10 ve 15 veriden oluşan 3 farklı pencere boyutu kullanılmıştır. Verilerin %60'ı eğitim için, %40'ı ise test için ayrılmıştır.

RNN, LSTM, GRU tekrarlayan derin öğrenme modelleri için hiperparametre optimizasyonu yapılmıştır. Derin öğrenme modelinde tekrarlayan 1, 2 veya 3 tekrarlayan derin öğrenme katmanı bulunmaktadır. Her bir katmanda 8-256 arasında ünite bulunmaktadır.

Örnek olması için 2 katman 2 üniteli tekrarlayan derin öğrenme yapısı Şekil 7'de gösterilmiştir.



Şekil 7. Tekrarlayan derin öğrenme yapısı (Recurrent deep learning structure)

Bellek katmanlarından sonra 1-3 arasında tam bağlantı katmanı mevcuttur. Düğüm ve ünitelerde Sigmoid, TanH, ReLU aktivasyon fonksiyonları kullanılmıştır. Eğitim esnasında aşırı öğrenmenin önüne geçmek için her bir katmandan sonra %0 ile %40 arasında bırakma yapılmıştır. Öğrenme oranları  $10^{-1}$ ,  $10^{-2}$ ,  $10^{-3}$  veya  $10^{-4}$  olarak seçilmiştir. Hiperparametre optimizasyonunda kullanılan parametreler Şekil 8'de gösterilmektedir.

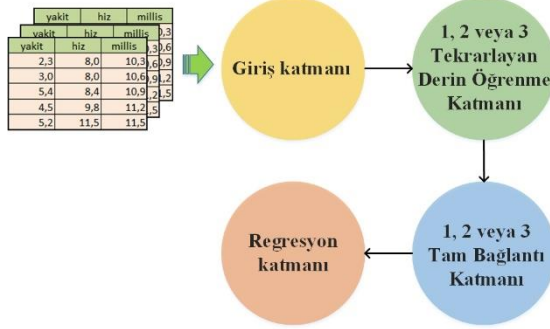
Üç pencere boyutunda her bir model rastgele arama ile her birisinde 2 yürütme olmak üzere 300 denemeye tabi tutulmuştur. Dolayısıyla toplam 2700 farklı modelde 5400 yürütme gerçekleştirilmiştir. Kullanılan hiperparametrelerle elde edilen sistemin test setindeki performansı 5 adımda iyileşmeyi bıraktığında sistemin eğitimi durdurulmuştur.

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 8. Hiperparametre optimizasyonu parametreleri (Hyperparameter optimization parameters)

### 3.1. Önerilen modellerin performans bulguları (Performance results of the proposed models)

Verilerin eğitilmesi ve test işleminde kullanılan tekrarlayan derin öğrenme modellerinin temel yapısı Şekil 9'da gösterilmektedir. Bu şekilde, tekrarlayan derin öğrenme katmanında RNN, LSTM veya GRU ünitelerden bir tanesi bulunmaktadır.



**Şekil 9.** Tekrarlayan derin öğrenme modelleri blok diyagramı (Block diagram of the recurrent deep learning models)

Hiperparametre optimizasyonunda tahmin doğrulukları en yüksek RNN, LSTM, GRU modelleri için elde edilen eğitim ve doğrulama kayıpları Tablo 1'de verilmektedir.

