KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI

DOKTORA TEZİ

HAREKETLİ NESNELERE AİT GEZİNGE VERİLERİNİN YENİ BİR YÖNTEM VE İLERİ BESLEMELİ YAPAY SİNİR AĞLARI İLE İNDİRGENMESİ

SAMET DİRİ

KOCAELİ 2023

KOCAELİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ

ANABİLİM DALI

DOKTORA TEZİ

HAREKETLİ NESNELERE AİT GEZİNGE VERİLERİNİN YENİ BİR YÖNTEM VE İLERİ BESLEMELİ YAPAY SİNİR AĞLARI İLE İNDİRGENMESİ

SAMET DİRİ

Prof. Dr. Mehmet YILDIRIM		
Danışman,	**Kocaeli Üniversitesi**	**.....**
Prof. Dr. Sevinç İlhan OMURCA		
Jüri Üyesi,	**Kocaeli Üniversitesi**	**.....**
Prof. Dr. Celal ÇEKEN		
Jüri Üyesi,	**Sakarya Üniversitesi**	**.....**
Doç. Dr. Adem TUNCER		
Jüri Üyesi,	**Yalova Üniversitesi**	**.....**
Dr. Öğr. Üyesi Erkan KIYAK		
Jüri Üyesi,	**Piri Reis Üniversitesi**	**.....**

Tezin Savunulduğu Tarih: 16.06.2023

ETİK BEYAN VE ARAŞTIRMA FONU DESTEĞİ

Kocaeli Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez/proje çalışmasında,

- Bu tezin/projenin bana ait, özgün bir çalışma olduğunu,
- Çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı,
- Bu çalışma kapsamında elde edilen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi,
- Bu çalışmanın Kocaeli Üniversitesi'nin abone olduğu intihal yazılım programı kullanılarak Fen Bilimleri Enstitüsü'nün belirlemiş olduğu ölçütlere uygun olduğunu,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Tezin/Projenin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez/proje çalışması olarak sunmadığımı,

beyan ederim.

[X] Bu tez/proje çalışmasının herhangi bir aşaması hiçbir kurum/kuruluş tarafından maddi/alt yapı desteği ile desteklenmemiştir.

[ ] Bu tez/proje çalışması kapsamında üretilen veri ve bilgiler ..... tarafından ..... no'lu proje kapsamında maddi/alt yapı desteği alınarak gerçekleştirilmiştir.

Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçları kabul ettiğimi bildiririm.

……..

Samet DİRİ

YAYIMLAMA VE FİKRİ MÜLKİYET HAKLARI

Fen Bilimleri Enstitüsü tarafından onaylanan lisansüstü tezimin/projemin tamamını veya herhangi bir kısmını, basılı ve elektronik formatta arşivleme ve aşağıda belirtilen koşullarla kullanıma açma izninin Kocaeli Üniversitesi'ne verdiğimi beyan ederim. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak, tezimin/projemin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanımı bana ait olacaktır.

Tezin/projenin kendi özgün çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin/projenin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanılması zorunlu metinlerin yazılı izin alarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

Yükseköğretim kurulu tarafından yayınlanan ***"Lisanüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge"*** kapsamında tezim aşağıda belirtilen koşullar haricinde YÖK Ulusal Tez Merkezi/ Kocaeli Üniversitesi Kütüphaneleri Açık Erişim Sisteminde erişime açılır.

☐ Enstitü yönetim kurulu kararı ile tezimin/projemin erişime açılması mezuniyet tarihinden itibaren 2 yıl ertelenmiştir.
☐ Enstitü yönetim kurulu gerekçeli kararı ile tezimin/projemin erişime açılması mezuniyet tarihinden itibaren 6 ay ertelenmiştir.
☒ Tezim/projem ile ilgili gizlilik kararı verilmemiştir.

……

Samet DİRİ

ÖNSÖZ VE TEŞEKKÜR

Uydu teknolojilerindeki gelişme ve mobil cihazların gündelik yaşamın her alanındaki yaygın kullanımı, hareketliliğe ait verilerin işlenmesi ve saklanmasına yönelik çalışmaları her geçen gün artırmaktadır. Bu tez çalışmasında, Küresel Uydu Navigasyon Sistemleri sayesinde elde edilen verilerin, elde edildiği anda indirgenmesi için kullanılabilecek bir yöntem geliştirilmeye çalışılmıştır. Ayrıca, geliştirilen bu yöntemin test edilmesi amacıyla, toplanan verileri bulut veri tabanında saklayan bir mobil uygulama geliştirilmiştir. Geliştirilen yöntem ile yüksek indirgeme ve düşük hata oranı elde edilmesi hedeflenmiştir.

Doktora eğitimim süresince, yoğun çalışma temposuna rağmen her türlü konuda benimle bilgi ve birikimlerini paylaşan, çalışmamın tamamlanması için zaman ve mekân gözetmeksizin benimle çalışan ve bu tezde büyük emeği olan değerli danışmanım Prof. Dr. Mehmet YILDIRIM'a sonsuz teşekkür ederim.

Ayrıca, çalışmam sırasında fikirleri ile bana yol gösteren, beni destekleyen hocalarım Prof.Dr. Sevinç İLHAN OMURCA, Prof.Dr. Celal ÇEKEN, Doç. Dr. Suhap ŞAHİN ve Dr. Selçuk ÖĞÜTCÜ'ye, destekleri için Kocaeli Üniversitesi Bilgisayar Uygulama ve Araştırma Merkezinde görev yapan çalışma arkadaşlarıma ve Ayhan GÜLTEKİN'e teşekkür ederim.

Son olarak, doğduğum günden bugüne kadar sevgisini ve desteğini benden bir an olsun esirgemeyen annem Ayşe DİRİ'ye, babam Nebi DİRİ'ye, kardeşlerim Semih DİRİ ve Seda Diri BURÇAK'a, yüksek lisans ve doktora eğitimim boyunca her koşulda yanımda olan değerli eşim Elif Ayşe DİRİ'ye ve biricik kızım Zeynep İdil DİRİ'ye tüm destekleri ve gösterdikleri sabır için teşekkürü bir borç bilirim. Bu süreçte bana her türlü konuda yardımcı olan ancak ismini hatırlayamadığım herkese teşekkürlerimi sunuyorum.

Haziran – 2023 Samet DİRİ

İÇİNDEKİLER

ETİK BEYAN VE ARAŞTIRMA FONU DESTEĞİ i
YAYIMLAMA VE FİKRİ MÜLKİYET HAKLARI ii
ÖNSÖZ VE TEŞEKKÜR iii
İÇİNDEKİLER iv
ŞEKİLLER DİZİNİ vi
TABLOLAR DİZİNİ viii
SİMGELER VE KISALTMALAR DİZİNİ ix
ÖZET xi
ABSTRACT xii
1. GİRİŞ 1
2. GEZİNGE İNDİRGEME YÖNTEMLERİ 5
2.1. Küresel Seyrüsefer Uydu Sistemleri 5
2.1.1. GNSS'lerin Çalışma Prensipleri 6
2.1.2. Yörüngede Bulunan GNSS'ler 7
2.1.2.1. BeiDou Seyrüsefer Uydu Sistemi 7
2.1.2.2. Galileo 8
2.1.2.3. GLONASS 8
2.1.2.4. GPS 9
2.1.2.5. IRNSS 10
2.1.2.6. QZSS 10
2.2. Gezinge Nedir? 10
2.3. Gezinge İndirgeme Yöntemleri 12
2.4. GNSS Veri Kümeleri ve Dosya Biçimleri 22
2.4.1. GNSS Veri Kümeleri 22
2.4.1.1. University of Central Florida (UCF) Veri Kümesi 22
2.4.1.2. Microsoft GeoLife Veri Kümesi 23
2.4.1.3. Movebank Veri Kümesi 23
2.4.1.4. Ulusal Fırtına Merkezi (NHC) Veri Arşivi 24
2.4.1.5. FlightRadar24 Veri Arşivi 24
2.4.2. GNSS Verisi Dosya Biçimleri 24
2.5. Mesafe ve Hata Ölçüm Yöntemleri 25
2.5.1. Mesafe Ölçüm Yöntemleri 25
2.5.1.1. Öklid Mesafesi 25
2.5.1.2. Manhattan Mesafesi 26
2.5.1.3. Haussdorff Mesafesi 27
2.5.1.4. Frechet Mesafesi 28
2.5.2. Hata Ölçüm Yöntemleri 29
2.5.2.1. Ortalama Mutlak Hata 29
2.5.2.2. Ortalama Karesel Hata 30
2.5.2.3. Kök Ortalama Kare Hatası 30
3. GELİŞTİRİLEN İNDİRGEME YÖNTEMİ 32
3.1. Trigonometrik Tanımlar 32
3.2. İndirgeme Yöntemi 34
3.3. Performans Değerlendirmesi 40
4. YAPAY SİNİR AĞI MODELİ 49
4.1. Yapay Sinir Ağı Nedir? 49

4.1.1. Aktivasyon Fonksiyonları 53
4.1.1.1. Sigmoid Aktivasyon Fonksiyonu 53
4.1.1.2. Hiperbolik Tanjant Aktivasyon Fonksiyonu 54
4.1.1.3. ReLU (Rectified Linear Unit) Aktivasyon Fonksiyonu 55
4.1.1.4. Softmax Aktivasyon Fonksiyonu 56
4.2. TensorFlow 56
4.3. Yapay Sinir Ağı Modeli Eğitimi 58
4.3.1. Kullanılan Veri Kümesi 58
4.3.2. Model Eğitimi 58
4.4. Performans Değerlendirmesi 61
5. GELİŞTİRİLEN MOBİL UYGULAMA 69
5.1. Mobil Uygulamada Kullanılan Teknolojiler 69
5.1.1. Android Studio 70
5.1.2. Dart Dili ve Flutter Çerçevesi 71
5.2. Mobil Uygulamanın İşlevleri 73
5.3. Mobil Uygulamada Kullanılan Veri Modeli 73
6. SONUÇLAR VE ÖNERİLER 76
KAYNAKLAR 79
KİŞİSEL YAYIN VE ESERLER 86
ÖZGEÇMİŞ 88

ŞEKİLLER DİZİNİ

Şekil 2.1. GPS'in çalışma şekli ve segmentleri 9
Şekil 2.2. Gezinge verisi örneği 11
Şekil 2.3. Orijinal ve indirgenmiş gezinge örneği 13
Şekil 2.4. Öklid mesafesi örneği 26
Şekil 2.5. Manhattan mesafesi örneği 27
Şekil 2.6. Hausdorff mesafesi örneği 28
Şekil 3.1. Trigonometrik ifadeler ve tanımları 33
Şekil 3.2. Yöntemin açıklaması için kullanılan örnek gezinge 37
Şekil 3.3. Yöntemin çalışması sırasında kullanılan 3 hücreli pencereleme sistemi örneği 38
Şekil 3.4. Orijinal gezinge ve yöntemin çalışması sonucunda elde edilen indirgenmiş gezinge örneği 39
Şekil 3.5. Önerilen yöntemin "20090522094234.plt" dosyası üzerinden görsel kıyaslaması 41
Şekil 3.6. RDP algoritmasının "20090522094234.plt" dosyası üzerinden görsel kıyaslaması 42
Şekil 3.7. Önerilen yöntemin "20081214235553.plt" dosyası üzerinden görsel kıyaslaması 43
Şekil 3.8. RDP algoritmasının "20081214235553.plt" dosyası üzerinden görsel kıyaslaması 44
Şekil 3.9. Önerilen yöntemin "20090608060034.plt" dosyası üzerinden görsel kıyaslaması 45
Şekil 3.10. RDP algoritmasının "20090608060034.plt" dosyası üzerinden görsel kıyaslaması 46
Şekil 3.11. Önerilen yöntemin "20090608060034.plt" dosyası üzerinden görsel kıyaslaması 48
Şekil 3.12. RDP algoritmasının "20090608060034.plt" dosyası üzerinden görsel kıyaslaması 48
Şekil 4.1. Yapay sinir hücresinin modellenmesi (Cantemir, 2019) 49
Şekil 4.2. YSA'ya ait örnek ağ modeli 50
Şekil 4.3. Sigmoid fonksiyonu 54
Şekil 4.4. Hiperbolik tanjant fonksiyonu (Sharma ve diğ., 2020) 55
Şekil 4.5. Relu fonksiyonu (Sharma ve diğ., 2020) 56
Şekil 4.6. Google Colab üzerinde Python dili kullanarak tensörleri yeniden şekillendiren kod örneği 57
Şekil 4.7. Eğitilen ağın modeli 59
Şekil 4.8. Eğitilen modelin her bir devir için çalışma süresi grafiği 60
Şekil 4.9. Eğitilen modelin doğruluk grafiği 60
Şekil 4.10. Eğitilen modelin kayıp grafiği 61
Şekil 4.11. YSA modelinin sonuçlarının "20090522094234.plt" dosyası üzerinden görsel kıyaslaması 63
Şekil 4.12. Önerilen yöntemin sonuçlarının "20090522094234.plt" dosyası üzerinden görsel kıyaslaması 64
Şekil 4.13. YSA modelinin sonuçlarının "20081214235553.plt" dosyası üzerinden görsel kıyaslaması 65

Şekil 4.14. Önerilen yöntemin sonuçlarının “20081214235553.plt” dosyası üzerinden görsel kıyaslaması 66
Şekil 4.15. YSA modelinin sonuçlarının “20090608060034.plt” dosyası üzerinden görsel kıyaslaması 67
Şekil 4.16. Önerilen yöntemin sonuçlarının “20090608060034.plt” dosyası üzerinden görsel kıyaslaması 68
Şekil 5.1. (a) Geliştirilen mobil uygulamanın ana ekranı, (b) geliştirilen mobil uygulamanın ana menüsü..... 69
Şekil 5.2. Android Studio ekran örneği..... 70
Şekil 5.3. Dart dili kullanılarak yazılmış bir ilk durum kodu örneği 71
Şekil 5.4. Flutter çerçevesi kullanılarak yazılmış bir liste parçacığı örneği..... 72
Şekil 5.5. Firebase konsolu ekranı..... 74
Şekil 5.6. Google Firebase Gerçek Zamanlı Bulut Veri Tabanına ait girdi örnekleri 75

TABLOLAR DİZİNİ

Tablo 2.1. Literatürde var olan indirgeme algoritmalarının karşılaştırılması 21
Tablo 2.2. ".plt" uzantılı dosya formatı örneği 23
Tablo 2.3. Mesafe Ölçüm Yöntemleri Kıyaslaması 29
Tablo 3.1. Önerilen yöntemin sözde kodu 35
Tablo 3.2. Önerilen yöntemin ve RDP algoritmasının sonuçlarının "20090522094234.plt" dosyası üzerinden kıyaslanması 40
Tablo 3.3. Önerilen yöntemin ve RDP algoritmasının sonuçlarının "20081214235553.plt" dosyası üzerinden kıyaslanması 42
Tablo 3.4. Önerilen yöntemin ve RDP algoritmasının sonuçlarının "20090608060034.plt" dosyası üzerinden kıyaslanması 44
Tablo 3.5. Önerilen yöntemin ve RDP algoritmasının sonuçlarının "20090608060034.plt" dosyası üzerinden kıyaslanması 47
Tablo 4.1. YSA modeli ve geliştirilen yöntemin sonuçlarının "20090522094234.plt" dosyası üzerinden kıyaslanması 62
Tablo 4.2. YSA modeli ve geliştirilen yöntemin sonuçlarının "20081214235553.plt" dosyası üzerinden kıyaslanması 64
Tablo 4.3. YSA modeli ve geliştirilen yöntemin sonuçlarının "20090608060034.plt" dosyası üzerinden kıyaslanması 66

SİMGELER VE KISALTMALAR DİZİNİ

Δ	: [AB] doğru parçasının x-eksenine göre açısı
ε	: Çizgi segmentinin sınır uzaklığı
μ	: Eşik açısı
Σ	: Seri toplamı
θ	: [AB] ve [BC] doğru parçaları arasındaki yönlerin açısı

Kısaltmalar

ADS-B	: Automatic Dependent Surveillance-Broadcast (Otomatik Bağımlı Gözetim-Yayını)
API	: Application Programming Interface (Uygulama Programlama Arayüzü)
BaaS	: Backend as a Service (Servis Olarak Arka Uç)
BDS	: The BeiDou Navigation Satellite System (BeiDou Seyrüsefer Uydu Sistemi)
CBS	: Coğrafi Bilgi Sistemi
CNN	: Convolutional Neural Network (Evrişimsel Sinir Ağı)
DNN	: Deep Neural Network (Derin Sinir Ağı)
GEO	: Geostationary Earth Orbit (Sabit Konumda Bulunan Dünya Yörüngesi)
GLONASS	: Globalnaya Navigazionnaya Sputnikovaya Sistema (Global Seyrüsefer Uydu Sistemi)
GNSS	: Global Navigation Satellite System (Küresel Uydu Seyrüsefer Sistemi)
GPS	: Global Positioning System (Küresel Konumlama Sistemi)
GPU	: Graphics Processing Unit (Grafik İşlem Birimi)
GSM	: Global System for Mobile Communications (Mobil İletişim için Küresel Sistem)
IDE	: Integrated Development Environment (Tümleşik Geliştirme Ortamı)
IGSO	: Inclined Geosynchronous Orbit (Eğimli Jeosenkron Yörünge)
IRNSS	: Indian Regional Navigation Satellite System (Hindistan Bölgesel Seyrüsefer Uydu Sistemi)
JSON	: JavaScript Object Notation (JavaScript Nesne Gösterimi)
LSTM	: Long Short-Term Memory (Uzun Kısa Süreli Bellek)
MAE	: Mean Absolute Error (Ortalama Mutlak Hata)
MEO	: Middle Earth Orbit (Orta Dünya Yörüngesi)
MSE	: Mean Squared Error (Ortalama Karesel Hata)
NHC	: National Hurricane Center (Ulusal Fırtına Merkezi)
QZSS	: Quasi-Zenith Satellite System (Quasi-Zenith Uydu Sistemi)
RDP	: Ramer-Douglas-Peucker Algoritması
ReLU	: Rectified Linear Unit (Doğrultulmuş Doğrusal Birim)
RFID	: Radio Frequency Identification (Radyo Frekansı ile Tanımlama)
RMSE	: Root Mean Square Error (Kök Ortalama Kare Hatası)
RNN	: Recurrent Neural Network (Yinelemeli Sinir Ağı)
SDK	: Software Development Kit (Yazılım Geliştirme Kiti)
T	: Trajectory (Gezinge)
TF	: TensorFlow
TFLite	: TensorFlow Lite
TPU	: Tensor Processing Unit (Tensör İşlem Birimi)

UCF : University of Central Florida (Orta Florida Universitesi)
VPS : Visual Positioning System (Görsel Konumlandırma Sistemi)
YSA : Yapay Sinir Ağı

HAREKETLİ NESNELERE AİT GEZİNGE VERİLERİNİN YENİ BİR YÖNTEM VE İLERİ BESLEMELİ YAPAY SİNİR AĞLARI İLE İNDİRGENMESİ

ÖZET

Küresel Navigasyon Uydu Sistemlerinden elde edilen konum verilerinin kullanımı günlük yaşamın vazgeçilmez bir parçası halini almıştır. Tarımdan sosyal medyaya, araç filolarından güvenlik uygulamalarına, yabani hayvanlardan fırtınaların hareketlerine kadar her alanda toplanan verilerin depolanması ve işlenmesi her geçen gün artan veri boyutu da dikkate alındığında önemli bir sorun olarak ortaya çıkmaktadır. Depolama ve işleme maliyetlerinin azaltılması amacıyla, bu veriler üzerinde önem seviyesi düşük olan verilerin veri kümesinden atılmasıyla, orijinal gezingeden en az kayıp ile konum verisi elde edildiği anda indirgeme yapılabilmektedir. Birçok araştırmacı hareketli nesnelerin konum verilerinin indirgenmesi alanında matematiksel formüller kullanarak indirgeme yapabilen farklı teknikler önermişlerdir.