**Tablo 1.** Derin öğrenme modelleri en düşük hata sonuçları (Lowest error results of the deep learning models)

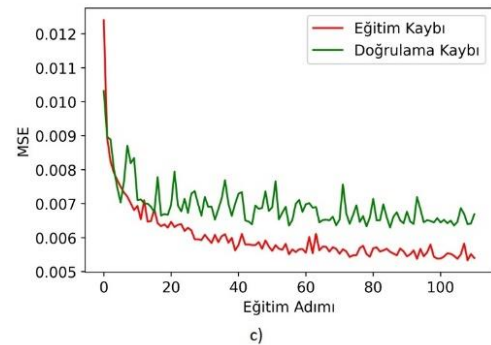
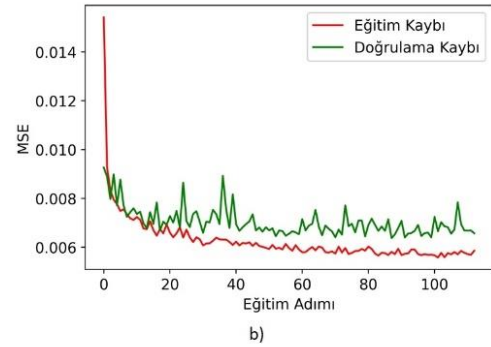
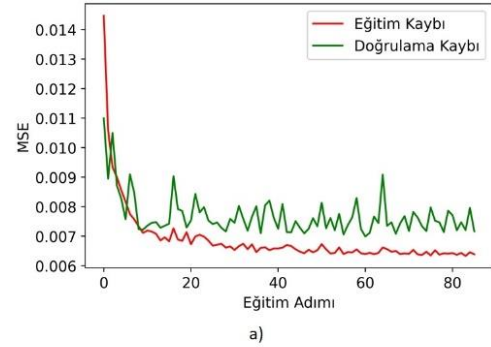
Yöntem	Pencere Sayısı	Eğitim Kayıpları	Doğrulama Kayıpları
RNN	5	0,0064	0,0071
RNN	10	0,0057	0,0064
RNN	15	0,0056	0,0062
LSTM	5	0,007	0,0075
LSTM	10	0,0127	0,011
LSTM	15	0,0058	0,0063
GRU	5	0,0063	0,007
GRU	10	0,0077	0,0072
GRU	15	0,006	0,0064

Tablo 1'de eğitim ve doğrulama kaybını (MSE) en aza indiren modellerin pencere boyutu 15 olan RNN, LSTM ve GRU modelleri olduğu görülmektedir. En performanslı çalışan model LSTM-15'te eğitim kaybının 0,0058'e ve doğrulama kaybının 0,0063'e kadar düştüğü görülmüştür. Bununla birlikte farklı denemelerde RNN, LSTM ve GRU modellerinin tahmin hatalarının birbirine çok yakın çıktığı gözlenmiştir.

Tablo 2'de hiperparametre optimizasyonu sonucu tahmin hatası en düşük olan derin öğrenme modelleri, bu modellerde kullanılan aktivasyon fonksiyonları, modeldeki ünite-düğüm sayıları ve eğitilen parametre sayıları verilmiştir. Hiperparametre optimizasyonu sonucu hatası en düşük modellerin çoğunun tek katman tekrarlayan

derin öğrenme ve tam bağlantı katmanından oluştuğu görülmektedir. Tek katmanlı modeller dışında LSTM-10'da üç tane LSTM katmanı ve GRU-15'te iki tane GRU katmanı olduğu görülmüştür. Birden fazla tam bağlantı katmanı olan modellerde hata oranları yüksek çıkmıştır. Kullanılan aktivasyon fonksiyonları TanH, ReLU ve Sigmoid olmasına rağmen TanH ve ReLU Sigmoid fonksiyonuna 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 regresyon katmanı bulunmaktadır.

Şekil 10'da RNN modellerinin eğitim ve doğrulama kayıpları verilmektedir. Şekil 10'da pencere boyutu 5, 10 ve 15 olan RNN modelinin eğitim ve doğrulama kayıpları gösterilmektedir. RNN-5 86. adımda, RNN-10 113. adımda ve RNN-15 111. adımda eğitim işlemini sonlandırmıştır.