Bu tez çalışmasında, yoğun gürültülü koşullar altında, algoritmanın doğruluğu aşırı değişken gezinge şekillerine dayanıklı, seçilen noktaların önem seviyesini hesaplayan, bir düğümün yeri doldurulamazlığını ve komşu düğümler arasındaki uzamsal ilişkileri doğru tespit eden, çevrim içi ve çevrim dışı çalışabilen, daha düşük hesaplama karmaşıklığında çalışan yeni bir konum verisi indirgeme yöntemi geliştirilmiştir. Ayrıca, geliştirilen bu yöntem kullanılarak, yapay sinir ağı eğitiminde kullanılmak üzere bir veri kümesi oluşturulmuş, çevrim içi ve çevrim dışı indirgeme yapabilen bir yapay sinir ağı modeli eğitimi gerçekleştirilmiştir. Önerilmiş olan yöntem ve eğitilmiş olan yapay sinir ağı modeli çalışma kapsamında geliştirilmiş olan mobil uygulama üzerinde koşularak konum verisi indirgemesi yapılmış ve indirgenen konumlar bulut tabanlı veri üzerinde saklanmıştır. Yapılan testler sonucunda, geliştirilen konum verisi indirgeme yönteminin literatürdeki tekniklere kıyasla daha hızlı, daha yüksek hassasiyetle, yüksek indirgeme oranıyla ve konum verisinin elde edildiği anda indirgeme yapabildiği görülmüştür. Buna ilave olarak, eğitimi gerçekleştirilmiş olan yapay sinir ağı modelinin de hem geliştirilen konum indirgemesi yönteminden hem de literatürdeki tekniklere kıyasla daha hızlı, daha yüksek hassasiyetle ve konum verisinin elde edildiği anda indirgeme yapabildiği görülmüştür.

Anahtar Kelimeler: Çizgi Basitleştirme, Gezinge, Gezinge Verisi İndirgeme, GPS, Yapay Sinir Ağı.

REDUCTION OF TRAJECTORY DATA OF MOVING OBJECTS WITH A NOVEL METHOD AND FEED FORWARD ARTIFICIAL NEURAL NETWORKS

ABSTRACT

Using location data obtained from Global Navigation Satellite Systems has become indispensable to daily life. The storage and processing of data collected in every field, from agriculture to social media, from vehicle fleets to security applications, and from wild animals to storm movements, emerges as an important problem considering the ever-increasing data size. In order to reduce the storage and processing costs, the data with low importance on these data can be removed from the data set, and the reduction can be made as soon as the location data is obtained with the least loss from the original trajectory. Many researchers have proposed different techniques in the field of the reduction of position data of moving objects using mathematical formulas.

In this study, a new location algorithm, which can withstand extremely variable trajectory shapes, calculates the importance level of selected points, accurately detects the irreplaceability of a node and spatial relationships between neighboring nodes, can work online and offline, operates at lower computational complexity, under intense noisy conditions. A data reduction method has been developed. In addition, using this developed method, a data set was created to be used in artificial neural network training. An artificial neural network model training that can perform online and offline reduction was carried out. The proposed method and the trained artificial neural network model were run on the mobile application developed within the scope of the study, location data reduction was made, and the reduced locations were stored on cloud-based data. The results of the tests demonstrated that the developed position data reduction method could perform faster, with higher precision, high reduction rate and reduction as soon as the position data is obtained compared to the techniques in the literature. In addition, it has been observed that the artificial neural network model, which has been trained, can perform reductions faster, with higher precision, and when the position data is obtained, compared to both the developed position reduction method and the techniques in the literature.

Keywords: Line Simplification, Trajectory, Trajetory Data Reduction, GPS, Neural Network.

1.GİRİŞ

Geçtiğimiz yüzyılın ikinci yarısında İnternet'in icadı, uydu sistemlerinin geliştirilmesi, mobil cihazların üretilmesi gibi pek çok teknolojik gelişme yaşanmıştır. Bahsedilen teknolojik buluşların kullanımının yaygınlaşmasıyla birlikte üretilen veri miktarı da katlanarak artmış, bu verilerin saklanması ve işlenmesi ise zorunluluk halini almıştır. Uydu teknolojilerindeki gelişmeler ile İnternet'in ve mobil cihazların gündelik yaşamın her alanındaki yaygın kullanımı, hareketliliğe ait verilerin işlenmesi ve amaca uygun bir şekilde saklanmasına yönelik çalışmalar daha da artırmıştır.

Küresel Seyrüsefer Uydu Sistemi (GNSS – ing: Global Navigation Satellite Systems), bölgesel veya küresel ölçekte zamanlama, navigasyon ve konumlandırma hizmetleri sunan bir uydu takımını tanımlayan genel bir ifadedir (Shan-shi ve diğ., 2019; United Nations Office for Outer Space Affairs, 2010; URL-1).

Teknolojik ilerlemeler; giderek daha güçlü, verimli ve taşımayı kolaylaştıran ulaşım araçlarının üretilmesini de sağlamıştır. Bunun yanı sıra, iletişim alanındaki gelişmeler de hareketliliğe (ing: mobility) yeni bir yön vermiştir. GNSS uyduları kullanarak konum bilgisi edinme ve mobil hesaplama tekniklerindeki ilerlemeler; insanlar, hayvanlar, kara, deniz ve hava taşıtları gibi çeşitli hareketli nesnelerin hareketliliğini temsil eden büyük miktarda gezinge verisi üretmektedir (Long ve diğ., 2013; Muckell ve diğ., 2010; Potamias ve diğ., 2006; Sandu Popa ve diğ., 2014; L. Zhao & Shi, 2018; Zheng, 2015).

GNSS'lerin özgür, açık ve güvenilir yapısı günümüz modern hayatını etkileyen ve ayrılmaz bir parçası olan yüzlerce uygulamanın geliştirilmesine yol açmıştır. Hareket yeteneğinin artması ve GNSS'lerin sağladığı güvenilir verinin yardımıyla; evden işe takip ettiğimiz rotaların otomatik olarak tutulması, işlenmesi, çektiğimiz fotoğraflarla belgelenmesi ve hatta bu belgelere konum ve zaman bilgilerinin eklenmesi gereksinimi doğmuştur. Bu, hareketliliklerin verilerini tutan küçük büyük pek çok veri tabanının oluşturulmasına ve hâlihazırda saklanan büyük miktardaki verinin daha da artmasına neden olmuştur. Artan veri miktarının üstesinden gelinmesi ve akıllıca kullanılması amacıyla veri indirgemesi, saklanması ve işlenmesi gibi çeşitli alanlarda daha fazla uygulama ve yöntem geliştirilmiştir ve geliştirilmeye devam etmektedir (URL-1; URL-2; URL-3).

Hareketli bir nesnenin gezingesi, ardışık zamanlı bir konum noktaları dizisi olarak düşünülebilir (Potamias ve diğ., 2006). Bir gezinge genellikle her bir nesne için gözlenen ardışık konum çiftlerini bağlayan bir dizi doğru parçası olarak gösterilir. Öklid düzlemi üzerinde hareket eden bir nesnesinin *T* gezingesi, zaman içindeki zaman damgalı konumlarının sınırsız bir dizisidir.

Gezinge basitleştirme ya da başka bir deyişle gezinge indirgemesi, ham gezingeden hangi noktaların atılması gerektiğine veya ham gezingede hangi noktaların tutulması gerektiğine karar vermek için kullanılan yöntemleri ifade etmektedir (Lin ve diğ., 2017). Büyük gezinge setleri veri depolama, iletme ve işleme işlemleri sırasında zaman ve ekonomik maliyet gibi çeşitli sorunları yaratmaktadır (Lin ve diğ., 2017; Meng ve diğ., 2017; Potamias ve diğ., 2006; Zheng, 2015). Uygun yöntemler kullanılarak, ana karakteristik unsurları korurken, gereksiz bilgileri sıkıştırarak orijinal gezingelerin doğru bir temsili elde edilebilir (S. K. Zhang ve diğ., 2016). Gezinge basitleştirme algoritmaları için önemli noktalardan bir tanesi, bilgi kaybını en aza indirirken gezinge verilerinin boyutunu azaltmaktır (Zheng, 2015). Ham ya da başka bir deyişle orijinal gezingede kalan noktalara basit gezinge denir (Lin ve diğ., 2017). Bu basitleştirilmiş gezinge daha verimli gezinge hesaplamalarında kullanılabilirken bu hesaplamaların sonuçları orijinal gezingelerden çok uzak değildir, ancak işlem zamanı orijinal gezingelere nazaran çok daha kısadır (Lin ve diğ., 2017).

Gezinge indirgemesi literatürde yol, eğri, çizgi veya zincir basitleştirme olarak da kullanılmaktadır. Yalnızca büyük veri işlenmesi ve veri madenciliği alanında değil aynı zamanda Coğrafi Bilgi Sistemlerinde (CBS), görüntü işlemede ve bilgisayar grafikleri gibi birçok uygulamada da gezinge indirgeme yöntemleri kullanılmaktadır (Abam ve diğ., 2014).

Son on yılda özellikle mobil cihazların her alanda kullanılması ve GNSS'ler kullanılarak konum bilgisinin düşük hata oranlarıyla sağlam ve güvenilir bir şekilde elde edilmesi ile toplanan büyük miktardaki gezinge verilerinin işlenmesi, yönetilmesi ve madenciliği için çok sayıda uygulama önerilmiştir (Zheng, 2015). Literatürde yapılan çalışmalar gezinge verilerinde aykırı değerleri, aşırı bükülmeleri ve dönüş noktalarını tespit etmeye çalışmaktadır. Bu tespitler genellikle segment uzunluklarının ve dönüş açılarının bir sonraki konuma göre açısal farkının hesaplanması ve el ile girilen ya da otomatik

hesaplanan eşik değerinin altında olup olmadığının kontrol edilmesiyle yapılmaktadır. Ayrıca, yine nokta sayısını azaltarak doğrusal çizgi bölümlerinden oluşan çizgileri pürüzsüzleştirmek yoluyla indirgeme işlemleri yapılmaktadır (Zheng, 2015).

İndirgenmek istenen veriler çoğunlukla büyük hacimlidir. Bu nedenle, hesaplama karmaşıklığı düşük bir sistem geliştirilmesi hesaplama süresini azaltacaktır. Bu sayede zaman maliyetinin düşürülmesi sağlanacaktır. İndirgeme yöntemleri çevirim içi veya çevirim dışı olarak çalışabilmektedir. Bu sistemin çevirim içi bir şekilde çalışması ve verinin elde edildiği sırada indirgenmesi iletim ve depolama maliyetlerini önemli ölçüde azaltacaktır. Ayrıca, basitleştirme işlemi hem konum bilgilerini hem de yön bilgilerini orijinal gezingeden çok farklı olmayacak biçimde kalite garantili bir şekilde yapılmalıdır.

Yoğun gürültülü koşullar altında, algoritmanın doğruluğu aşırı değişken gezinge şekillerine dayanıklı, seçilen noktaların önem seviyesini hesaplayan, bir düğümün yeri doldurulamazlığını ve komşu düğümler arasındaki uzamsal ilişkileri doğru tespit eden, daha düşük hesaplama karmaşıklığında çalışan daha gelişmiş yöntemlere ihtiyaç vardır.

Bu tez çalışmasında, verinin elde edildiği anda, cihaz üzerinde çevirim içi olarak çalışabilen, düşük hata oranı ve yüksek indirgeme oranına sahip bir yöntem geliştirilmiştir. Geliştirilen bu yöntemin sonuçları kullanılarak bir veri kümesi üretilmiş ve hata oranının geliştirilen yöntemden çok daha düşük olduğu, indirgeme oranının ise bu denli düşük hata oranı nedeniyle kabul edilebilir olduğu bir sinir ağı eğitilmiştir. Bu çalışma dört bölümden meydana gelmektedir:

Birinci bölümde, gezinge kavramından, alt başlıklarda ise literatürde var olan GNSS verisi indirgeme yöntemlerinden, hazır veri kümelerinden, mesafe ölçüm yöntemlerinden ve hata ölçüm yöntemlerinden bahsedilmektedir.

İkinci bölümde, tez çalışması kapsamında geliştirilen matematiksel yöntem yer almaktadır. Yöntemin akışına, kullanılan trigonometrik kavramlara ve parametrelere değinilmektedir.

Üçüncü bölümde, matematiksel yöntem çalıştırılarak oluşturulan veri kümesi kullanılarak eğitilen yapay sinir ağı modeli yer almaktadır. Alt başlıklarında ise yapay sinir ağı ile ilgili tanımlara ve kullanılan teknolojilere yer verilmektedir.

Dördüncü bölümde, hem matematiksel yöntem hem de yapay sinir ağı modeli için farklı veri kümeleri üzerine her iki yöntem de uygulanmakta, yapılan uygulamaların sonuçları irdelenmekte ve elde edilen sonuçlar literatürde var olan çalışmaların sonuçları ile karşılaştırılmaktadır.

Sonuç ve öneriler bölümünde ise bu tez çalışması kapsamında elde edilen bulgular ve sonraki çalışmalar için faydalı olacak önerilere ve yorumlara yer verilmektedir.

2. GEZİNGE İNDİRGEME YÖNTEMLERİ

2.1. Küresel Seyrüsefer Uydu Sistemleri

GNSS'ler, küresel bilgi altyapısında önemli bir yere sahiptirler. GNSS'lerin güvenilir, özgür ve açık yapısı, günümüz modern hayatını her yönüyle etkileyen pek çok uygulamanın geliştirilmesini sağlamıştır. GNSS'ler; cep telefonları, kol saatleri, iş makinaları, nakliye konteynırları, çiftçilik, inşaat, madencilik, haritacılık, uzay, kamu güvenliği, paket teslimatı ve lojistik tedarik zinciri yönetimi, finansal piyasalar, bankacılık sistemleri, büyük iletişim ağları, güç şebekeleri, hassas zaman senkronizasyonu gerektiren uygulamalar gibi hayatımızın geniş bir bölümünde yaygın olarak kullanılmaktadır ve uygulama sahasında verimliliği ciddi oranda artırmaktadır (URL-1; URL-2). GNSS'lerin günümüzdeki yaygın kullanım alanları (United Nations Office for Outer Space Affairs, 2010);

1. Keşif: Devriye ve keşif görevlerinde,
2. Hedef izleme: Havadaki ve yerdeki potansiyel tehditlerin hedef takip sistemi ile izlenmesinde,
3. Atış ve güdüm sistemleri: Hedeflerin tam isabetle vurulmasında ve silah sistemlerinin tam olarak güdümlenmesi,
4. Navigasyon: Bilinmeyen konumlardaki nesnelerin veya mekanların tespitinde,
5. Arama ve kurtarma: Açık ve geniş arazilerdeki arama çalışmalarında,
6. Tarım ve vahşi yaşam: Hayvanların göç yollarının tespiti ve tarımsal faaliyetlerin takibinin yapılmasında,
7. Sosyal medya ve reklamcılık: Mobil uygulamalarda, sosyal medya üzerindeki etiketlemeler ve bölgesel reklamcılık faaliyetleri gerçekleştirilmesinde

şeklinde sınıflandırılabilir.

Günümüzde askerî ve ticarî çıkarlarını korumak isteyen ülkeler bu amaçlarla kendi uydu sistemlerini ve GNSS'lerini geliştirmeye devam etmektedir (McNally & Breithaupt, 1987).

2.1.1.GNSS'lerin Çalışma Prensipleri

GNSS'ler; Uzay Segmenti, Kontrol Segmenti ve Kullanıcı Segmenti olmak üzere üç kısımdan oluşmaktadır. Uzay Segmenti, vericiden gelen sinyalleri toplamaktadır. Bu segment konumu belirlenmek istenen nesnenin ya da canlının konumunu belirlemekte ve yayınlamaktadır. Kullanıcı Segmenti, askeri ve sivil kullanıcıları belirtmektedir. Bu segment sinyali işlemeye yarayan bir işlemciden ve sinyalleri algılamakta kullanılan bir adet algılayıcıdan oluşmaktadır. Kontrol Segmenti, tüm sistemin en verimli ve düzgün biçimde çalışmasını sağlamaktadır. Bu segment ise uyduların belirlenen yörüngelerde dönmesini sağlamakta ve alınan sinyalleri anlık olarak güncellemektedir. Bu sayede, meydana gelebilecek hataların önlenmesi sağlanmaktadır (United Nations Office for Outer Space Affairs, 2010).

GNSS uyduları sinyallerini uzayda belirli bir doğrulukla yayınlamaktadır. Ancak, alınan sinyal uydu geometrisi, atmosfer koşulları, sinyal tıkanması, alıcı sistemin kalitesi ve tasarım özellikleri gibi ek etmenlere bağlı olarak değişmektedir (United Nations Office for Outer Space Affairs, 2010). GNSS uyduları kendi aralarında çok küçük gecikmeler ile sinyaller göndermektedirler. Bu çalışma şekli, yeryüzündeki GNSS alıcılarını tetikleyip, ilgili GNSS cihazının nerede olduğu hakkında net bilgiyi elde etmelerini sağlamaktadır. Örneğin, Küresel Konumlama Sistemini (GPS - ing: Global Positioning System) ele almak gerekirse, her bir GPS uydusu Dünya'nın etrafını günde iki kez turlamaktadır. Konum bilgisini hesaplamak için, yörüngelerinde hareket etmekte olan 24 adet GPS uydusundan yükseklik bilgisine de ihtiyaç varsa en az 4 tanesinden yoksa en az 3 en çok 12 tanesinden gelen oldukça hassas saat sinyallerine ihtiyaç duyulmaktadır (Noe & Myers, 1976). GPS alıcısı bulunan mobil cihazlar açık havada genellikle 4,9 m (16 ft.) yarıçapındaki doğruluk ile konum tespiti yapabilmektedir. Yukarıda bahsedilen faktörlere göre sinyal kalitesindeki değişimler ile hassasiyetin artması ya da azalması söz konusudur. Yani konum doğrulukları sinyal kalitesine bağlı olduğundan binaların, köprülerin ve ağaçların yakınında sinyal bozunumu olabileceğinden hassasiyet azalabilmektedir. İleri teknoloji cihazlar, çift frekanslı alıcılar ve/veya büyütme sistemleriyle konum doğruluğu artmaktadır. Bunlar birkaç santimetre içinde gerçek zamanlı konumlandırmayı ve milimetre düzeyinde uzun vadeli ölçümleri mümkün kılabilmektedir (URL-1).

2.1.2.Yörüngede Bulunan GNSS'ler

Hâlihazırda Dünya'nın çevresinde;

1. Çin'in geliştirmiş olduğu BeiDou (BDS - The BeiDou Navigation Satellite System) (Shan-shi ve diğ., 2019; URL-4)
2. Avrupa Birliği ülkelerinin geliştirmiş olduğu Galileo (Cojocaru ve diğ., 2009; URL-5),
3. Rusya'nın geliştirmiş olduğu GLONASS (Globalnaya Navigatsionnaya Sputnikovaya Sistema) (Cojocaru ve diğ., 2009; URL-6),
4. ABD'nin geliştirmiş olduğu GPS (URL-1),
5. Hindistan'ın geliştirmiş olduğu IRNSS (Indian Regional Navigation Satellite System) (URL-7),
6. Japonya'nın geliştirmiş olduğu QZSS (Quasi-Zenith Satellite System)(URL-8)

olmak üzere çeşitli bölgesel ve küresel seyrüsefer uydu sistemleri mevcuttur (Cojocaru ve diğ., 2009; Shan-shi ve diğ., 2019; United Nations Office for Outer Space Affairs, 2010; URL-1; URL-4; URL-5; URL-6; URL-7; URL-8).

Mevcut sistemler arasında en yaygın ve etkin biçimde kullanılan GNSS, GPS'tir (Noe & Myers, 1976; URL-1).

2.1.2.1. BeiDou Seyrüsefer Uydu Sistemi

BeiDou (BDS), Çin Halk Cumhuriyeti tarafından geliştirilen bölgesel bir GNSS'dir. Daha önce Compass olarak adlandırılmakta olan BDS, 35 uydu ile küresel kapsama alanı sağlayacak şekilde genişletilmeye çalışılmaktadır. Daha yaygın, kapsamlı, entegre ve akıllı bir sistemin 2035 yılına kadar kurulması planlanmaktadır. BDS uzay segmenti, Sabit Konumda Bulunan Dünya Yörüngesinde (GEO – ing: Geostationary Earth Orbit), Eğimli Jeosenkron Yörüngede (IGSO -ing: Inclined Geosynchronous Orbit) ve Orta Dünya Yörüngesinde (MEO – ing: Middle Earth Orbit) bulunan birkaç uydudan meydana gelmektedir (URL-4).