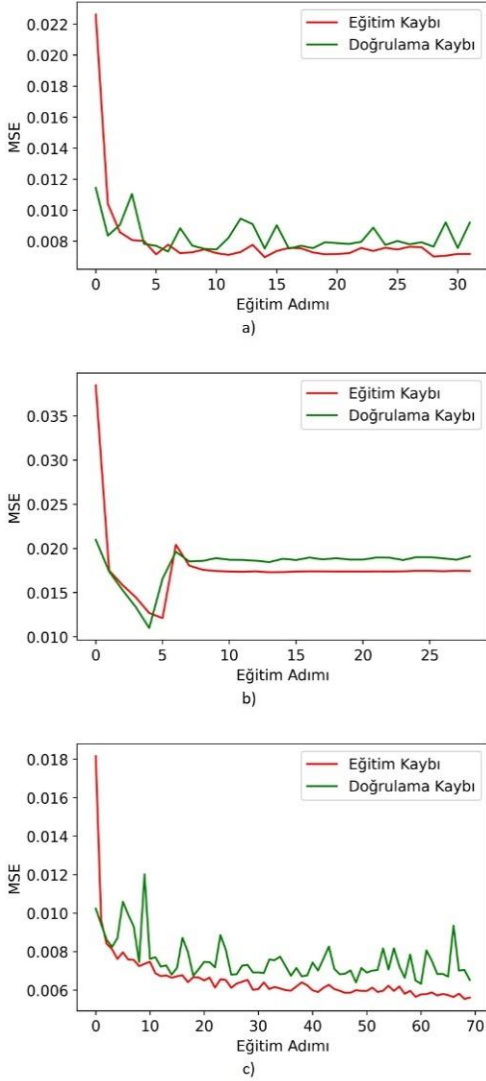
**Şekil 10.** a) 5, b) 10, c) 15 pencere boyutlu RNN eğitim ve doğrulama kayıpları (Train and validation losses of a) 5, b) 10, c) 15 window-sized RNN)

**Tablo 2.** Hatayı en aza indiren hiperparametreler ve toplam parametre sayıları (Hyperparameters that minimize the error and the total number of parameters)

Model		Aktivasyon	Ünite- Düğüm Sayısı	Bırakma Yüzdeleri	Eğitilen Parametre Sayısı	Toplam Eğitilen Parametre Sayısı
RNN 5	RNN Katmanı	TanH	128	30	16640	49921
	Tam Bağlantı Katmanı	ReLU	256	10	33024	
	Regresyon Katmanı		1		257	
RNN 10	RNN Katmanı	TanH	128	20	16640	33281
	Tam Bağlantı Katmanı	ReLU	128	10	16512	
	Regresyon Katmanı		1		129	
RNN 15	RNN Katmanı	TanH	64	20	4224	12673
	Tam Bağlantı Katmanı	ReLU	128	10	8320	
	Regresyon Katmanı		1		129	
LSTM 5	LSTM Katmanı	TanH	32	10	1088	5441
	Tam Bağlantı Katmanı	ReLU	128	20	4224	
	Regresyon Katmanı		1		129	
LSTM 10	1. LSTM Katmanı	TanH	32	40	1088	8113
	2. LSTM Katmanı	TanH	16	0	784	
	3. LSTM Katmanı	TanH	64	0	5184	
	Tam Bağlantı Katmanı	Sigmoid	16	10	1040	
	Regresyon Katmanı		1		17	
LSTM 15	LSTM Katmanı	TanH	256	10	66048	99073
	Tam Bağlantı Katmanı	ReLU	128	10	32896	
	Regresyon Katmanı		1		129	
GRU 5	GRU Katmanı	ReLU	256	20	66048	99073
	Tam Bağlantı Katmanı	ReLU	128	20	32896	
	Regresyon Katmanı		1		129	
GRU 10	GRU Katmanı	TanH	64	30	4224	21121
	Tam Bağlantı Katmanı	ReLU	256	20	16640	
	Regresyon Katmanı		1		257	
GRU 15	1. GRU Katmanı	TanH	128	20	16640	82817
	1. GRU Katmanı	TanH	128	20	32896	
	Tam Bağlantı Katmanı	TanH	256	10	33024	
	Regresyon Katmanı		1		257	