2.1.2.2. Galileo

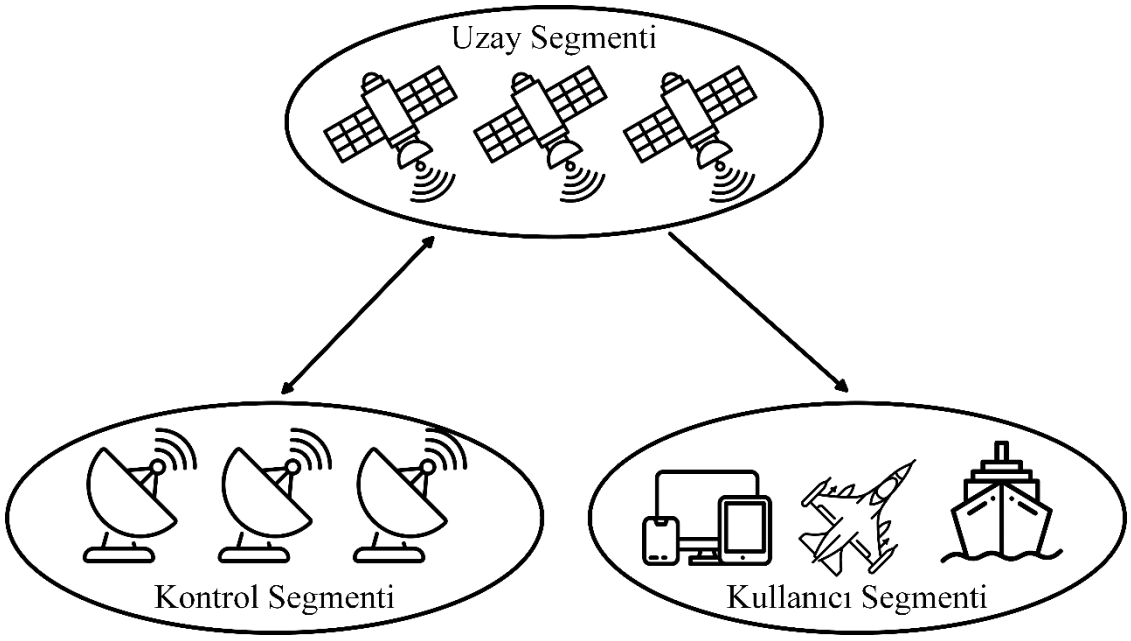
Galileo, Avrupa Birliği üye ülkeleri tarafından geliştirilen küresel bir GNSS'dir. Sistem 2016 yılı itibariyle hizmet sunmaya başlamıştır. Şu anda 24 adet uydusu bulunmaktadır. Galileo, 23.222 km yükseklikteki üç yörünge düzlemine eşit olarak yayılmış olarak ve ekvatora 56 derecelik bir açıyla eğimli şekilde konumlanmış uydulardan meydana gelmektedir. Her bir uydu Dünya'nın etrafını yaklaşık 14 saatte dönmektedir. Herhangi bir operasyonel uydunun devre dışı kalması durumuna karşın, her bir düzlemde iki yedek uydu bulunmaktadır. Sistem, çoğu konumdan, altı ila sekiz uydu her zaman görünür olacak ve konumların ve zamanlamanın birkaç santimetre içinde, çok doğru bir şekilde belirlenmesine izin verecek şekilde tasarlanmıştır. Galileo, kullanıcıların tam konumlarını diğer mevcut GNSS'lerin sunduklarından daha hassas bir şekilde tespit edebilmesini sağlamaktadır. GPS uydu sistemiyle birlikte çalışabilmektedir ve bu sayede Galileo'nun sunduğu hizmetlerin güvenilirliği artmaktadır. İlerleyen aşamalarda GLONASS ile birlikte de çalışması planlanmaktadır (URL-5).

2.1.2.3. GLONASS

GLONASS, Rusya Federasyonu tarafından geliştirilen küresel bir GNSS'dir. Operasyonel sistemin tamamı 24 uydudan oluşmaktadır. GLONASS, Ekim 1982'de "Kosmos-1413" uydusunun lansmanı ile başlamıştır. GLONASS sistemi resmi olarak 1993 yılında faaliyete geçmiştir. 1995 yılında ise tamamen işlevsel bir hale getirilmiştir. Ancak, sivil amaçlarla kullanılması için ekipmanları bulunmadığından GPS'e göre dezavantajlı durumda bulunmaktaydı. Sovyetler Birliğinin yıkılması ve soğuk savaşın bitmesiyle, uzay endüstrisi için ayrılan bütçe azaltıldığından, GLONASS uydu sistemi yeteri kadar destek görmemiş ve bozulmuştur. 2002 yılında yalnızca 7 uydusu kalan GLONASS uydu sistemi sınırlı kullanılabilirlikle bile Rusya topraklarının navigasyon desteği için yetersiz halde bulunmaktaydı. Bu sınırlı sayıdaki uydu nedeniyle konum hesaplama doğruluğu GPS'in çok gerisinde bulunmaktaydı. 2012 yılından itibaren başlatılan yeni çalışmalar ile sistem kararlı hale getirilmiştir ve hizmet sunmaktadır (URL-6).

2.1.2.4. GPS

GPS, yörünge içinde yer alan uyduların düzenli olarak dünyaya gönderdiği kodlanmış atomik saat sinyali sayesinde, dünya üzerinde bulunan bir cihazın kesin konumunun belirlenmesini sağlamak amacıyla, Amerika Birleşik Devletleri Savunma Bakanlığı tarafından oluşturulmuştur ve ABD Savunma Bakanlığına ait bir sistemdir (Noe & Myers, 1976; URL-1). Dünyanın herhangi bir yerindeki kullanıcıya, hava koşullarından bağımsız olarak, son derece hassas üç boyutlu konum düzeltmeleri ve sistem zamanı sağlayan, uydu tabanlı bir radyo navigasyon sistemidir (Leondes & Yonezawa, 1978). Düzenli olarak kodlanmış bilgi iletimi sağlayan ve 24 adet uydudan meydana gelen bir uydu ağına sahiptir (Leondes & Yonezawa, 1978; Noe & Myers, 1976; URL-1). Genel olarak, uydularla cihaz arasındaki mesafeyi (zaman gecikmesini) ölçerek dünya üzerindeki kesin konumun tespit edilmesini mümkün kılmaktadır (Leondes & Yonezawa, 1978; Noe & Myers, 1976; URL-1). GPS'in çalışma şekli ve segmentleri Şekil 2.1'de gösterilmektedir.



Şekil 2.1. GPS'in çalışma şekli ve segmentleri

GPS uyduları güneş enerjisinden elde ettikleri enerji ile çalışmaktadırlar. Uyduların üzerinde, Güneş ışığından faydalanamadığı zamanlarda veya teknik arızalar nedeniyle meydana gelebilecek kesintileri engellemek amacıyla yedek bataryalar bulunmaktadır. Bu bataryaların yanında, yörüngede meydana gelen sapmaları düzeltmek amacıyla küçük roketler yer almaktadır (URL-1).

GPS, 1983 yılında meydana gelen Kore yolcu uçağının düşürülmesi olayına kadar askerî amaçla kullanılmaktaydı. Ancak, bu olayda uçağın hatalı konum saptaması yapması nedeniyle uçuşa yasak hava sahasına girdiği için düşürülmesi ve 269 kişinin hayatını kaybetmesi sonrasında önemi ve gereği anlaşılarak sivil kullanımın önü açılmıştır (URL-9).

2.1.2.5. IRNSS

IRNSS, Hindistan Hükümeti tarafından geliştirilen bölgesel bir GNSS'dir. IRNSS, Hindistan bölgesini ve Hint anakarasındaki kullanıcılara ve birincil hizmet alanı olan sınırlarında 1500 km çevresini kapsayacak şekilde, doğru konum bilgisi hizmeti sunmak için tasarlanmış özerk bir sistemdir. 2018 yılında faaliyete geçmiş olan sistem 7 uydudan oluşmaktadır. Genişletilmiş Hizmet Alanı ise birincil hizmet bölgesi ile bir dikdörtgen tarafından çevrelenen alanın 30° güney ile 50° kuzey enlemleri ve 30° doğu ile 130° doğu boylamları arasında bulunmaktadır. IRNSS Sisteminin, öncelikli servis alanında 20 metreden daha iyi bir konum doğruluğu sağlaması beklenmektedir (URL-7).

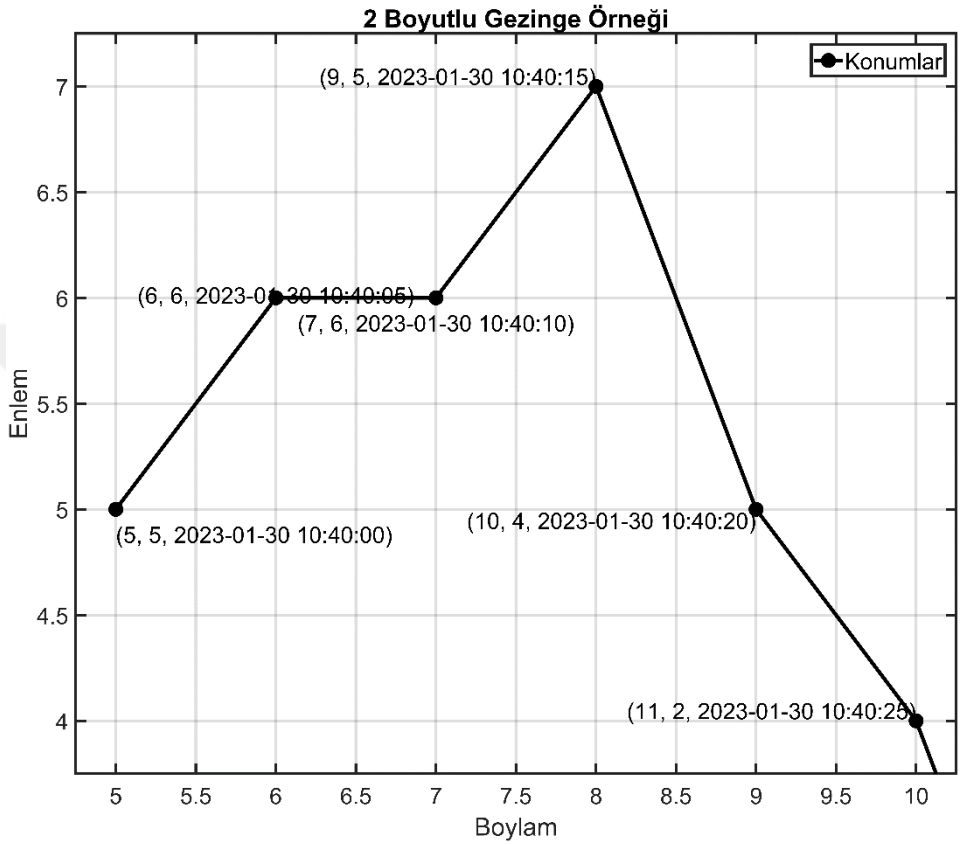
2.1.2.6. QZSS

QZSS, Japonya Hükümeti tarafından geliştirilen bölgesel bir GNSS'dir. QZSS, GPS'in yardımıyla Okyanusya ve Doğu Asya'daki kapsama alanını geliştirmektedir. Sistem Japonya tarafından 2018 yılına kadar 4 uydu ile işletilmekteydi. 2023 yılı sonuna kadar ise otonom kabiliyet geliştirebilmesi amacıyla sistemin 7 uyduya genişletilmesi planlamaktadır. QZSS uyduları Kuzey Yarım Küre'de yaklaşık 13 saat, Güney Yarım Küre'de yaklaşık 11 saat geçirerek Japonya'nın yakınında uzun süre kalmaları sağlanmaktadır (URL-8).

2.2.Gezinge Nedir?

Gezinge (İng: trajectory) kavramı, hareketli bir nesnenin ardışık zamanlı konum noktalarının dizisidir (Potamias ve diğ., 2006). Başka bir deyişle, Öklid düzleminde hareket eden bir nesnenin *T* gezingesi, zaman içindeki zaman damgalı konumlarının sınırsız dizisidir. Yani, değer kümeleri $(id, t_i, x_i, y_i), i \in \{1, ..., N, ...\}$ şeklindedir. Burada; *id* nesneyi, x_i ve y_i *id* nesnesinin bulunduğu koordinatları, t_i ise *id* olarak tanımlanan nesnenin (x_i, y_i) koordinatlarındaki o andaki zaman damgasını ifade etmektedir (Potamias

ve diğ., 2006). Gezingenin aksine yol ise zamandan bağımsız olarak iki nokta arasında güzergâhtır. Gezinge bir nesnenin iki nokta arasında zaman bağlı olarak geçtiği konumlar iken yol iki nokta arasında geçilmesi gereken konumlardır. Gezinge ile yol arasındaki en temel fark gezingenin bir nesneye ait zaman etiketi içermesidir (Gasparetta ve diğ., 2015).



Şekil 2.2. Gezinge verisi örneği

Kaydedildikleri teknolojiye bağlı olarak, gezinge verileri farklı şekillerde sınıflanabilmektedir. McMaster (1987), indirgeme algoritmalarını bağımsız nokta rutinleri (n. noktayı saklama), yerelleştirilmiş işleme rutinleri (mesafe eşiğine göre saklama), genişletilmiş yerel işleme rutinleri (önemli noktalarının ötesinde arama yaparak bulunan noktaları saklama), kısıtlanmamış genişletilmiş yerel işleme rutinleri (kullanım mesafesi minimum yaparak seçileni saklama) ve kısıtlanmış küresel rutinler (belirli bir alan dışında kalan noktaları saklama) şeklinde beş grupta sınıflamıştır (McMaster, 1987). Spinsanti ve diğ. (2013), GPS, GSM ve jeo-sosyal ağ tabanlı gezinge verileri şeklinde sınıflamıştır (Spinsanti ve diğ., 2013). Bu sınıflamaya Pelekis ve Theodoris (2014), radyo frekansı ile tanımlama (RFID – ing: Radio Frequency Identification) tabanlı ve Wi-Fi

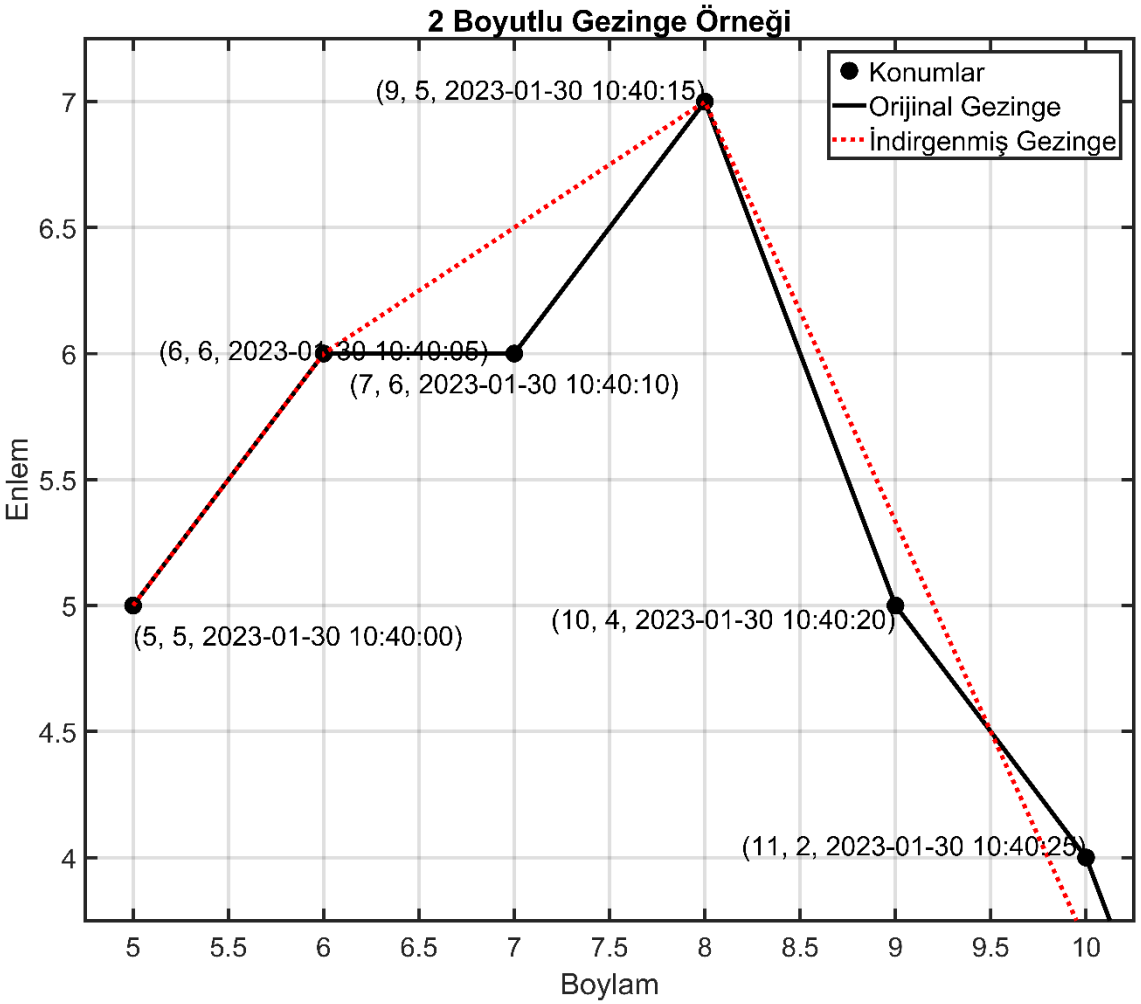
tabanlı gezinge verileri ismiyle iki form daha eklemiştir (Pelekis & Theodoridis, 2014). GPS tabanlı veriler, hareketli nesne tarafından taşınan GPS özellikli bir cihaz tarafından kaydedilen geçici olarak sıralanmış coğrafi koordinat dizilerinden oluşmaktadır. Mobil İletişim için Küresel Sistem (GSM – ing: Global System for Mobile Communications) tabanlı veriler, hareket eden nesnenin geçtiği hücrelerin geçici olarak sıralı tanımlayıcı dizilerinden oluşmaktadır. Jeo-sosyal ağ tabanlı veriler, sosyal medyada bulunan ve coğrafi koordinatların eklendiği içeriklerden meydana gelmektedir. RFID tabanlı veriler, hareketli nesnenin içinden geçtiği RFID okuyucularının bir dizi tanımlayıcısını içerirken, Wi-Fi tabanlı veriler, hareketli nesne ile iletişim kuran bir dizi erişim noktası tanımlayıcısı içermektedir (Bian ve diğ., 2019; Mazimpaka & Timpf, 2016).

Bunların yanı sıra görseller yardımıyla konum tespiti yapabilen Görsel Konumlandırma Sistemi (VPS) de geliştirilmiştir. VPS, bir aygıtın gerçek dünyadaki kesin konumunu ve yönünü belirlemek için bilgisayar görüşü, görüntü işleme ve artırılmış gerçekliği birleştirmektedir. Yalnızca GNSS veya diğer konumlandırma sistemlerine güvenmek yerine görsel ipuçlarını kullanarak konumların tespit edilmesine olanak tanımaktadır. VPS, cihazın kamerasından elde edilen görüntüleri analiz etmektedir. Bu görüntülerdeki görsel işaretler veya referans noktalarını tespit ederek bir veri tabanıyla karşılaştırarak konumu elde etmektedir. Bu işaretler veya referans noktaları, ortamda önceden haritalanmış ve belirli coğrafi konumlarla ilişkilendirilmiş görüntüler, desenler veya nesneler olabilir. VPS, GNSS sinyallerinin zayıf olduğu veya mevcut olmadığı iç mekânda konumlama gibi uygulamalarda kullanılmaktadır (Hyeon ve diğ., 2023; M. J. L. Lee ve diğ., 2021).