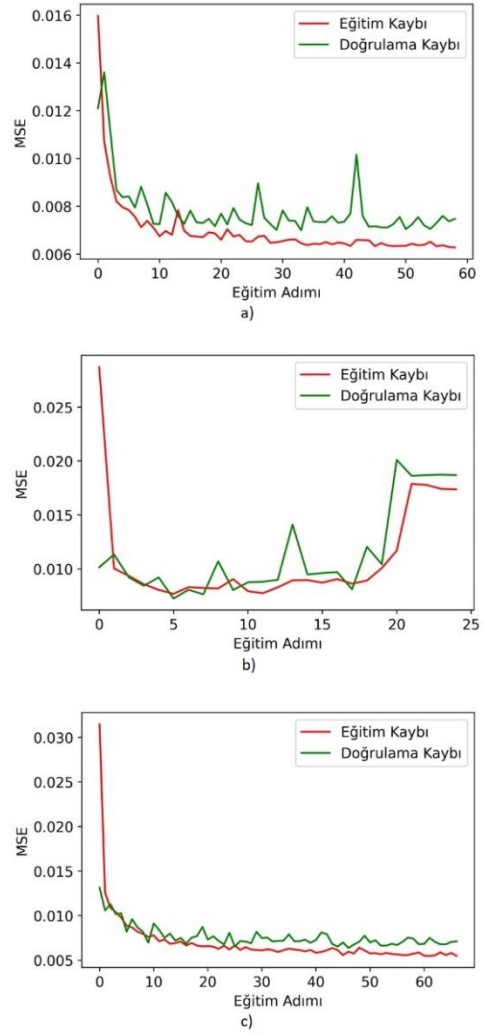


Şekil 11’de LSTM modellerin eğitim ve doğrulama kayıpları verilmektedir. Şekil 11 a’da pencere boyutu 5 olan LSTM, b’de pencere boyutu 10 olan LSTM ve c’de pencere boyutu 15 olan LSTM modelinin eğitim ve doğrulama kayıpları gösterilmektedir. LSTM-5 32. adımda, LSTM-10 30. adımda ve LSTM-15 86. adımda eğitim işlemini sonlandırmıştır.



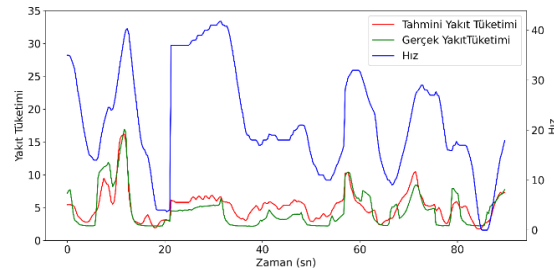
Şekil 11. a) 5, b) 10, c) 15 pencere boyutlu LSTM eğitim ve doğrulama kayıpları (Train and validation losses of a) 5, b) 10, c) 15 window-sized LSTM)

Şekil 12’de GRU modellerin eğitim ve doğrulama kayıpları verilmektedir. Şekil 12 a’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 gösterilmektedir. GRU-5 67. adımda, GRU-10 59. adımda ve GRU-15 67. adımda eğitim işlemini sonlandırmıştır.

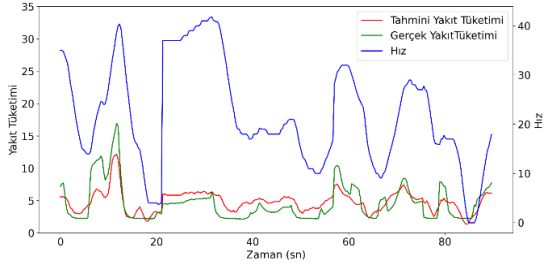


Şekil 12. a) 5, b) 10, c) 15 pencere boyutlu GRU eğitim ve doğrulama kayıpları (Train and validation losses of a) 5, b) 10, c) 15 window-sized GRU)