2.3. Gezinge İndirgeme Yöntemleri

Gezingenin, çizgiden veya yoldan farklı olarak koordinat bilgisi haricinde zaman etiketi de barındırması nedeniyle gezinge indirgemesi literatürde yol, eğri, çizgi veya zincir basitleştirme gibi farklı isimler altında da çalışılmış olan bir konudur. Gezinge indirgeme yöntemleri; büyük verinin işlenmesi, veri madenciliği, CBS, görüntü işlem ve bilgisayar grafikleri gibi birçok farklı uygulama alanında işlem süresini kısaltmak, hesaplama maliyetini azaltmak ve etkili sonuçlar elde etmek için yaygın bir biçimde kullanılmaktadır (Abam ve diğ., 2014). Şekil 2.3‘te örnek bir gezingenin orijinal ve indirgenmiş halleri gösterilmektedir.

Son on yılda literatürde, özellikle mobil cihazların her alanda kullanılması ve GNSS'ler kullanılarak konum bilgisinin düşük hata oranlarıyla sağlam ve güvenilir bir şekilde elde edilmesi ile toplanan büyük miktardaki gezinge verilerinin işlenmesi, yönetilmesi, saklanması ve madenciliği için çok sayıda farklı uygulamalar önerilmiştir (Zheng, 2015).



Şekil 2.3. Orijinal ve indirgenmiş gezinge örneği

Çizgi genelleme, seçilen gezinge noktalarının birleştirilmesiyle oluşturulan bir dizi doğrusal parçanın kullanıldığı bir gezingeyi temsil etmektedir. İndirgeme, seçilmemiş gezinge noktaları atılarak gerçekleştirilir. Literatürde yapılan çalışmaların önemli bir kısmı maksimum hata eşiğini kullanarak çalışmaktadır. Bu şekilde çalışan yöntemlerde, orijinal gezingeden atılan noktaların, noktaları kapsayan çizgi segmentlerinin belirli bir eşik değerinin altında ya da üzerinde olması durumunda orijinal gezingeden çıkarılmasına karar verilir. Bu tarz yaklaşımlarda önemli olan nokta, birbirini izleyen veri noktalarının doğrusala yakın bir hat üzerinde olması halinde genellikle yüksek bir indirgeme oranı

elde edebilmesidir. Bununla birlikte, gerçek dünyada elde edilen gezingeler, verinin elde edildiği ortamın koşulları nedeniyle ya da bazen eğrilerin olumsuz etkisini artıran düşük örnekleme hızlarının eşlik ettiği çok sayıda eğrilik ve/veya gürültü içerebilir. Bu gibi durumlarda, indirgeme oranını iyileştirmenin yolu, kullanıcının örneğin 20 metre veya 25 metre gibi büyük bir eşik değeri kullanarak, kabul edilebilir seviyenin üzerinde hata payı belirlemesidir, ancak çoğu kullanıcı verisi için daha küçük eşik değerleri ve daha yüksek hassasiyete ihtiyaç duyulmaktadır (Nibali & He, 2015). Gezingeleri indirgemek için çizgi genelleme yaklaşımını benimseyen çok sayıda algoritma vardır. Aşağıda listelenen ve literatürde yapılmış çalışmalardan da görüleceği üzere, var olan gezinge verisi indirgeme yöntemleri çoğunlukla 2 boyutlu gezinge verileri üzerinde koşulmaktadır ve performans değerleri bunlara göre belirlenmektedir.

Douglas ve Peucker (1973), yinelemeli son nokta uyum algoritması olarak da bilinmektedir. Douglas-Peucker algoritması, nokta sayısını azaltarak doğrusal çizgi bölümlerinden oluşan çizgileri pürüzsüzleştirmektedir. Basitleştirilmiş eğri, orijinal eğrinin kaba şeklini korumalıdır, ancak yalnızca orijinal eğriyi tanımlayan noktaların bir alt kümesinden meydana gelmelidir. Kabalaşma derecesi, orijinal noktalar ve basitleştirilmiş eğri arasındaki maksimum mesafeyi tanımlayan ε parametresi tarafından kontrol edilmektedir. Yöntem, veriyi parçalar ve alt parçalar arasında ε parametresinden yakın olan verileri siler (Douglas & Peucker, 1973; Gültekin ve diğ., 2023).

Hershberger ve Snoeyink (1992), klasik Douglas ve Peucker algoritmasının en kötü durumunun, giriş noktası sayısı olan n'de ikinci dereceden olduğunu göstererek "*path hull*" adını verdikleri anlık yol gövdelerinin bölünmesi ve dış bükeyler bulunması prensibine dayanan bir yöntem önermiştir (Hershberger ve diğ., 1992).

Wu ve Marquez (2003), klasik Douglas-Peucker algoritmasında kendi kesişme noktalarının önlenmesini iyileştiren bir yöntem sunmuştur. Her bir iyileştirme adımında, bölünen bir noktanın seçilme şeklindeki ufak değişikliklerde yıldız biçimli bir bölgede bulunan basitleştirilmiş çizgiler için yinelenmiş kesişmelerden kaçınılabileceği gösterilmiştir. Farklı yıldız biçimli bölgelerde bulunan çizgiler arasında ortaya çıkabilecek çatışmaların tanımlanmasını da iyileştirmiştir (Wu & Márquez, 2003).

Lee ve diğ. (2007), mevcut gezinge indirgeme algoritmalarının bir bütün olarak çalışması nedeniyle, bazı ortak alt yörüngeleri kaçırabileceği varsayımıyla TRACLUS ismini verdikleri bir yöntem geliştirmişlerdir. Yöntem bir hareketi bir dizi çizgi parçasına ayıran ve daha sonra benzer çizgi parçalarını bir kümede gruplayan bir bölümleme ve gruplama çerçevesi önermişlerdir. Algoritma iki aşamadan meydana gelmektedir. Bunlar bölümleme ve gruplamadır. İlk aşamada, Minimum Description Length (MDL) prensibini kullanılarak bölümleme yapmaktadır. İkinci aşamada ise, yoğunluğa dayalı çizgi segmenti kümeleme algoritması ile yeniden gruplama yapılmaktadır. Deneysel sonuçlar, TRACLUS'un ortak alt hareketleri gerçek gezinge verilerinden doğru bir şekilde keşfettiğini göstermektedir (J.-G. Lee ve diğ., 2007).

Xuan ve Xu (2011), konum verilerinde var olan dönüş noktalarını tespit etmeye çalışan ve yüksek sıkıştırma oranına sahip bir yöntem geliştirmiştir. Yöntem yüksek sıkıştırma oranına sahip olmasına rağmen aşırı bükülmeler konusunda eksiklikleri bulunmaktadır (Xuan & Xu, 2011).

Long ve diğ. (2013), DPTS (Direction Preserving Trajectory Simplification) algortimasını önermiştir. Yöntem hem analitik hem de deneysel olarak geleneksel gezinge indirgemesinden geniş bir uygulama alanında çalışabilmektedir. Algoritma hareketli nesnenin bir sonraki konumunun açısal farkını hesaplayarak eşik değerin arasında olup olmadığını kontrol etmektedir. DPTS bir polinom-zaman algoritmasıdır ve kalite garantili şekilde çalışmaktadır. Gerçek gezinge verileri ile kapsamlı deneysel değerlendirmeler yapılarak faydası gösterilmektedir (Long ve diğ., 2013).

Katsikouli ve diğ. (2014), büyük hacimli konum ve sensör verilerini indirgemek için kullanılan çevrim içi bir algoritma önermiştir. Yöntem, veri akışının büyük ölçekli keskin özelliklerini tanımlamak için topolojik kalıcılığı kullanmaktır. Gezingeler gibi tek boyutlu veri akışları için, topolojik olarak kalıcı özelliklere dayanan basitleştirmenin çevrim içi olarak sürdürülebileceğini ve her bir yeni veri noktasının O(1) zaman karmaşıklığında işlenebileceğini göstermektedir (Katsikouli ve diğ., 2014).

Li ve diğ. (2014), hem çevrim dışı hem de çevrim içi bir şekilde gezinge verilerini indirgeyebilen 3 algoritma önermiştir. Segment uzunlukları ve dönüş açıları gibi uzamsal bilgilerini dahil edebilmek için ağırlıklandırma yapılmaktadır. Ayrıca, bir noktanın

gürültü derecesini komşularıyla olan konumsal ilişkisine dayanarak ölçmektedirler. Yoğun gürültülü koşullar altında, algoritmanın, doğruluğunun yüksek olduğu ve hesaplama maliyetinin düşük olduğunu göstermişlerdir (Li ve diğ., 2014).

Liu ve diğ. (2015), Bounded Quadrant System (BQS), hata sınırlamalı gezinge sıkıştırması için, dışbükey gövdeleri kullanarak O(1) maliyetli bir çevrim içi algoritması önermiştir. Algoritma, bir başlangıç noktasında ortalanmış dörtgenlerin her birinde dikdörtgen bir sınırlama kutusu ve iki sınırlama çizgisi bulunan bir sanal koordinat sistemi oluşturmaktadır. Bu sayede, koordinat düzlemine göre verileri indirgemektedir (J. Liu ve diğ., 2015).

Fernández-García ve diğ. (2016), Douglas-Peucker yönteminin yeni bir simetrik versiyonunu kullanarak, seçilen noktaların önem seviyesini hesaplayan adaptif bir yöntem önermiştir. Yöntemin iyi bir performansa sahip olduğu gösterilmiştir. Ayrıca, Douglas-Peucker'den farklı olarak, herhangi bir parametre ayarlaması gerektirmemektedir (Fernández-García ve diğ., 2016).

Lin ve diğ. (2016), hız özelliğini dikkate alarak uyarlanabilir bir gezinge indirgeme yaklaşımı olan Adaptive Trajectory Simplification (ATS) algoritmasını önermiştir. Algoritma, hız değerlerini ve gezinge verilerini hız koruma segmentlerine bölerek birkaç aralıkta gruplandırır. İndirgeme işlemi, herhangi bir eşik değerinin el ile ayarlanmadan belirlenebildiği, parçalanabilen her segmentte algoritmanın sadeleştirme yaklaşımının uygulanmasıyla elde edilmektedir. Çalışma sonucunda, ATS algoritmasının hız özelliğini ve konum özelliğini aynı anda koruyarak gezingeleri etkili bir şekilde basitleştirebildiği gösterilmiştir (Lin ve diğ., 2017).

Liu ve diğ. (2016), Amnesik Sınırlı Çeyrek Sistem (ABQS - Amnesic Bounded Quadrant System) adı verilen çevrim içi bir gezinge verisi indirgeme yöntemi önermiştir. Yöntem, verilen depolama alanını akıllıca yönetmektedir ve gezinge verilerini yaşlarına bağlı olarak farklı hata toleranslarıyla sıkıştırmaktadır. Yapılan çalışmada; modern algoritmalara göre %45'e kadar daha başarılı indirgeme yapabildiği, çalışma süresinin %41'e kadar uzayabildiği ve yöntemin aşırı değişken gezinge şekillerine dayanıklı olduğunun gözlemlendiği belirtilmiştir (J. Liu ve diğ., 2016).

Deng ve diğ. (2017) çalışmasında, normalde çevirim dışı bir yöntem olan DPTS'i geliştirerek çevrim içi DPTS'i önermiştir. Ayrıca DPTS'in sıkıştırma süresini azaltmak için Bounded Quadrant System'i (BQS) kullanarak çevrim içi DPTS+ adını verdikleri çevrim içi DPTS'in de optimize edilmiş bir sürümünü önermişlerdir. Çalışma süresini azaltmak amacıyla geliştirilen yöntemler grafik işlem birimi (GPU) üzerinde koşulmuştur. Çalışma sonucunda, çevrim içi DPTS'in %21'e kadar daha yüksek indirgeme süresiyle çevrim dışı DPTS'den daha iyi performans gösterdiği, çevrim içi DPTS+ algoritmasının çalışma süresinin çevrim içi DPTS'den 3.95 kat daha hızlı olduğu gözlenmiştir (Deng ve diğ., 2017).

Samsonov ve Yakimova (2017), şematizm, düzgünlük ve düzenlilik özelliklerini kullanarak çeşitli çizgileri ayırt edebilen bir yöntem geliştirmiştir. Model ön işleme, işleme ve son işleme aşamalarından oluşmaktadır. Önişleme aşamasında farklı çizgilerin algılanmasına dayalı çizgi bölütlemesi yapılır. İşleme aşamasında oluşturulan segmentlere Li–Openshaw, Douglas–Peucker ve ortogonal sadeleştirme yöntemleri uygulanmaktadır. Son işleme safhasında ise basitleştirilmiş şeklin üzerine ek olarak düzenlilik eklenmesini sağlanmaktadır. Yöntemin birden fazla aşamalı olması nedeniyle çalışma süresinin daha yavaş olduğu, indirgeme oranının yüksek ve Haussdorf mesafesinin ise düşük olduğu gözlemlenmiştir (Samsonov & Yakimova, 2017).

Bao ve Du (2018), gezinge verilerinde aykırı değer tespiti için mesafeye dayalı bir yaklaşım önermiştir. Öncelikle, orijinal gezingelere ait desenler oluşturmak için Gürültülü Uygulamaların Yoğunluğa Dayalı Mekansal Kümelenmesi (DBSCAN) uygulanmıştır. Ardından, hesaplama maliyetini azaltmak için oluşturulan kümelerden durma noktaları tespit edilmiştir. Gezinge noktaları ve kümeler arasındaki benzerliği ölçmek için küme bağıl mesafesi ve küme açısal mesafesi kullanılmıştır. Yöntem, Çin QiongZhou Boğazı'ndan geçen gemilerin gerçek gezinge verileri üzerinde uygulanmıştır. Uygulama sonucunda, yöntemin mesafe ve hız/rota anomalisini etkili bir şekilde saptayabildiği ve hesaplama maliyetini büyük ölçüde azalttığı gözlemlenmiştir (Bao & Du, 2018).

Byon ve diğ. (2018), konum verilerini iki adımda işleyen bir gezinge verisi azaltma yöntemi önermiştir. 1. adımda kalma noktaları (sabit yerler, ev vb.), 2. adımda dönüş

noktaları tespit edilmektedir. Yöntem 563 veriden oluşan örneğe uygulanmış ve 5 kalma noktası ve 108 dönüş noktasına indirgenmiştir (Byon ve diğ., 2018).

Chen ve diğ. (2018), gezinge verisi depolama maliyetini düşürmeyi amaçlayan yeni bir Fırsatçı Sıkıştırma ve İletim yaklaşımı olan OCT'yi önermiştir. Yöntem hem geliştirilen terminal hem de bulut üzerinde hesaplama yapmaktadır. Ham veri, harita eşleştirme yöntemiyle yol ağı ile hizalanmaktadır. Fırsatçı aktarım metoduyla, yalnızca yol kimliği dizisi değiştiğinde veya öngörülen değeri ρ ile gerçek ρ değeri arasında büyük bir sapma olduğunda iletim yapılmaktadır. Deneysel veri toplama çalışmaları devam etmektedir (J. Chen ve diğ., 2018).

Kwon ve diğ. (2018), bir Ulaşım Yönetim Sistemi sunucusunda depolanan gezinge verilerinin boyutunu azaltmak için T-Azaltma adını verdikleri yeni bir mobil gezinge azaltma yöntemi önermiştir. İlk olarak, araçların ham konum verilerinin konum doğruluğunu iyileştirmek için bir yöntem geliştirilmiştir. İkinci olarak, sunucuya alınan konum güncelleme verilerini parçalayarak bir gezinge kümesini tanımlama yöntemi geliştirilmiştir. Kümeleme yöntemi ile veri sıkıştırılması sağlanmıştır. Önerilen yöntem ortalama gezinge hatasını önemli ölçüde azaltmıştır ve veriyi %88'e kadar indirgemiştir (Kwon ve diğ., 2018).

Chen ve diğ. (2019), mobil ortamda çalışan gürültülü ve seyrek GPS noktalarını yollara hizalayan Spatial-Directional Matching (SD-Matching) Algoritması ile birlikte çalışarak kavşaklardaki pozisyon değişikliğinin tespitini sağlamaya çalışan bir çevrim içi gezinge indirgeme yöntemi sunmuştur. Çerçeve iki aşamadan oluşmaktadır, çevrim içi gezinge haritalama ve hareket indirgemesi. Çalışma sonuçlarına göre, SD-Matching algoritması mevcut yöntemlere göre daha yüksek bir ortalama doğruluk sunmaktadır. İndirgeme algoritması ise daha az zaman harcayarak daha yüksek performans göstermektedir (C. Chen ve diğ., 2019).

Hao ve diğ. (2019), hem konum bilgilerini hem de yön bilgilerini dikkate alarak istikameti ve konumları koruyan bir algoritma önermiştir. Algoritmanın çalışması için bir eşik açısına ihtiyaç duyulmaktadır. Bu eşiğin uygun belirlenmesi sayesinde algoritma, hareketleri geleneksel gezinge sıkıştırma algoritmalarından daha doğru, etkin ve esnek bir şekilde indirgeyebilmektedir (Hao ve diğ., 2019).

Xu ve diğ. (2019), kapsamlı araç gezinge verilerinden madencilik yoluyla düğüm önemini sıralamak için Başlangıç-Hedef Akışı Entropisi (ODEF - Origin-Destination Entropy With Flow) algoritması ve Kavşak Sırası (CRRank - Crossroad-Rank) algoritmasının birleştirilmesi ile yeni bir yöntem önermiştir. ODEF, bir düğümün merkeziyetini değerlendirmek ve merkeziyetini trafik akışı ile bütünleştirerek önem derecesini hesaplamak için bilgi kazancı temelli çalışır. CRRank, OD çifti, yol ve kesişim arasındaki karşılıklı pekiştirici ilişkileri yakalayan özvektör merkeziyetini temel alır. ODEF'te göz önünde bulundurulan faktörlere ek olarak CRRank bir düğümün yeri doldurulamazlığını ve komşu düğümler arasındaki uzamsal ilişkileri de dikkate alır. Geliştirilen yöntemin başarımı 30.000'den fazla taksinin verisinin bulunduğu veri kümesi üzerinde test edilmiştir ve başarılı bulunmuştur (Xu ve diğ., 2019).

Zhang ve diğ. (2019), yol genişliği (W), araç sürüş yönü (D) ve dönüş açısı (A) birleşimini kullanarak WDA adını verdikleri bir gezinge indirgeme algoritması önermiştir. Bu yöntem, araç hareket noktalarının W ve D'ye göre izin verilen bir hata aralığında doğrusal olup olmadığını belirlemektedir ve daha sonra bir aracın A'ya göre önemli ölçüde döndüğü noktaları belirlemektedir. Algoritma, araçların gezingeleri üzerindeki yol şekillerinin kısıtlamalarını dikkate almaktadır. Böylece hareketin, dönen bir ize işaret eden yaklaşık düz bir çizgi mi yoksa bir eğri mi oluşturduğunu tespit edebilmektedir. Basitleştirilmiş ve orijinal yörüngeler arasında yüksek verimli indirgeme ve yüksek benzerlik sağlamaktadır. Bunun yanısıra, algoritma D ile de ilgili yön bilgisini korumaktadır. Çalışmada Microsoft GeoLife veri kümesi kullanılarak yapılan testte, önerilen algoritmanın araç gezinge verileri için ortalama %4.03 sıkıştırma oranına ulaştığını ve ortalama indirgeme süresinin her bir veri dosyası için yaklaşık 1.15 dakika olduğu gözlemlenmiştir (H. Zhang ve diğ., 2019).

Liu ve diğ. (2020), D-P algoritmasına, monotonik (verilen seti koruyan, sıralı) zincirlere ve ikiliğe dayanan yeni bir çizgi basitleştirme algoritması önermiştir. Yöntem ilk olarak, orijinal çizgileri basitleştirmek için geleneksel D-P algoritması kullanmaktadır. Daha sonra indirgenmiş çizgiler birkaç monoton zincire bölünmektedir. Ardından dikotomi (ikili, ikiye bölünen), monotonik zincirlerin kesişme konumlarını etkili bir şekilde araştırmak için kullanılmaktadır ve kesişen monotonik zincirler işlenmektedir. Bu sayede yöntem kendi kesişme problemlerini çözmektedir. D-P algoritması ve Yıldız şekilli

algoritma ile karşılaştırıldığında algoritmik verimlilik ve doğrulukta avantajlara sahip olduğunu göstermektedir. Yöntem, O(nm) + O(m log n) karmaşıklıkta olmasına rağmen test verilerinde D-P algoritmasına göre %23 daha yavaş, ortalama vektör yer değiştirme oranı olarak %3 daha iyi ve Haussdorf mesafesinin ise %2.7 daha iyi olduğu gözlemlenmiştir (B. Liu ve diğ., 2020).

Zhao ve diğ. (2023), traktör ve birden fazla römorklu araç için gerçek zamanlı hareket planlaması yapabilmek amacıyla maksimum sonsal tahmin problemini çözebilmek amacıyla bir yöntem önermiştir. Planlanan gezinge ve gerçekleşen gezinge arasındaki farkın tespit edilmesi ve yolun engelsiz olacak biçimde planlanması için bir gezinge indirgemesi yöntemi önermiştir. Gezingeye ait dinamikler göz önünde bulundurularak denetleyici tabanlı bir indirgeme yöntemi önerilmiştir. Bu sayede orijinal ve indirgenmiş gezinge arasındaki uyumsuzluğun azaltılması amaçlanmıştır. Yöntemin sonucunda oluşturulan gezingeler verimli bir şekilde, engelsiz ve dinamik olarak oluşturulmuştur. Yöntemin performansı gerçek hayattaki endüstriyel çekici aracı üzerinde denenmiştir (H. Zhao ve diğ., 2023).

Chen vd.(2005) tarafından yapılmış olan literatürde yer alan bazı yöntemlerin kıyaslaması ve bazı farklı yöntemler ile ilgili yöntemlerin çevrim içi çalışma durumunun eklenmiş hali Tablo 2.1‘de gösterilmektedir (L. Chen ve diğ., 2005).

Tablo 2.1. Literatürde var olan indirgeme algoritmalarının karşılaştırılması (L. Chen ve diğ., 2005).

İndirgeme Yöntemi	**Çevrim içi**	**Görsel fark**	**Belirli durumda yer değiştirme**	**Belirsiz durumda yer değiştirme**	**Belirli durumda şekilsel bozulma**	**Belirsiz durumda şekilsel bozulma**	**Düşük hesaplama süresi**
Douglas–Peucker	-	***	***	***	***	***	*
Zhao–Saalfield	-	***	**	**	**	*	***
Reumann–Witkam	-	***	**	**	**	**	*
Visvalingham–Whyatt	-	***	**	**	**	*	*
Noktalar arası mesafe	+	***	*	*	*	*	***
Opheim	-	***	*	*	*	*	*
Lang	-	***	**	**	**	*	**
Dikey mesafe	-	**	*	*	*	*	***
n’inci nokta	+	*	*	*	*	*	***
T-Reduce	-	***	***	**	***	**	**
SD-Matching	+	***	**	**	**	**	***
ODEF-CRRank	-	**	**	**	***	**	*
WDA	-	***	***	**	***	**	*
Samsonov-Yakimova	-	*	**	**	**	**	*
Diktonik D-P	-	***	***	***	**	**	**

2.4.GNSS Veri Kümeleri ve Dosya Biçimleri

GNSS'lerden elde edilen veriler çeşitli formlarda ve farklı dosya türlerinde saklanabilmektedir. Yaygın olarak kullanılan pek çok dosya türü bulunmakla birlikte, genel kabul görmüş tek bir format bulunmamaktadır. Her dosya türünün veriyi barındırma biçimi ve işleme şekli farklıdır.

2.4.1.GNSS Veri Kümeleri

Farklı amaçlarla, hareketliliklere ait GNSS verileri çeşitli veri kümelerinde saklanmaktadır. GNNS verilerinin sınıflama, indirgeme vb. uygulamalarını ve geliştirilen modelleri sınamak için birçok veri kümesi bulunmaktadır (Bian ve diğ., 2019; URL-10; URL-11; URL-12). Bu veri kümeleri yürüyen insan hareketleri, hayvan hareketleri, uçan nesnelerin hareketleri, kara ve deniz taşıtları gibi pek çok farklı davranışı içermektedir. Bazıları birden fazla harekete ait veri barındırırken bazıları yalnızca tek bir harekete odaklanmaktadır. Örneğin, göçmen hayvanlara ait göç verileri Movebank isimli veri kümesinde, fırtınalara ait hareket verileri NHC arşivindeki veri kümesiyle paylaşılmaktadır. Aşağıda listelenen veri kümeleri, içeriğinde barındırdığı veri miktarı ve çeşitliliği bakımından zengin veri kümeleridir. Bunlar:

1. UCF,
2. Microsoft GeoLife,
3. Movebank,
4. NHC Data Archive,
5. FlightRadar24

veri kümeleridir.

2.4.1.1. University of Central Florida (UCF) Veri Kümesi

UCF (University of Central Florida) veri kümei, UCF tarafından toplanan ve insanların yapılan aktivitelerini içeren bir veri kümesidir. Yaklaşık 10'dan fazla veri kümesinde, 50 farklı aktivite kategorisinde, 13.000'den fazla videoda insanların hareketlerini takip eden veriler bulunmaktadır. UCF veri kümesi, insanların yapılan aktivitelerin tanınması ve anlaşılması, video analitik, görüntü işleme ve benzeri alanlarda çalışmalar yapılması için

faydalı bir kaynaktır. UCF50 ve UCF Sport gezinge sınıflamasına dahil olan ana veri kümeleridir (URL-13).

2.4.1.2. Microsoft GeoLife Veri Kümesi

Microsoft Research Asia tarafından 2007 yılından 2011 yılına kadar, yaklaşık 178 kişiye ait yürüme, otomobil, tren vb. seyahatlerine ait toplam 1.251.654 kilometre mesafede ve 48.203 saat sürede, 17.621 hareket verisinin toplandığı GeoLife veri kümesi yaygın olarak kullanılmaktadır. Veri kümesi oluşturulurken farklı GPS kaydedicileri ile çeşitli örnekleme hızlarında kayıt yapılmıştır. Hareketlerin yaklaşık %91'i 1-5 saniye ya da 5-10 metre aralıklarla yoğun bir biçimde elde edilmiştir (Zheng ve diğ., 2008, 2009, 2010). Microsoft GeoLife veri kümesinde ".plt" uzantılı dosyalar kullanılmaktadır. GeoLife veri kümesine ait örnek bir GPS verisi aşağıdaki Tablo 2.2'deki gibidir. Tablo 2.2 veri parçalarının karşılıkları gösterilmektedir.

Tablo 2.2. ".plt" uzantılı dosya formatı örneği

Enlem (Derece)	39.984094
Boylam (Derece)	116.319236
Boş	0
Yükseklik (Ayak)	492
Zaman Etiketi (30/12/1899'den beri)	39744.2451967593
Tarih	23.10.2008
Saat	05:53:05

2.4.1.3. Movebank Veri Kümesi

Movebank, Ulusal Bilim Vakfı, Alman Havacılık ve Uzay Merkezi, Alman Bilim Vakfı ve NASA'nın desteğiyle geliştirilmiştir. Movebank'a, Max Planck Hayvan Davranışları Enstitüsü tarafından North Carolina Doğa Bilimleri Müzesi, Ohio Eyalet Üniversitesi ve Konstanz Üniversitesi koordineli olarak ev sahipliği yapmaktadır. Movebank, hayvan hareketi ve biyolojik kayıt verileri için küresel bir açık ve ücretsiz arşiv görevi görmektedir. Mart 2020 itibarıyla 2,4 milyar yer, 3,2 milyar yer dışı olay, 7,662 çalışma, 989 takson bulunmaktadır (URL-12).

2.4.1.4. Ulusal Fırtına Merkezi (NHC) Veri Arşivi

Ulusal Fırtına Merkezi (NHC) Veri Arşivi, Ulusal Okyanus ve Atmosfer Dairesi (National Oceanic and Atmospheric Administration) tarafından işletilen NHC'nin depoladığı ve fırtınalarla ilgili verileri içeren bir veri arşividir. Veri arşivi, fırtınalarla ilgili olarak toplanan verileri toplamaktadır. Elde ettiği bu verileri araştırmacılar, kuruluşlar ve diğer ilgili tarafların kullanımına sunmaktadır. NHC Veri Arşivi, fırtınaların yapısını, rotalarını, şiddetini ve etkilerini anlamaya yardımcı olmak için faydalı bir kaynaktır (URL-11).

2.4.1.5. FlightRadar24 Veri Arşivi

FlightRadar24, Dünya çapında hizmet sunan bir uçuş takip sistemidir. 2006 yılında iki İsveçli havacılık meraklısının Kuzey ve Orta Avrupa'da bir Otomatik Bağımlı Gözetim-Yayını (ADS-B) alıcı ağı kurmaya karar vermesiyle bir hobi projesi olarak başlamıştır. ADS-B, bir uçağa ait uydu navigasyonu ve/veya farklı sensörler kullanılarak konumunun belirlenmesini ve belirlenen konumun periyodik olarak yayınladığı sinyaller ile izlenmesini sağlayan gözetim teknolojisidir (URL-14). 2023 yılı itibariyle, 190 ülkede günlük 180.000'den fazla uçuşu takip eden 20.000'den fazla alıcıya sahiptir. Elde edilen veriler dünya çapında herhangi bir ticarî uçakta görsel güncellemeler sağlamak için gerçek zamanlı olarak bir harita üzerinde görselleştirilmektedir. Ayrıca, üyelik oluşturularak sistemin sağladı ek özellikler ve uçuşlar ile ilgili veriler kullanılabilmektedir (URL-15).

2.4.2.GNSS Verisi Dosya Biçimleri

Günümüzde en yaygın kullanıma sahip olan GNSS'lerin başında GPS gelmektedir. Hem CBS'lerde hem de harita uygulamalarında sıklıkla kullanılan GPS verilerine ait dosya biçimleri:

1. .kml (Keyhole Markup Language),
2. .kmz (Keyhole Markup Language Zipped),
3. .gpx (GPS eXchange Format),
4. .plt,
5. .loc,

6. .dat,
7. .log,
8. .csv,
9. .gpl (GPSLogger Binary File)

şeklinde sıralanabilir (URL-16; URL-17).

2.5.Mesafe ve Hata Ölçüm Yöntemleri

Gezinge verileri ile çalışıldığında, özellikle de indirgeme konusundaki çalışmalarda, mesafe ve hata ölçümleri yapılması gerekebilir. İndirgenen verinin orijinal veriden ne kadar farklı olduğunun tespiti sırasında mesafe ve hata ölçüm yöntemlerine gereksinim duyulmaktadır. Literatürde basit gezingelerin orijinal gezingelerden sapmasını belirlemek için çeşitli mesafe ölçüm yöntemleri bulunmaktadır.

2.5.1.Mesafe Ölçüm Yöntemleri

Literatürde yapılan gezinge indirgeme çalışmalarında indirgemenin yapılması sırasında genellikle iki konum arasındaki uzaklığa ihtiyaç duyulmaktadır. Bunun yanı sıra yine hatanın ölçülmesi için gereken hesaplamalar sırasında da indirgenmiş gezingenin orijinal gezingeden ne kadar farklı olduğunun tespitinin yapılması amacıyla da bu yöntemlere gereksinim duyulmaktadır. Literatürde indirgenmiş gezingelerin orijinal gezingelerden sapmasını belirlemek için çeşitli mesafe ölçüm yöntemleri bulunmaktadır. Bunlardan en yaygın kullanılanları;

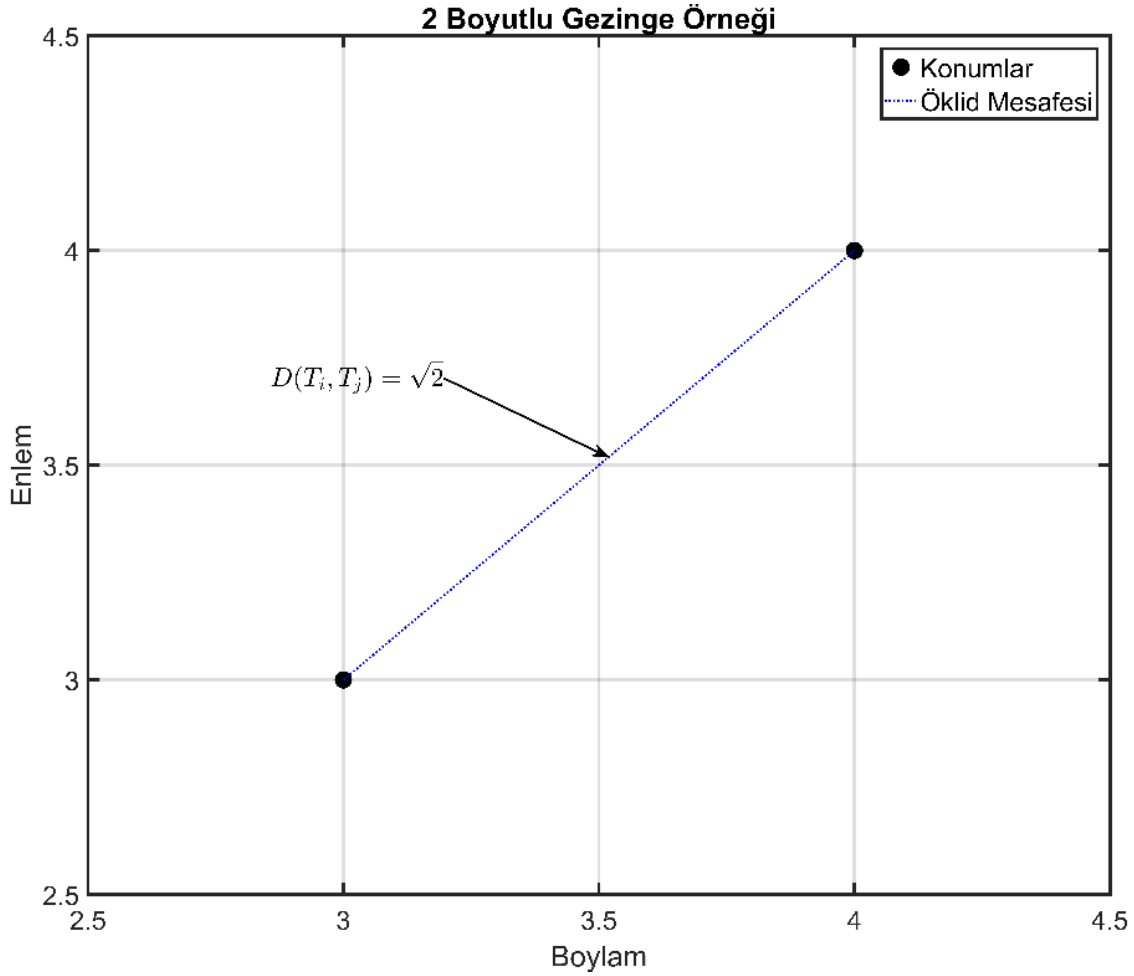
1. Öklid Mesafesi
2. Manhattan Mesafesi
3. Hausdorff Mesafesi
4. Frechet Mesafesidir.

2.5.1.1. Öklid Mesafesi

Öklid mesafesi literatürde uzaklık, benzerlik ve kümeleme gibi alanlarda en yaygın kullanılan yöntemlerdendir. İki nokta arasındaki doğrusal en kısa mesafedir. Denklem (2.1)'de gösterilen formül ile hesaplanmaktadır (L. Chen ve diğ., 2005; M. Chen ve diğ., 2012; Meng ve diğ., 2017; Vlachos ve diğ., 2002). Örnek olarak seçilen (3, 3) ve (4, 4)

noktalarının Öklid mesafesi Şekil 2.4'te gösterilmektedir. Burada: N: nokta sayısnı, x_n^i : T_i gezingesine ait n. noktanın x koordinatını, x_n^j : T_j gezingesine ait n. noktanın x koordinatını, y_n^i : T_i gezingesine ait n. noktanın y koordinatını, y_n^j : T_j gezingesine ait n. noktanın y koordinatını göstermektedir.

$$D(T_i, T_j) = \frac{1}{N}\sum_N \sqrt{\left(x_n^i - x_n^j\right)^2 + \left(y_n^i - y_n^j\right)^2} \tag{2.1}$$



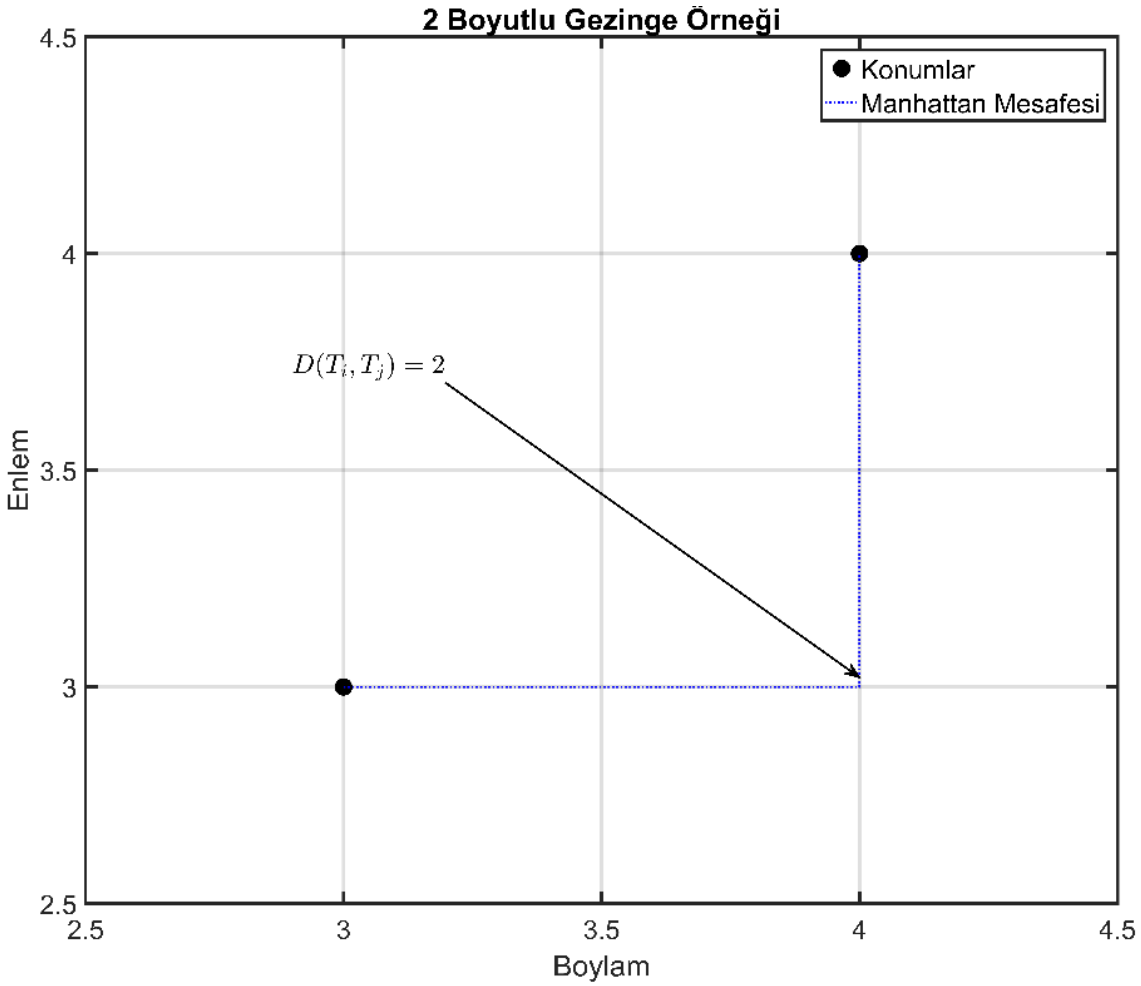
Şekil 2.4. Öklid mesafesi örneği

2.5.1.2. Manhattan Mesafesi

Manhattan mesafesi literatürde Öklid mesafesinden sonra kümeleme, benzerlik ve veri madenciliği konularında yaygın olarak kullanılan yöntemlerden birisidir. Noktalar arasında mutlak uzaklıkların yani yatay ve dikey doğruların uzunluklarının toplamı şeklinde hesaplanmaktadır. Denklem (2.2)'de gösterilen formül ile hesaplanmaktadır