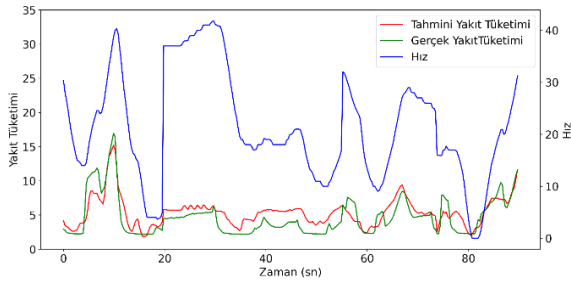
Şekil 13, Şekil 14 ve Şekil 15’te tahmin doğruluğu en yüksek olan modellerin yakıt tüketimi tahminleri gösterilmiştir. Sol y eksenini yakıt tüketimini, sağ y eksenini ise hızı göstermektedir. Mavi çizgi ölçülen hızı, yeşil çizgi ölçülen yakıt tüketimini, kırmızı çizgi ise modellerin tahmini sonucu elde edilen yakıt tüketimini göstermektedir.



Şekil 13. RNN-5 modeli yakıt tüketim tahmini, hız ve gerçek yakıt tüketimi (RNN-5 model fuel consumption prediction, speed and actual fuel consumption)



**Şekil 14.** LSTM-5 modeli yakıt tüketim tahmini, hız ve gerçek yakıt tüketimi (LSTM-5 model fuel consumption prediction, speed and actual fuel consumption)



**Şekil 15.** GRU-15 modeli yakıt tüketim tahmini, hız ve gerçek yakıt tüketimi (GRU-15 model fuel consumption prediction, speed and actual fuel consumption)

Kasislerdeki yakıt tüketiminin aracın markası, modeli, yakıt türü, motor hacmi, ağırlığı, araçtaki yük miktarı, yolun eğimi gibi pek çok parametreye göre değişkenlik göstereceği aşikardır. Bu çalışmanın amacı ise kasislerin ortalama bir araçta, yakıt tüketimine yaklaşık etkisinin araştırılmasıdır. Bundan dolayı makine öğrenmesi modelleri sadece hız ve yakıt tüketimi bilgileri ile eğitilmiştir. Yine de yakıt tüketim tahminlerinin karşılaştırılması Tablo 3'te verilmiştir. Tabloda RMSE kök ortalama kare hatası (root mean square error), MAE ortalama mutlak hata (mean absolute error) için kullanılmıştır.

**Tablo 3.** Modellerin önceki çalışmalarla karşılaştırması (Comparison of the models with the previous studies)

Çalışma	Model	RMSE	MAE	R <sup>2</sup>
Wickrama-nayake ve Bandara (2016)	RF	0,0404	0,02295	-
	Gradient Boosting	0,0471	0,02585	-
	ANN	0,04740	0,02756	-
Perrotta vd. (2017)	SVM	5,12	3,56	0,83
	RF	4,64	3,21	0,87
	ANN	4,88	3,46	0,85
Panapakidis vd. (2020)	SVM	2,190	1,697	-
	LSTM	1,936	1,398	-
Bu çalışma	RNN-15	0,2487	0,0599	0,664
	LSTM-15	0,2522	0,0612	0,654
	GRU-15	0,2459	0,0604	0,638

Tablo 4'te gerçek sürüş esnasında kasislerden geçişlerdeki hız ve zaman bilgileri gösterilmektedir. Bu tabloda "Giriş Hızı" kasis için yavaşlamaya başlamadan önceki hızı, "En Düşük Hız" kasisin kaynaklı en düşük

hızı, "Yavaşlama Süresi" Giriş Hızı ile En Düşük Hız arasında geçen süreyi, "Çıkış Hızı" kasisin etkisinin geçtiği ve kasisin önceki benzer sabit hıza geçildiği hızı, "Hızlanma Süresi" En Düşük Hızla Çıkış Hızı arasındaki geçen süreyi ifade etmektedir.