(Bekhet & Ahmed, 2020). Örnek olarak seçilen (3, 3) ve (4, 4) noktalarının Manhattan mesafesi Şekil 2.5'te gösterilmektedir.

$$D(T_i,T_j)=\frac{1}{N}\sum_{N}|x_n^i-x_n^j|+|y_n^i-y_n^j| \tag{2.2}$$

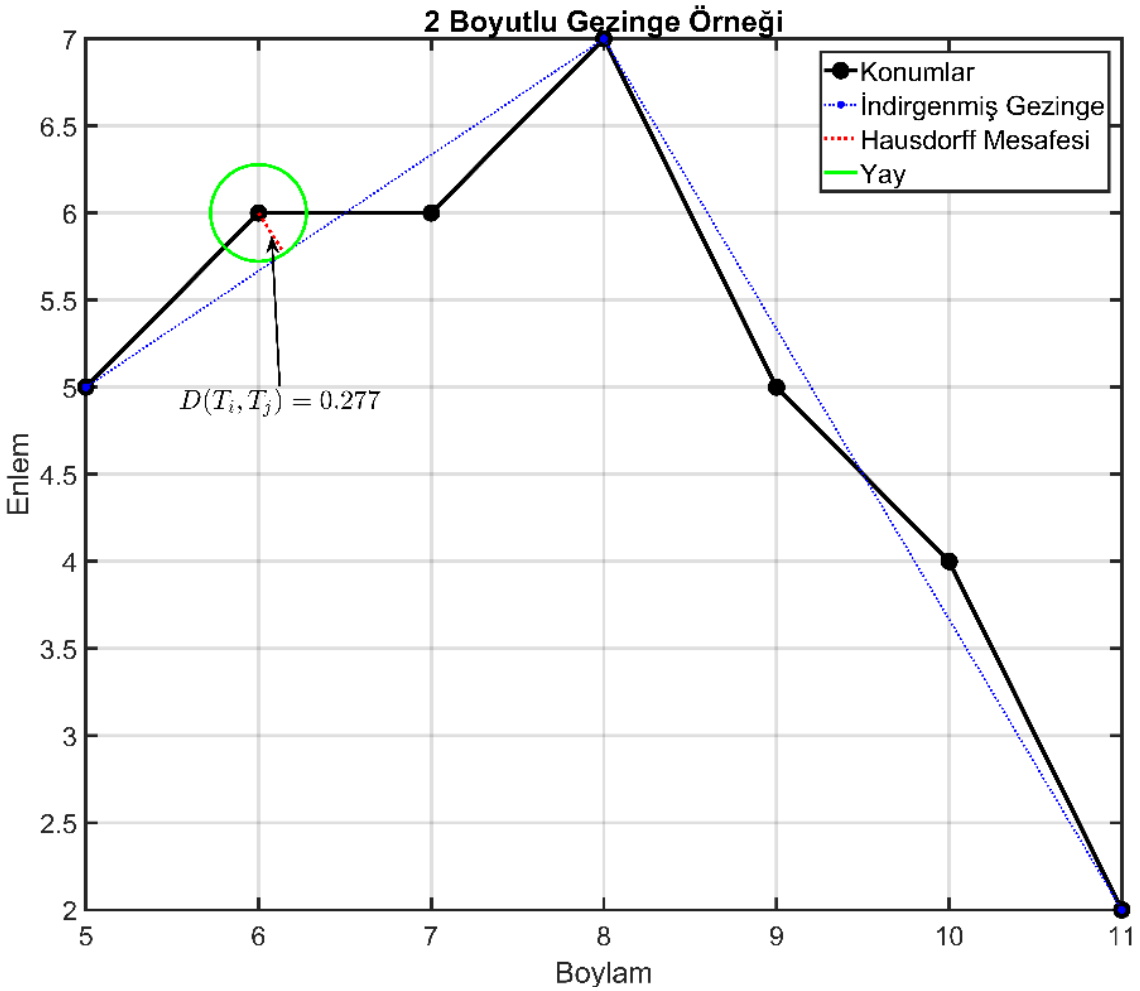


Şekil 2.5. Manhattan mesafesi örneği

2.5.1.3. Haussdorff Mesafesi

Haussdorff mesafesi literatürde; benzerlik tespiti gibi konularda yaygın olarak kullanılan yöntemlerden birisidir. Bir gezingenin her noktasının diğer bir gezingenin bazı noktalarına ne kadar yakın olduğunun benzerliklerini ölçmektedir. *T* ve *T`* gezingeleri için *T* ve *T`*'deki uzunlukları birleştirmeden ölçmektedir. Denklem (2.3)'de gösterilen formül ile hesaplanmaktadır (Abam ve diğ., 2014). Örnek olarak seçilen noktaların Hausdorff mesafesi Şekil 2.6'da gösterilmektedir.

$$D(T_i,T_j)=\max\{d(T_i,T_j),d(T_j,T_i)\} \quad (2.3)$$



Şekil 2.6. Hausdorff mesafesi örneği

2.5.1.4. Frechet Mesafesi

Frechet mesafesi literatürde; coğrafi bilgi sistemleri, hareket planlaması, görüntü işleme ve biyoinformatik gibi alanlarda yaygın olarak kullanılan yöntemlerden birisidir. Frechet mesafesi, iki eğri arasındaki benzerlik veya farklılık derecesini ölçmektedir. *T* ve *T`* gezingeleri arasındaki minimum uzaklığı hesaplamaktadır. Başka bir deyişle *T* ve *T`*'nin orijinden itibaren birbirleriyle eşleştirildiği varsayılan parametrik bir eşleştirme yolu oluşturarak, maksimum değeri hesaplar. Bu maksimum değer, *T* ve *T`* arasındaki en yakın mesafedir ve Frechet mesafesi olarak adlandırılmaktadır. Denklem (2.4)'te gösterilen formül ile hesaplanmaktadır (Abam ve diğ., 2014).

$$D(T_i, T_j) = \inf_{\alpha,\beta} \max_{t\in[0,1]} \{d\left(T_i(\alpha(t)), T_j(\beta(t))\right)\} \quad (2.4)$$

Burada d; S, α, β'nın uzaklık fonksiyonudur ve süreklidir.

Literatürde kullanılan mesafe ölçüm yöntemlerine ait özellikler Tablo 2.3'de gösterilmektedir.

Tablo 2.3. Mesafe Ölçüm Yöntemleri Kıyaslaması (Bian ve diğ., 2019)

Mesafe	**Uzunluk Birleştirme**	**Hesaplama Karmaşıklığı**
Öklid Mesafesi	Evet	O(n)
Manhattan Mesafesi	Evet	O(n)
Hausdorff Mesafesi	Hayır	O(mn)
Frechet Mesafesi	Hayır	O(mn)

2.5.2.Hata Ölçüm Yöntemleri

Literatürde yapılan gezinge indirgeme çalışmaların sonucunda elde edilen indirgenmiş gezingenin orijinal gezingeden ne kadar farklı olduğunun hesaplanması ile meydana gelen hatanın tespit edilmesine ihtiyaç duyulmaktadır. Elde edilen hata değeri, indirgeme yönteminin performansının değerlendirilmesinde kullanılmaktadır. Literatürde indirgenmiş gezingelerin orijinal gezingelerden sapma miktarını belirlemek için çeşitli hata ölçüm yöntemleri bulunmaktadır. Bunlardan en yaygın kullanılanları:

1. Ortalama Mutlak Hata
2. Ortalama Karesel Hata
3. Kök Ortalama Kare Hatası.

2.5.2.1. Ortalama Mutlak Hata

Ortalama mutlak hata (MAE – ing: mean absolute error), indirgenmiş gezinge ile orijinal gezingenin birbirine ne kadar yakın veya benzer olduğunu bulmayı sağlamaktadır. Bu yöntem benzerlik açısından kabaca bir bilgi vermektedir. MAE, Denklem (2.5)'de gösterilen formül ile hesaplanmaktadır. İndirgenen gezinge, orijinal gezingeye ne kadar benzerse MAE o kadar küçük olur; aksi durumda ise indirgenmiş gezinge, orijinal gezingeden ne kadar farklıysa MAE o kadar büyük olur. Literatürde pek yaygın

kullanılmamaktadır. Burada; x_n^i ve y_n^iorijinal gezingenin boylam ve enlem bilgilerini, x_n^j ve y_n^j ise indirgenmiş gezingenin boylam ve enlem bilgilerini temsil etmektedir.

$$\text{MAE} = \frac{1}{N}\sum_{N} | x_n^i - x_n^j | + | y_n^i - y_n^j | \tag{2.5}$$

2.5.2.2. Ortalama Karesel Hata

Ortalama karesel hata (MSE – ing: mean square error), indirgenmiş gezinge ile orijinal gezingenin birbirine ne kadar yakın veya benzer olduğunu bulmayı sağlamaktadır. MSE, Denklem (2.6)'da gösterilen formül ile hesaplanmaktadır. İndirgenen gezinge, orijinal gezingeye ne kadar benzerse MSE o kadar küçük olur; aksi durumda ise indirgenmiş gezinge, orijinal gezingeden ne kadar farklıysa MSE o kadar büyük olur. Her bir noktanın orijinal ve indirgenmiş gezingeler için sapma miktarının karesinin alınması, MSE'nin sıfırdan büyük veya eşit olmasını sağlamaktadır. Tüm noktalardaki hataların toplamı alınmadan önce her bir noktadaki hatanın karesi alındığından büyük miktardaki farklılıkların olduğu noktaların toplam hata üzerindeki etkisi daha benzer noktaların etkisinden çok daha fazla olmaktadır. Burada; x_n^i ve y_n^iorijinal gezingenin boylam ve enlem bilgilerini, x_n^j ve y_n^j ise indirgenmiş gezingenin boylam ve enlem bilgilerini temsil etmektedir.

$$\text{MSE} = \frac{1}{N}\sum_{N} \left(\left(x_n^i - x_n^j\right) + \left(y_n^i - y_n^j\right) \right)^2 \tag{2.6}$$

2.5.2.3. Kök Ortalama Kare Hatası

Kök ortalama kare hatası (RMSE – ing: root mean square error), indirgenmiş gezinge ile orijinal gezingenin birbirine ne kadar yakın veya benzer olduğunu bulmayı sağlamaktadır. RMSE, Denklem (2.7)'de gösterilen formül ile hesaplanmaktadır. İndirgenen gezinge, orijinal gezingeye ne kadar benzerse RMSE o kadar küçük olur; aksi durumda ise indirgenmiş gezinge, orijinal gezingeden ne kadar farklıysa RMSE o kadar büyük olur. MSE, farkın ne kadar olduğunu gösterir, RMSE ise bu farklılıkların ne kadar yayılmış olduğunu gösterir. RMSE gezinge indirgemesi konusunda altmışlık sayı sisteminde (ing: sexagesimal) değer vereceğinden indirgeme hatası ölçümünde

kullanılması daha uygun olmaktadır. Burada; x_n^i ve y_n^iorijinal gezingenin boylam ve enlem bilgilerini, x_n^j ve y_n^j ise indirgenmiş gezingenin boylam ve enlem bilgilerini temsil etmektedir.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{N}\left(\left(x_n^i - x_n^j\right) + \left(y_n^i - y_n^j\right)\right)^2} \tag{2.7}$$

3. GELİŞTİRİLEN İNDİRGEME YÖNTEMİ

Tez çalışmasında, günümüz koşullarında birikimli biçimde saklanmaya devam edilen gezinge verilerinin işlenmesi ve saklanmasını kolaylaştırmak amacıyla gezinge verisi indirgemesi yapan yeni bir yöntem geliştirilmiştir. Bu yöntem, literatürden derlenmiş mevcut çalışmaların kısıtları da göz önüne alındığında, aşağıdaki sorunlara da çözüm üretebilmektedir. Geliştirilen yöntem:

1. var olan yöntemlerden daha verimli,
2. yoğun gürültülü koşullar altında çalışabilen,
3. algoritmanın doğruluğu aşırı değişken gezinge şekillerine dayanıklı,
4. seçilen noktaların önem seviyesini hesaplayan,
5. bir düğümün yeri doldurulamazlığını dikkate alan,
6. komşu düğümler arasındaki uzamsal ilişkileri doğru tespit eden,
7. daha düşük hesaplama karmaşıklığına sahip,
8. yüksek indirgeme oranına sahip

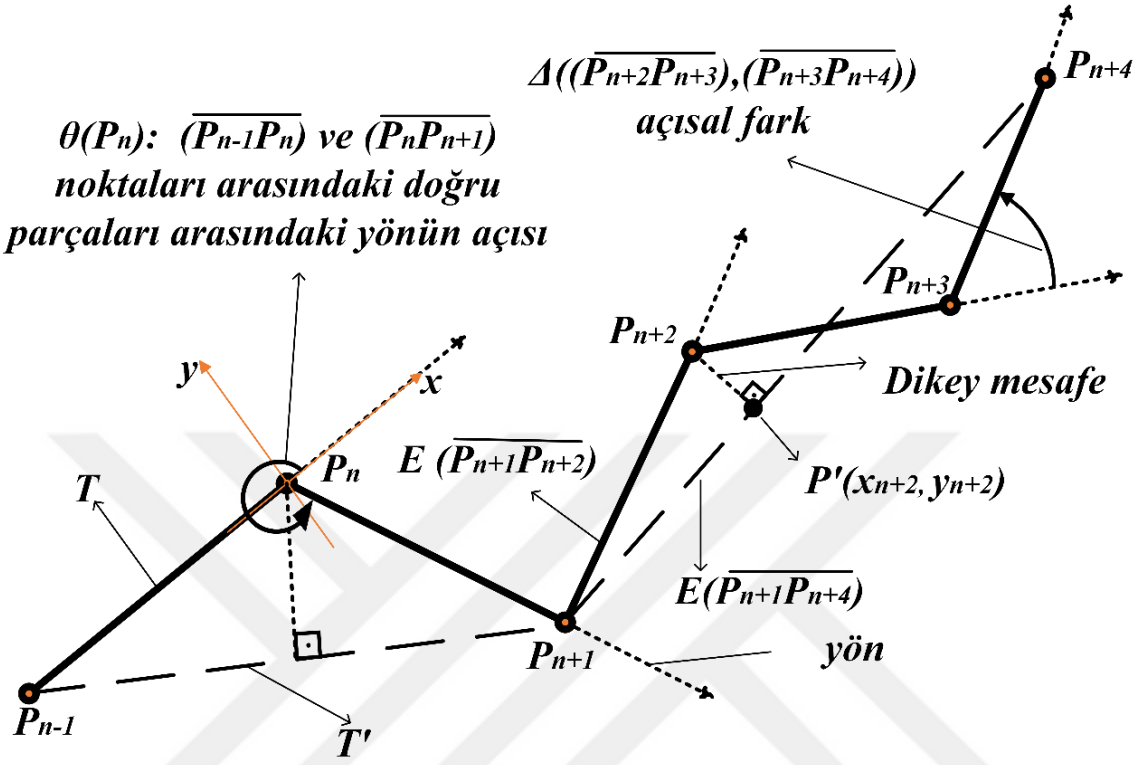
bir yöntemdir.

Yukarıda bahsedilen kısıtların yanı sıra, mobil cihazların; depolama alanı, batarya ve işlem gücü gibi kısıtları bulunduğundan, literatürde sunulmuş olan özellikle çevrim dışı algoritmaların verimli sonuçlar üretmeyeceği düşünülmektedir. Geliştirilmiş olan bu yöntem yalnızca var olan veri kümeleri üzerinde ve bilgisayarlarda değil mobil cihazlar üzerinde de çevrim içi çalışabilmektedir. Bu tezde önerilen matematiksel indirgeme yöntemi kullanılarak oluşturulan veri kümesi yardımıyla, bir yapay sinir ağı modeli eğitimi gerçekleştirilmiştir. Eğitim sonucunda elde edilen model kullanılarak mobil cihazdaki uygulamanın gerçek zamanlı indirgeme yapması sağlanmıştır.

3.1. Trigonometrik Tanımlar

Bu tez kapsamında geliştirilmiş olan yöntem, McMaster tarafından önerilen çizgi sadeleştirme sınıflarından biri olan “Kısıtlanmamış Genişletilmiş Yerel İşleme” (ing: Unconstrained Extended Local Processing) olarak sınıflandırılmaktadır (McMaster, 1987). “Kısıtlanmamış Genişletilmiş Yerel İşleme” yöntemleri belirli bir hattın içerisinde bulunan noktalar arasındaki açısal ve uzaklık farklarını kullanarak çalışmaktadır. Bu

nedenle, yöntemin çalışması sırasında gerekli terimlerin tanımlanmasına ihtiyaç duyulmaktadır. Bu terimler Şekil 3.1'de gösterilmektedir.



Şekil 3.1. Trigonometrik ifadeler ve tanımları

Burada; *T* orijinal gezingeyi, *T`* orijinal gezingenin indirgenmiş hâlini, $P_n(x, y)$ *n* zaman etiketine sahip olan ve *(x, y)* koordinatlarındaki bir noktayı, $E(\overline{P_{n+1}P_{n+2}})$ P_{n+1} ve P_{n+2} noktalarının oluşturduğu kenarı, $E(\overline{P_{n+1}P_{n+4}})$ indirgenmiş gezinge üzerindeki P_{n+1} ve P_{n+4} noktalarının meydana getirdiği kenarı, $\theta(P_n)$ $(\overline{P_{n-1}P_n})$ ve $(\overline{P_nP_{n+1}})$ doğru parçaları arasındaki yönelim açısını, $\Delta(P_n)$ $(\overline{P_{n+2}P_{n+3}})$ ve $(\overline{P_{n+3}P_{n+4}})$ çizgilerinin yönelimlere ait açısal farkı, $P`(x_{n+2}, y_{n+2})$ P_{n+2} noktasının indirgenmiş gezinge üzerindeki dikey karşılığını temsil etmektedir. İki noktanın birbirine göre açısı için Denklem (3.1)'de gösterilen formül kullanılmaktadır.

$$\theta = \tan^{-1}\left(\frac{x}{y}\right) * \frac{180}{\pi}; \{x,y \mid x = x_n\text{-}x_{n\text{-}i};\ y = y_n\text{-}y_{n\text{-}i}\} \quad (3.1)$$

Burada θ, düzlemdeki bir noktanın açısal değerini, *x* ve *y* ise düzlemdeki her bir noktanın enlem ve boylam bilgisini göstermektedir. x_{n-i} ve y_{n-i} değişkenleri pencerelemede referans alınarak hesaplamada kullanılacak hücrenin boylam ve enlem bilgilerini, x_n ve y_n ise

açısal konumu buna göre hesaplanacak olan referans konumun boylam ve enlem bilgilerini temsil etmektedir.

3.2. İndirgeme Yöntemi

Geliştirilmiş olan matematiksel yöntemin nasıl çalıştığı aşağıda anlatılmaktadır. Yöntemin akışına ait sözde kod ise Tablo 3.1'de gösterilmektedir.

Yöntem çalışmaya başlamadan önce bazı parametrelerin tanımlarının yapılmasına ihtiyaç duymaktadır. Ön tanım yapılması gereken parametreler, kullanılacak olan eşik değeri, pencereleme boyutu ve ortalama yer değiştirme miktarıdır. Her üç parametre de yöntemin çalışması sonucunda elde edilecek olan hassasiyetin değişmesine neden olacağından doğru bir biçimde belirlenmesi önemlidir. Pencereleme boyutunun çok küçük olması ya da çok büyük olması karar verme aşamasında kontrol sayısını etkilemektedir. Aşırı büyük bir pencereleme boyutu, çok fazla referans noktası gerektirecek, bu da düşük bir indirgeme oranı ve orijinal gezingeye daha yüksek bir benzerlik ile sonuçlanacaktır. Küçük pencereleme boyutu, kontrol edilecek nokta sayısını azalttığından yöntemin hızlı çalışmasına sağlayabilir ancak benzerliğin de aşırı biçimde azalmasına neden olabilir. Yine, ortalama yer değiştirme miktarı karşılaşılabilecek olan konum değeri okuma problemlerinin önüne geçilmesi amacıyla kullanılmaktadır. Ayrıca, eşik değerinin çok küçük olması da indirgenen nokta sayısının azalmasına yani orijinal gezingeye daha benzer bir gezinge elde edilmesine bu nedenle de daha düşük bir indirgeme oranına neden olacaktır.

Yöntemde kullanılan açısal eşik değerlerinin sayısı pencereleme boyutuna göre değişmektedir. Açısal eşik değerleri, ilk hücreye atanan ilk eşik değerinden başlayarak, pencerenin her hücresi için ikiye bölünerek atanır. Bu nedenle, başlangıç aşamasında tek bir değerin belirlenmesi yeterlidir. Her bir pencere hücresi için farklı eşik değeri kullanılması hassasiyeti artırmak için yapılmaktadır ve her bir eşik değerinin hesaplama yöntemi Denklem (3.2) ile gösterilmektedir.

$$\mu_i = \frac{\mu_0}{2 * (i - 1)} \tag{3.2}$$

Denklem (3.2)'de μ_i, o anda indirgemesine karar verilecek olan pencere hücresi için kullanılacak eşik değerini temsil etmektedir. Öte yandan μ_0, *P1*'in açısal konumuna göre karar vermek için kullanılmaktadır ve başlangıç durumunda pencerelemenin boyutu dikkate alınarak belirlenmektedir. *i*, karar verilecek pencere hücresinin indeks değerini göstermektedir. Örneğin; 4 hücreli bir pencereleme için, P_1 için açısal eşik değeri $\mu_0 = \pi$ olarak belirlenirse, P_2 için $\mu_2 = \pi * \frac{1}{2*(2-1)} = \pi * \frac{1}{2} = \frac{\pi}{2}$, P_3 için $\mu_3 = \pi * \frac{1}{2*(3-1)} = \pi * \frac{1}{4} = \frac{\pi}{4}$ olarak belirlenir.

Tablo 3.1. Önerilen yöntemin sözde kodu

Algoritma 1. Önerilen yöntemin sözde kodu

```
Giriş: p = pencereleme boyutu, μ0 = eşik açısı, ε = ortalama yer değiştirme miktarı
Çıkış: indirgenmiş konum kümesi
while (mevcutKonum = oku(gnssAlıcısı))
    while (kontrolNoktası = pencere.pop())
        Θi =açıHesapla(mevcutKonum, kontrolNoktası); //Denklem (3.1)
        μi = eşikHesapla(μ0, i); //Denklem (3.2)
        di =mesafeHesapla(mevcutKonum, kontrolNoktası); //Denklem (2.1)
        if(Θi >= μi || (i = p && di = ε))
            pencere.push(mevcutKonum);
            break;
```

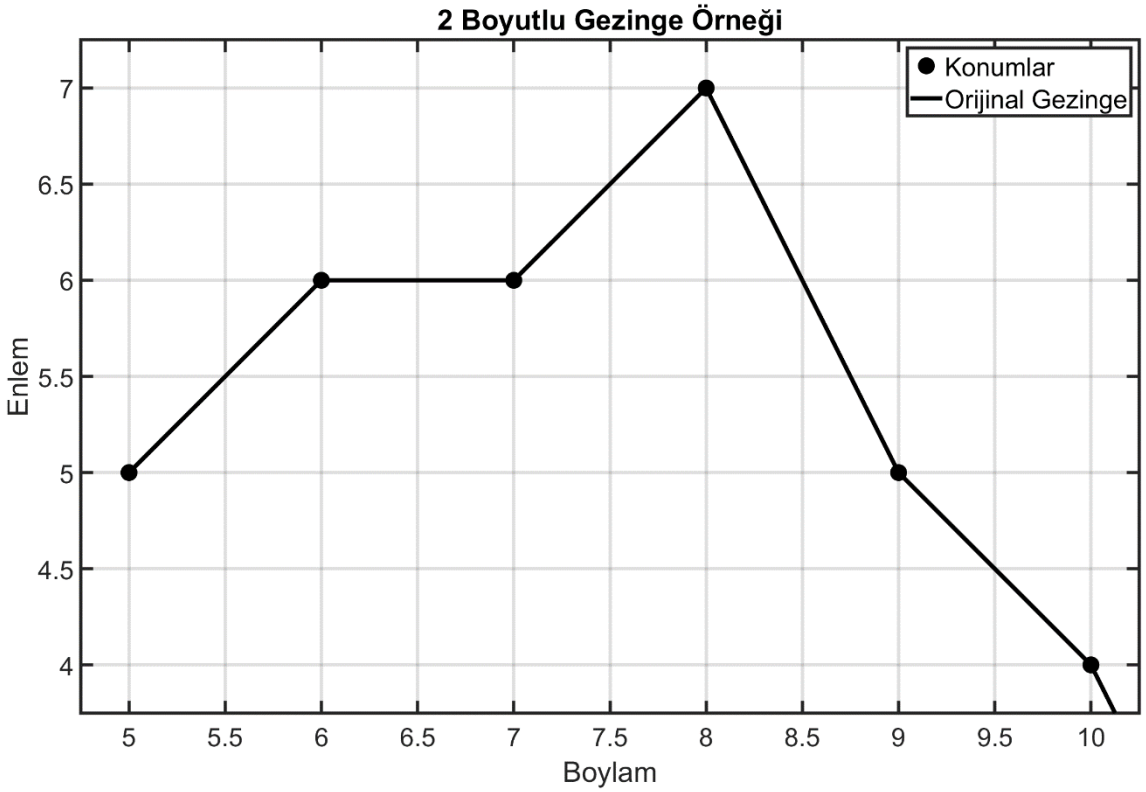
Yöntem, veri akışı olduğu sürece çalışmaktadır. Eğer çevrim içi çalıştırılıyorsa, cihazın GNSS alıcısından gelen veriler dinlenerek yapılmaktadır. Aksi takdirde, çevrim dışı çalışırken bir dosyadaki veriler okunarak yapılmaktadır. Konum verisi bir sensörden alındığında veya bir dosyadan okunduğunda, penceredeki her bir hücre için o hücrenin eşik değerinin aşılıp aşılmadığı bir iterasyon içerisinde kontrol edilmektedir. Eşik değerlerinden herhangi biri aşılırsa o hücredeki nokta için bir dönüş yapıldığı kabul edilir. Bu durumda iterasyon sonlandırılır ve dönme noktası indirgenmiş gezinge dizisine eklenir. Dönme noktası tespit edildiğinde penceredeki tüm noktalar birer birer sola kaydırılır ve orijinal gezingeden gelen bir sonraki nokta pencereye eklenir. Yeni alınan nokta bir dönme noktası değilse pencerede herhangi bir değişiklik yapılmaz. Yöntemin çıktısında indirgenmiş koordinat dizisi elde edilir.

Yöntem ilk çalışmaya başladığında *n* hücreli pencereleme yapıldığında *n-1* adet nokta pencerenin içine alınmaktadır. Örneğin, 3 hücreli pencereleme yapıldığında ilk 2 nokta

kontrol edilmeksizin referans alınabilmesi adına pencereye dahil edilmektedir. İlk adımda, başlangıç noktasına ihtiyaç duyulacağından alıcı tarafından tespit edilen ilk nokta yani gezingenin başlangıç noktası boş olan pencerenin üçüncü hücresine eklenir. İkinci adımda birinci nokta ikinci hücrenin soluna kaydırılır ve ikinci nokta üçüncü hücreye eklenir. Üçüncü adımda, ikinci ve üçüncü hücrelerdeki noktalar birer sola kaydırılır ve üçüncü hücreye yeni okunan üçüncü nokta eklenir. Bu aşamada birinci hücredeki nokta ile ikinci hücredeki noktalar arasındaki kenarın ve ikinci ve üçüncü hücredeki noktalar arasındaki kenarın yönleri hesaplanır. Bu iki kenar arasındaki yön açıları belirlenir. Bu açılar arasındaki fark, önceden belirlenmiş olan eşik değerlerinden daha az ise, iki kenarın yönleri arasında çok fazla fark olmadığı varsayılır ve üçüncü hücredeki nokta pencerelemeden atılır. Yön açıları arasındaki fark önceden belirlenmiş eşik değerlerinden büyükse gezinge üzerinde bir dönüş yapıldığı kabul edilir ve dördüncü nokta, indirgenmiş gezingenin yeni noktası olarak seçilir. Bu durumda penceredeki ilk nokta pencereden çıkarılır ve diğer tüm noktalar birer birer sola kaydırılır. Açısal farkların kontrolü sağlanarak yapılan seçimin yanı sıra yöntemde aşırı hızlanma veya GNSS verisi okunması sırasında meydana gelebilecek sorunlar da göz önünde bulundurularak ortalama yer değiştirmesi kontrolü de yapılmaktadır. Penceredeki kontrol edilen nokta ile bir önceki nokta arasındaki yer değiştirme miktarı ortalama yer değiştirme mesafesinden fazla ise hata olabileceği düşünülür. Bu durumda ilgili noktanın seçimi yapılmakta ve hata oranı azaltılmaya çalışılmaktadır. Son olarak da yöntemin çalışması durduğunda veya GNSS alıcısından yeni nokta okuması yapılmadığı durumunda pencerelemeden atılmış veya seçilmiş olması fark etmeksizin elde edilmiş olan en son nokta da seçilmektedir. Seçilmiş olan her nokta çıkış dizisine kaydedilmektedir ve yöntemin çalışması sonlandırılmaktadır.

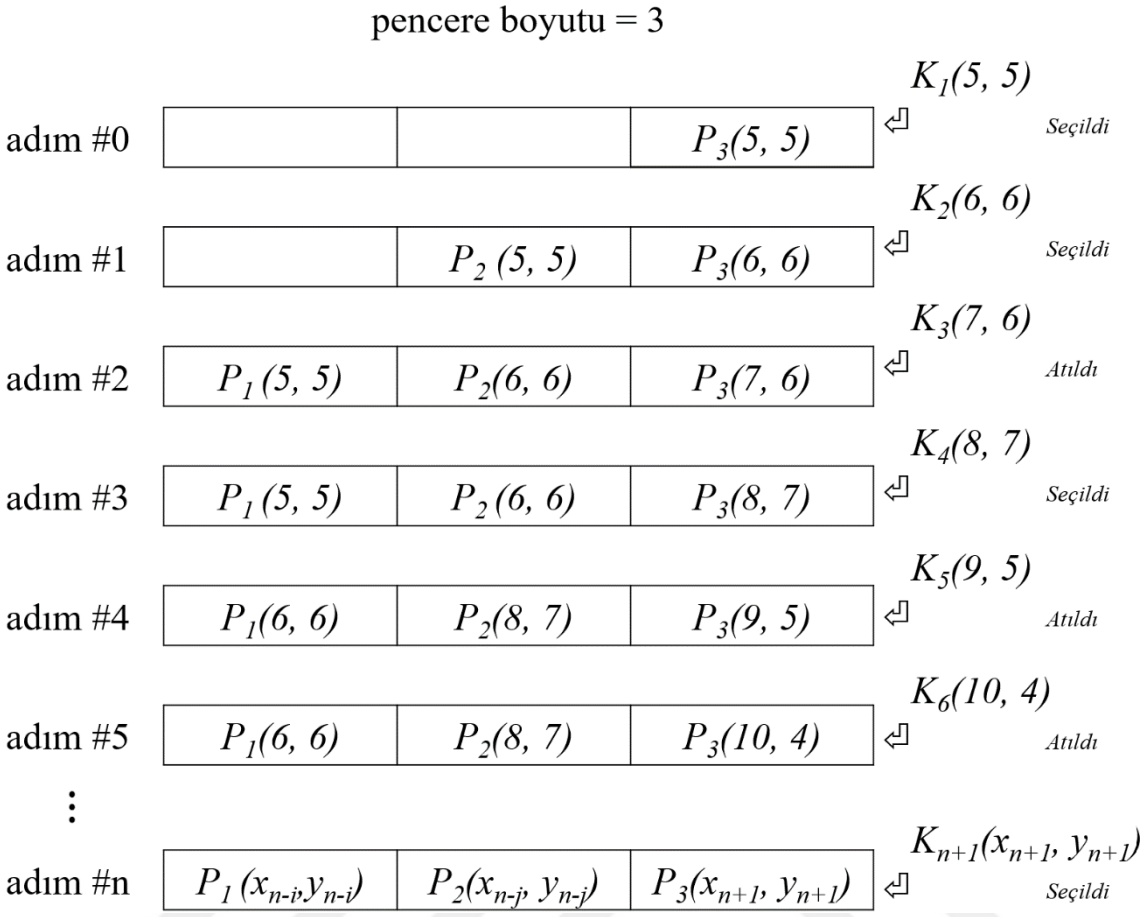
Yöntemin anlaşılmasını sağlamak amacıyla *T ile* temsil edilen ve Şekil 3.2'de gösterilen koordinatların olduğu örnek gezinge ve Şekil 3.3'de gösterilen 3 hücreli pencereleme sistemi kullanılarak ifade etmek gerekirse; ilk durumda mevcut gezingeye ait veri kümesi $T=\{(5, 5, 1),(6, 6, 2),(7, 6, 3),(8, 7, 4),(9, 5, 5),(10, 4, 6),\ldots,(x_n, y_n, t_n),(x_{n+1}, y_{n+1}, t_{n+1})\}$ şeklindedir. Burada; her bir koordinat $K_n(x_n,y_n,t_n)$ şeklindedir. Her bir koordinattaki ilk eleman boylam, ikinci eleman enlem, üçüncü eleman ise zaman etiketini ifade etmektedir. Yöntemin çalışma şeklinin anlaşılmasını kolaylaştırmak için bu kısımdaki görsellerde zaman etiketleri gösterilmemiştir. Ayrıca, *T* veri kümesindeki veriler bir GNSS

alıcısından çevrim içi bir şekilde anlık olarak okunuyor ya da çevrim dışı bir şekilde daha önce bir dosyaya kaydedilmiş veri kümesinden okunuyor olabilir. Yöntem her iki durumda da aynı parametreler ve aynı veriler ile aynı sonuçları üretecektir.



Şekil 3.2.Yöntemin açıklaması için kullanılan örnek gezinge

Adım 0'da pencereleme sistemi boş olduğundan ilk okunan konum olan $K_1(5, 5, 1)$, pencereleme sisteminin üçüncü hücresine yerleştirilmektedir. Referans alınabilecek herhangi bir nokta bulunmadığından bu nokta doğrudan seçilmektedir. Ardından adım 1'de de adım 0'da olduğu gibi okuma yapılan ikinci konum olan $K_2(6, 6, 2)$ elde edildiğinde referans alınabilecek yeterli nokta bulunmadığından seçilir. Bu aşamada P_3 hücresinde bulunan (5, 5, 1) noktası P_2'ye kaydırılır ve yeni elde edilmiş olan $K_2(6, 6, 2)$ noktası pencereleme sistemindeki P_3 hücresine yerleştirilir. Adım 2'de pencereleme sisteminde P_1 hücresi halen boş olduğundan P_2 ve P_3 hücrelerinde bulunan (5, 5, 1) ve (6, 6, 2) noktaları birer sola kaydırılır. Bu sayede yeni okunan $K_3(7, 6, 3)$ seçim kontrolü yapılmak için P_3 hücresine yerleştirilir. Ardından $E(\overline{P_1P_3})$ ve $E(\overline{P_2P_3})$ kenarlarının yönleri, $\theta(P_1P_3)$ ve $\Delta(P_2P_3)$ açıları μ_1 ve μ_2 eşik değerleri ile, $E(\overline{P_2P_3})$ kenarının uzunluğu ise ε ortalama yer değiştirme mesafesi ile kontrol edilmektedir. P_3 hücresinde bulunan $K_3(7, 6, 3)$ bu kriterleri sağlamadığından pencereleme sisteminden atılmaktadır.

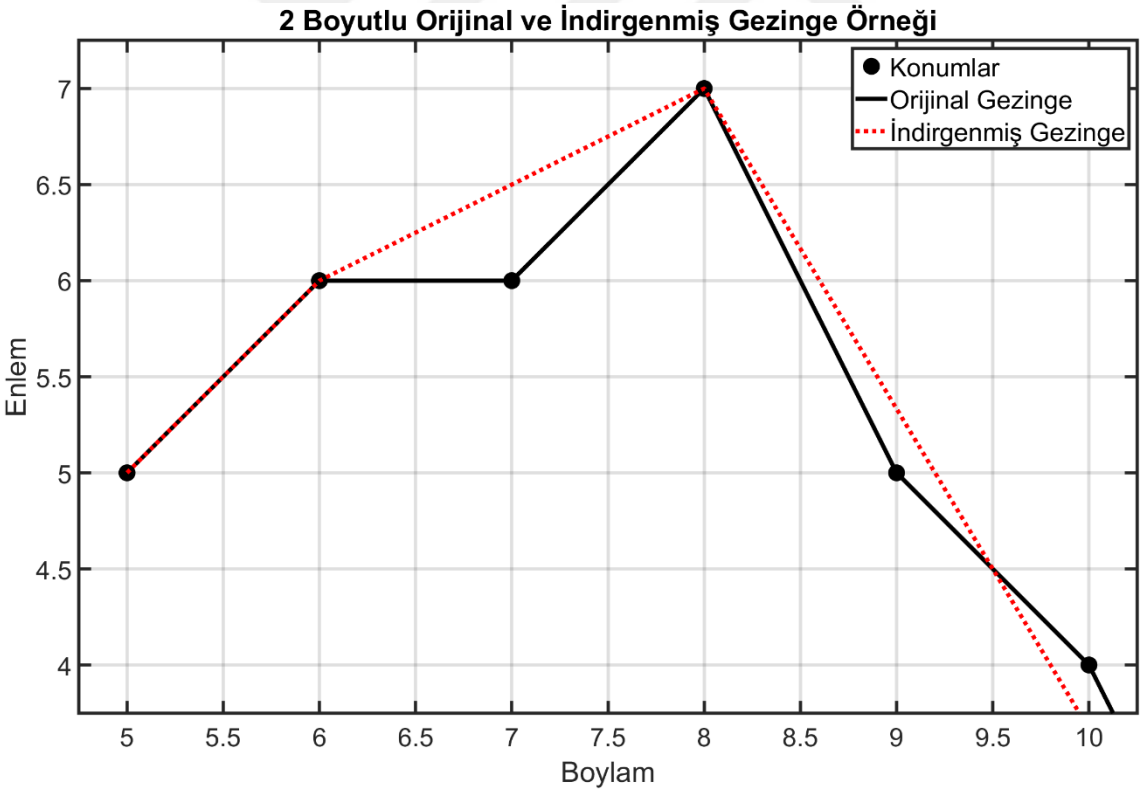


Şekil 3.3.Yöntemin çalışması sırasında kullanılan 3 hücreli pencereleme sistemi örneği

Adım 3'te ise bir önceki adımda kontrol edilen nokta atıldığından yeni okunan K_4(8, 7, 4) noktası P_3 hücresine yerleştirilir. Adım 2'de gerçekleştirilen hesaplamalar tekrardan gerçekleştirilir ve indirgeme kriterinden biri veya birkaçı sağlandığından P_3 hücresinde bulunan K_4(8, 7, 4) noktası seçilir. Seçim gerçekleştiği için P_1 hücresinde bulunan (5, 5, 1) noktası çıkış dizisine kaydedilir ve P2 hücresindeki (6, 6, 2) noktası P_1 hücresine, P_3 hücresindeki (8, 7, 4) noktası da P_2 hücresine kaydırılır. Bu sayede yeni elde edilecek noktaların kontrolü için P_3 hücresi boş bırakılmaktadır. Adım 4'te ise yeni okunan nokta olan K_5(9, 5, 5) P_3 hücresine yerleştirilir ve önceki adımlarda yapılan kontroller bu nokta için de aynı şekilde gerçekleştirilir. P_3 hücresinde bulunan K_5(9, 5, 5) noktasının herhangi bir indirgeme kriterini sağlamadığı tespit edildiğinden pencereleme sisteminden atılır. Benzer şekilde adım 5'te okunan K_6(10, 4, 6) noktası da önceki adımlardaki işlemlere tabi tutulur ve herhangi bir indirgeme kriterini sağlamadığı tespit edildiğinden pencereleme sisteminden atılır. Her bir yeni nokta okuması yapıldığında üstte bahsedilen adımlardaki işlemler tekrarlanmaktadır. Adım *n*'e gelindiğinde yeni okunan $K_{n+1}(x_{n+1},$

y_{n+1}, t_{n+1}) noktası pencereleme sistemindeki P_3 hücresine eklenir. Önceki adımlarda uygulanan kontroller P_3 hücresinde o anda bulunan $K_{n+1}(x_{n+1}, y_{n+1}, t_{n+1})$ noktasına da uygulanır. Bu nokta da indirgeme kriterlerini sağladığı için seçilebilir. Eğer kriterleri sağlamıyorsa ve yeni nokta okuması yapılmıyorsa da pencerenin içerisinde bulunduğundan seçimi gerçekleştirilir. Bu sayede gezingenin başlangıç ve bitiş noktaları belirlenmiş olur. Daha önce normal akış sırasında seçildiği için pencereleme sisteminde P_1 hücresine kadar gelmiş ve yeni noktaların seçilmesi nedeniyle pencerelemeden çıkarılıp çıkış dizisine kaydedilen noktaların ardına pencereleme sisteminin içindeki son durumda bulunan noktalar da kaydedilir. Bu sayede seçilen tüm noktalar çıkış dizisine kaydedilmiş olur ve yöntem sonlandırılır.