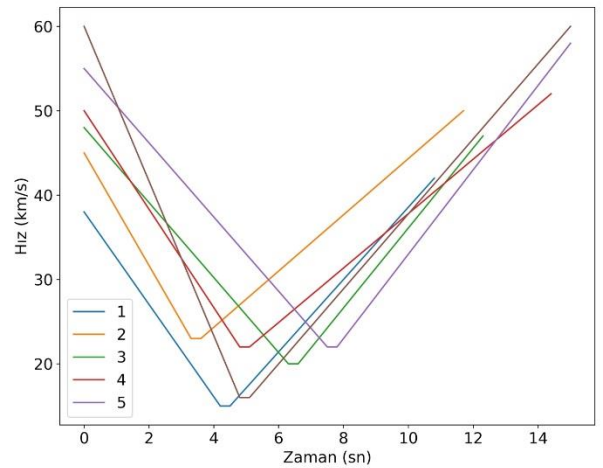
**Tablo 4.** Gerçek zamanlı kasis geçiş bilgileri (Real-time bump transition data)

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

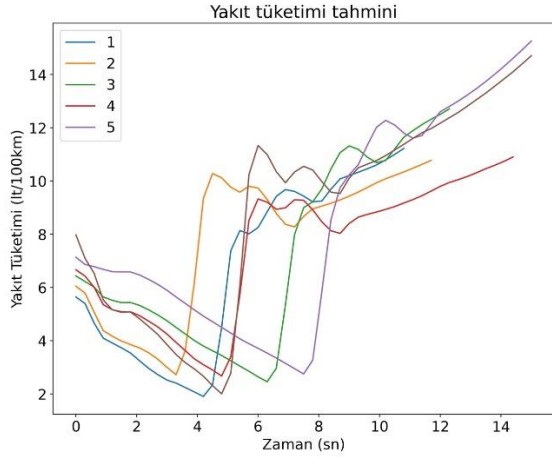
Tablo 4'teki veriler kullanılarak hızın belirtilen zamanlardaki doğrusal olarak azalma ve artmaları Şekil 16'da gösterilmiştir. Burada y ekseninde kasisin geçiş esnasında oluşan hızı, x ekseninde ise kasisin geçme süresini göstermektedir.

Şekil 17'de bulunan grafik Şekil 16'da hesaplanan kasis yavaşlama ve hızlanmaları için tahmin edilen yakıt tüketimlerini göstermektedir. Yakıt tüketim tahmini hiperparametre optimizasyonu sonucu en performanslı çalışan model olan RNN-15 kullanılarak yapılmıştır.

Tablo 5'te kasisin geçiş sırasında harcanan yakıt miktarı ve kasisin etkisi boyunca gidilen mesafe verilmektedir. Ayrıca aynı mesafe kasis olmadan sabit "Giriş Hızında" gidildiğinde harcanan yakıt miktarı ve kasisli yolda sabit hıza göre fazla harcanan yakıt miktarı yüzdesel olarak verilmiştir.



**Şekil 16.** Kasisin geçiş esnasında doğrusal yavaşlama ve hızlanma (Linear deceleration and acceleration during bump transition)



**Şekil 17.** Kasisten geçiş sırasındaki yakıt tüketim tahmini (Fuel consumption prediction during bump transition)

**Tablo 5.** Kasisten kaynaklı yakıt artış oranları (Fuel increase rates due to bump)

Kasis Tahmin Yakıt (lt/100km)	Sabit Hız Yakıt (lt/100km)	Fazla Harcanan Yakıt Yüzdesi	Gidilen Mesafe (m)
0,0062	0,0048	28,38	085,38
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

Tablo 5'te görüldüğü gibi aynı mesafedeki yolda kasis olmadığı durumda fazla harcanan 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.

#### 4. Sonuçlar (Conclusions)

Bu çalışmada trafik akışını yavaşlatmak için yaygın olarak kullanılan hız kesici tümseklerin yakıt tüketimine etkisi incelenmiştir. Bunun için araç sürüşü esnasında hız ve yakıt verileri alınmış ve bu veriler tekrarlayan derin öğrenme modellerinin eğitilmesinde kullanılmıştır. Eğitilen modellerde hiperparametre optimizasyonu yapılmış ve en düşük %63 ortalama kare hatası 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 geçerli 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 parametrelerin kasilerdeki yakıt sarfiyatını etkilemesi kaçınılmazdır. Ancak yapılan çalışma ortalama bir aracın kasisten kaynaklı fazla yakıt sarfiyatı ile ilgili genel bir fikir vermektedir. Şehirlerde çok sayıda

bulunan kasilerin üzerinden geçen araçların sayısı da göz önüne alındığında ciddi miktarda yakıt tüketimine sebep olabileceği anlaşılmaktadır. Kasilerde belirli standartların uygulanmaması, geçiş esnasında aracın zarar görmesi, sürüş konforunun azalmasına ve günün tüm saatlerinde kasisten kaynaklı gereğinden fazla yavaşlamaya da sebep olmaktadır. Bundan dolayı kasilerin kullanılması yerine araçların hızlarının elektronik olarak tespit edilmesi gibi daha modern yöntemlerin kullanılmasının daha uygun olduğu düşünülmektedir.

#### Teşekkür (Acknowledgment)

Bu çalışmada yer alan derin öğrenme modellerinin eğitimleri TÜBİTAK ULAKBİM, Yüksek Başarım ve Grid Hesaplama Merkezinde (TRUBA kaynaklarında) gerçekleştirilmiştir.

#### Kaynaklar (References)

- 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. <https://doi.org/10.1109/ICTER.2015.7377695>
- Coşkun, U. 2008. "Controller Area Network 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. <https://doi.org/https://doi.org/10.1016/j.jnca.2014.02.013>
- Fu, Y., Lou, F., Meng, F., Tian, Z., Zhang, H., Jiang, F., 2018. "An Intelligent Network Attack Detection Method Based on RNN". 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), 483–489. <https://doi.org/10.1109/DSC.2018.00078>
- Kowalik, B. 2018. "Introduction to car failure detection system based on diagnostic interface". In 2018 International Interdisciplinary PhD Workshop (IIPhDW (pp. 4-7). IEEE.
- Kowalik, B., Szpyrka, M. 2019. "Online environment for data acquisition from car sensors". Automatyka/Automatics, 23(1), 7-7.
- Lokman, S.-F., Othman, A. T., Abu-Bakar, M.-H., 2019. "Intrusion detection system for automotive Controller Area Network (CAN) bus system: a review". EURASIP Journal on Wireless Communications and Networking, 2019(1), 184. <https://doi.org/10.1186/s13638-019-1484-3>
- 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), 443–448. <https://doi.org/10.1109/CCNC.2015.7158016>
- Perrotta, F., Parry, T., Neves, L. C., 2017. "Application of machine learning for fuel consumption modelling of

- trucks". 2017 IEEE International Conference on Big Data (Big Data), 3810–3815.  
<https://doi.org/10.1109/BigData.2017.8258382>
- Somuncu, E., Atasoy, N., 2021. "Evrişimli tekrarlayan sinir ağı ile metin görüntüleri üzerinde karakter tanıma uygulaması gerçekleştirilmesi". Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 37 (1), 17-28. DOI: 10.17341/gazimmfd.866552
- Syahputra, R., 2016. "Application of neuro-fuzzy method for prediction of vehicle fuel consumption". Journal of Theoretical and Applied Information Technology, 86(1).
- Şen, B., 2020. "Estimating instant fuel consumption by machine learning and improving fuel consumption". Yüksek Lisans Tezi, Galatasaray Üniversitesi Fen Bilimleri Enstitüsü
- Uyanık, T., Karatug, Ç., Arslanoğlu, Y., 2020. "Machine learning approach to ship fuel consumption: A case of container vessel". Transportation Research Part D: Transport and Environment, 84, 102389.  
<https://doi.org/https://doi.org/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, J., Du, Y., Wang, J., 2020. "LSTM based long-term energy consumption prediction with periodicity. Energy", 197, 117197.  
<https://doi.org/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., 2016. "Fuel consumption prediction of fleet vehicles using Machine Learning: A comparative study". 2nd International Moratuwa Engineering Research Conference, MERCon 2016, 90–95.  
<https://doi.org/10.1109/MERCon.2016.7480121>
- Zhang, D., Kabuka, M, R., 2018. "Combining weather condition data to predict traffic flow: a GRU-based deep learning approach IET Intelligent Transport Systems", 12(7), 578-585.