Yöntemin çalıştırılması sonucunda elde edilecek olan indirgenmiş gezinge veri kümesi $T`=\{(5, 5, 1),(6, 6, 2),(8, 7, 4),\ldots,(x_{n+1}, y_{n+1}, t_{n+1})\}$ şeklinde, görselleştirilmiş hali ise Şekil 3.4'de gösterildiği gibi olacaktır.



Şekil 3.4.Orijinal gezinge ve yöntemin çalışması sonucunda elde edilen indirgenmiş gezinge örneği

3.3.Performans Değerlendirmesi

Bu tez çalışması kapsamında geliştirilmiş olan yöntemin performansını değerlendirebilmek amacıyla bazı denemeler yapılmıştır. Testler Microsoft GeoLife veri kümesindeki bazı dosyalar üzerinde gerçekleştirilmiştir. Elde edilen sonuçların anlamlandırılabilmesi amacıyla geliştirilen yöntemin ve RDP algoritmasının sonuçları karşılaştırılmaktadır.

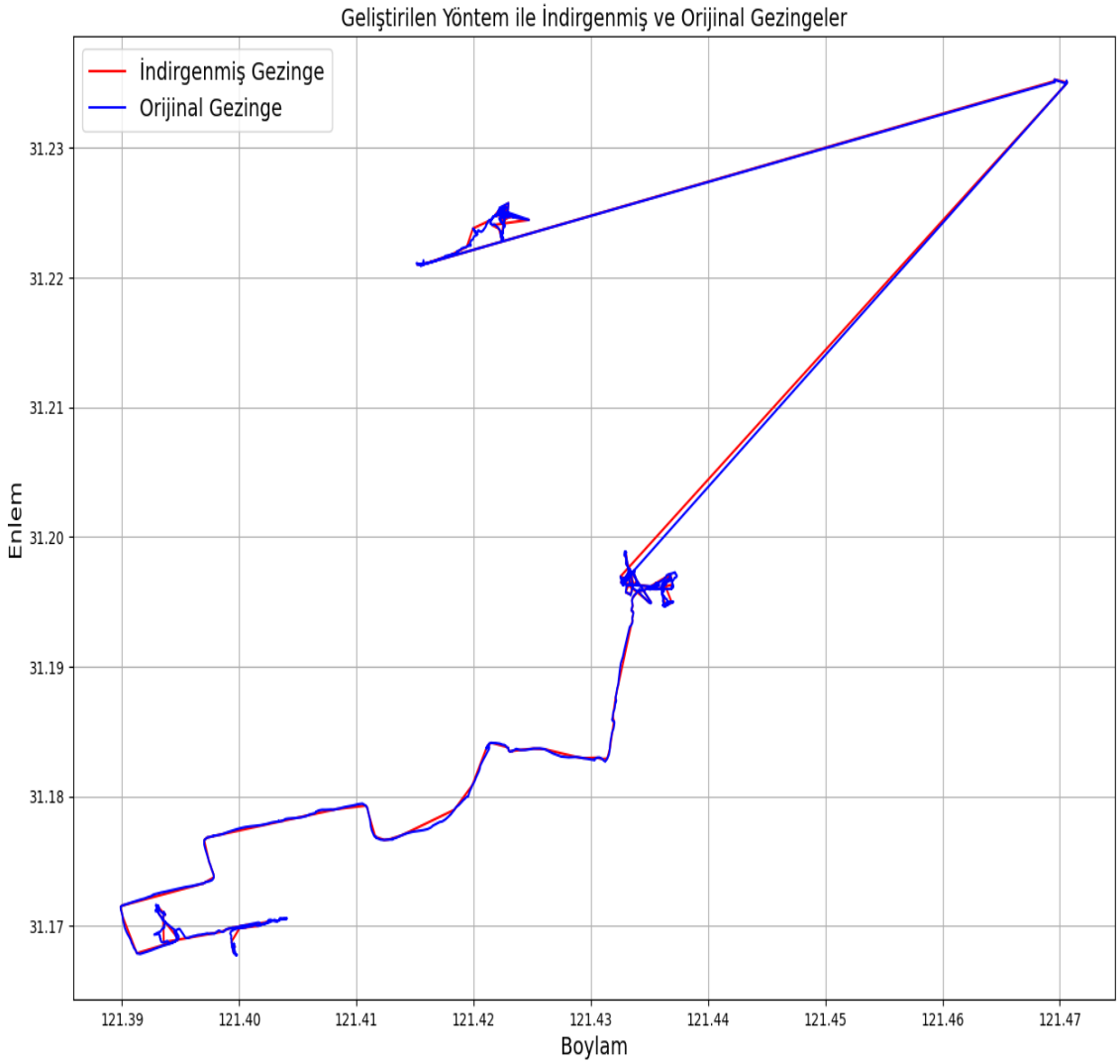
İlk örnekte Microsoft GeoLife veri kümesindeki "20090522094234.plt" dosyası üzerinde denemeler yapılmıştır. Test sonuçları Tablo 3.2'de gösterilmektedir.

Tablo 3.2.Önerilen yöntemin ve RDP algoritmasının sonuçlarının "20090522094234.plt" dosyası üzerinden kıyaslanması

	Önerilen Yöntem	**RDP Algoritması**
Konum Sayısı	2560	2560
İndirgeme Sonrası Konum Sayısı	170	55
İndirgeme Oranı	%93,36	%97,85
Hesaplama Süresi (ms)	14,41	113,19
RMSE Değeri	1,44114e-03	1,60097e+01
Eşik Değer	$\theta = 5°$	$\varepsilon = 0.001$
Pencereleme Boyutu	3	-

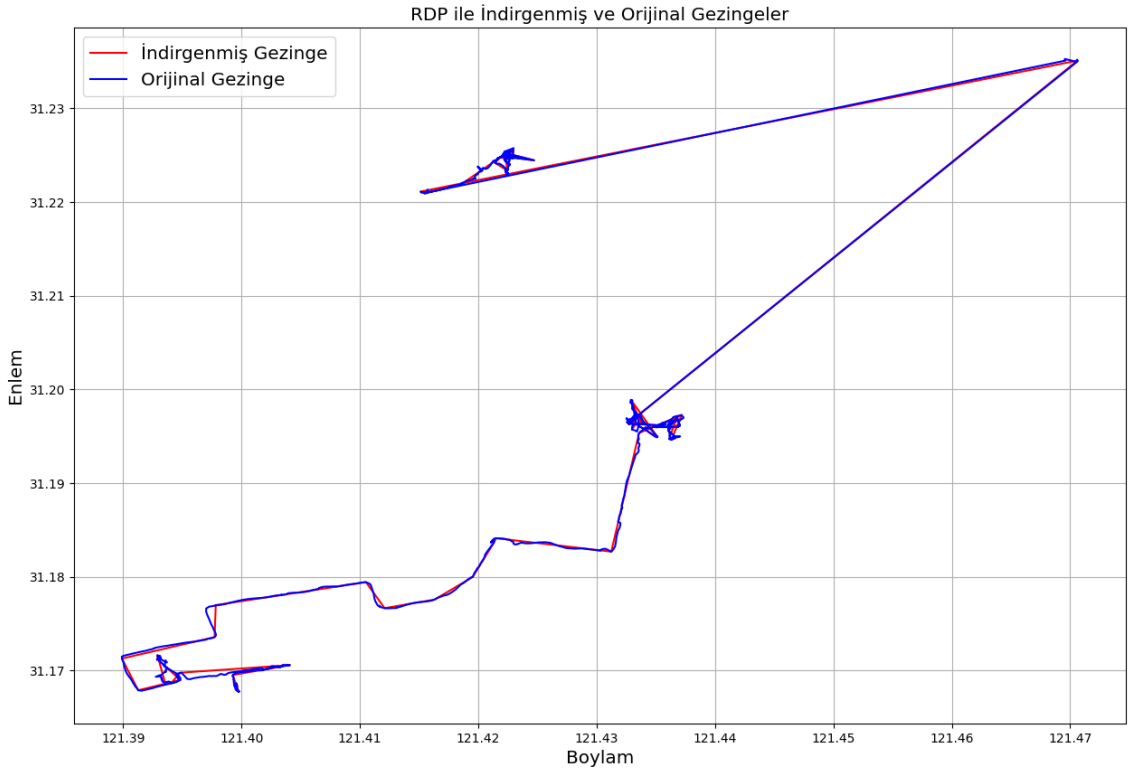
Bu örnekte pencereleme boyutu 3 ve eşik değeri 5° seçilerek Microsoft GeoLife veri kümesindeki "20090522094234.plt" dosyasına uygulanmıştır. Bu dosyanın içerisinde 2560 adet koordinat bulunmaktadır. Bu girdiler kullanılarak yapılan deneme sonucunda 2390 adet nokta indirgenerek aynı dosya 170 nokta ile temsil edilmiştir. Bu işlem sonucunda %93,36 oranında indirgeme sağlanmıştır. Ayrıca RMSE değerinin 1,44114e-

03, hesaplama süresinin ise 14,51ms olduğu gözlemlenmiştir. Orijinal gezinge ve yöntemin çalışması sonucunda elde edilen indirgenmiş gezingenin görsel açıdan kıyaslanabilmesi amacıyla bir arada gösterimi Şekil 3.5'teki gibidir.



Şekil 3.5.Önerilen yöntemin "20090522094234.plt" dosyası üzerinden görsel kıyaslaması

Aynı veri kümesi üzerinde RDP algoritması eşik değeri 0.001 seçilerek koşulmuştur. Bu girdiler kullanılarak RDP algoritması ile yapılan deneme sonucunda ise 2505 adet nokta indirgenerek aynı dosya 55 nokta ile temsil edilmiştir. Bu işlem sonucunda %97,85 oranında indirgeme sağlanmıştır. Ancak RMSE değerinin 1,60097e+01, hesaplama süresinin ise 113,19 ms olduğu gözlemlenmiştir. Orijinal gezinge ve RDP algoritmasının çalışması sonucunda elde edilen indirgenmiş gezingenin görsel açıdan kıyaslanabilmesi amacıyla bir arada gösterimi Şekil 3.6'daki gibidir.



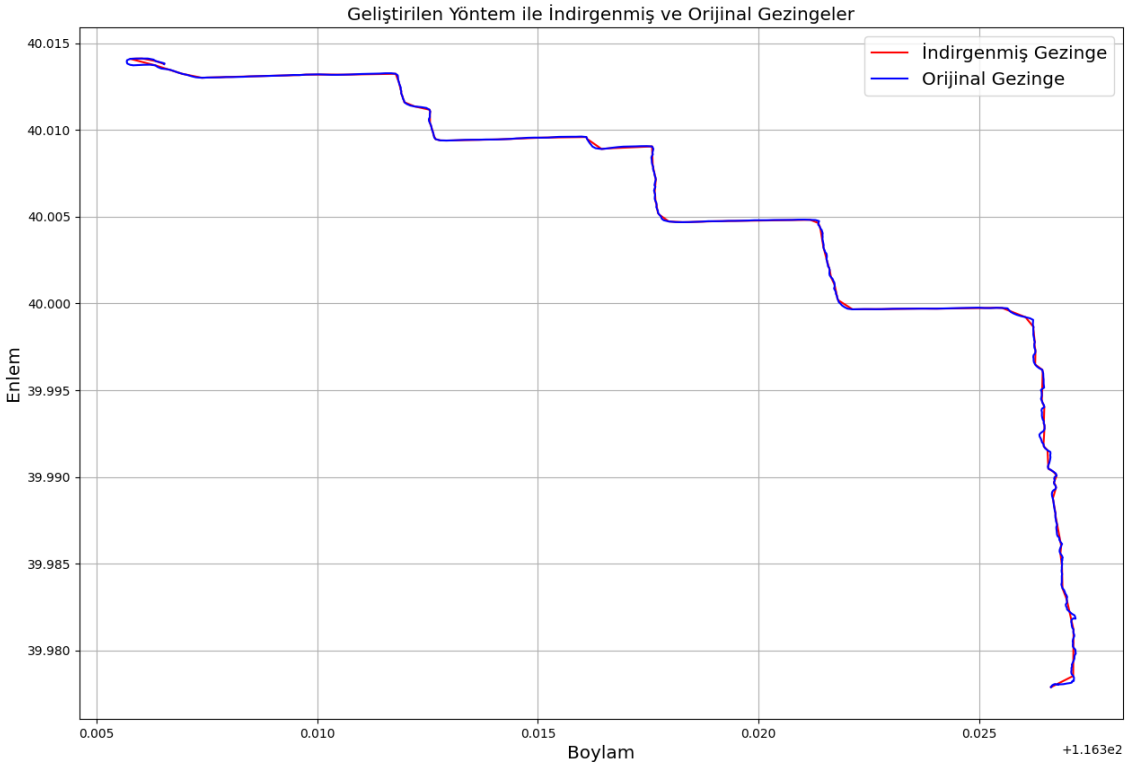
Şekil 3.6.RDP algoritmasının "20090522094234.plt" dosyası üzerinden görsel kıyaslaması

İkinci örnekte Microsoft GeoLife veri kümesindeki "20081214235553.plt" dosyası üzerinde denemeler yapılmıştır. Test sonuçları Tablo 3.3'te gösterilmektedir.

Tablo 3.3.Önerilen yöntemin ve RDP algoritmasının sonuçlarının "20081214235553.plt" dosyası üzerinden kıyaslanması

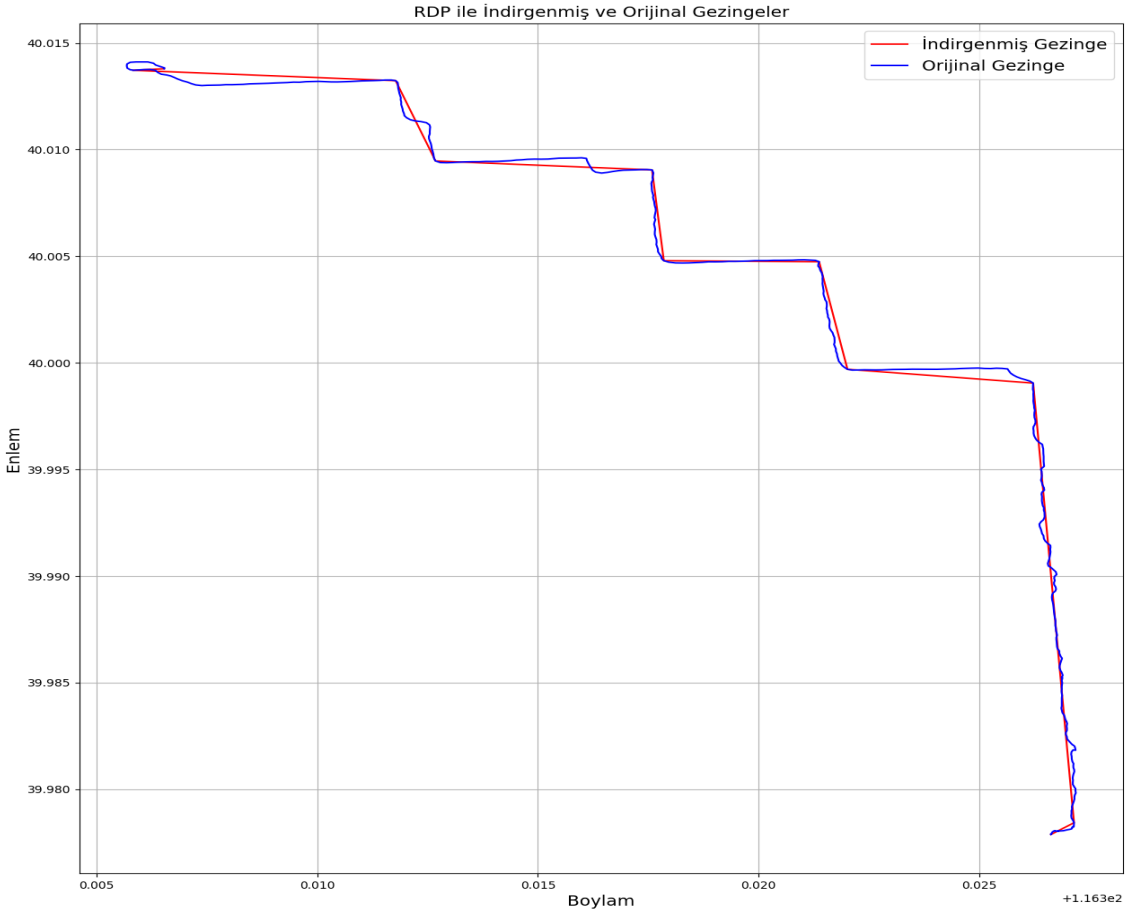
	Önerilen Yöntem	**RDP Algoritması**
Konum Sayısı	539	539
İndirgeme Sonrası Konum Sayısı	113	11
İndirgeme Oranı	%79,04	%97,96
Hesaplama Süresi (ms)	10,4	13,11
RMSE Değeri	5.08088e-05	2.82786e-04
Eşik Değer	$\theta = 20°$	$\varepsilon = 0.0001$
Pencereleme Boyutu	3	-

Bu örnekte pencereleme boyutu 3 ve eşik değeri 20° seçilerek Microsoft GeoLife veri kümesindeki “20081214235553.plt” dosyasına uygulanmıştır. Bu dosyanın içerisinde 539 adet koordinat bulunmaktadır. Bu girdiler kullanılarak yapılan deneme sonucunda 426 adet nokta indirgenerek aynı dosya 113 nokta ile temsil edilmiştir. Bu işlem sonucunda %79,04 oranında indirgeme sağlanmıştır. Ayrıca RMSE değerinin 5.08088e-05, hesaplama süresinin ise 10,4 ms olduğu gözlemlenmiştir. Orijinal gezinge ve yöntemin çalışması sonucunda elde edilen indirgenmiş gezingenin görsel açıdan kıyaslanabilmesi amacıyla bir arada gösterimi Şekil 3.7’deki gibidir.



Şekil 3.7.Önerilen yöntemin “20081214235553.plt” dosyası üzerinden görsel kıyaslaması

Aynı veri kümesi üzerinde RDP algoritması eşik değeri 0.001 seçilerek koşulmuştur. Bu girdiler kullanılarak RDP algoritması ile yapılan deneme sonucunda ise 539 adet nokta indirgenerek aynı dosya 11 nokta ile temsil edilmiştir. Bu işlem sonucunda %97,95 oranında indirgeme sağlanmıştır. Ancak RMSE değerinin 2.82786e-04, hesaplama süresinin ise 13,11 ms olduğu gözlemlenmiştir. Orijinal gezinge ve RDP algoritmasının çalışması sonucunda elde edilen indirgenmiş gezingenin görsel açıdan kıyaslanabilmesi amacıyla bir arada gösterimi Şekil 3.8’deki gibidir.



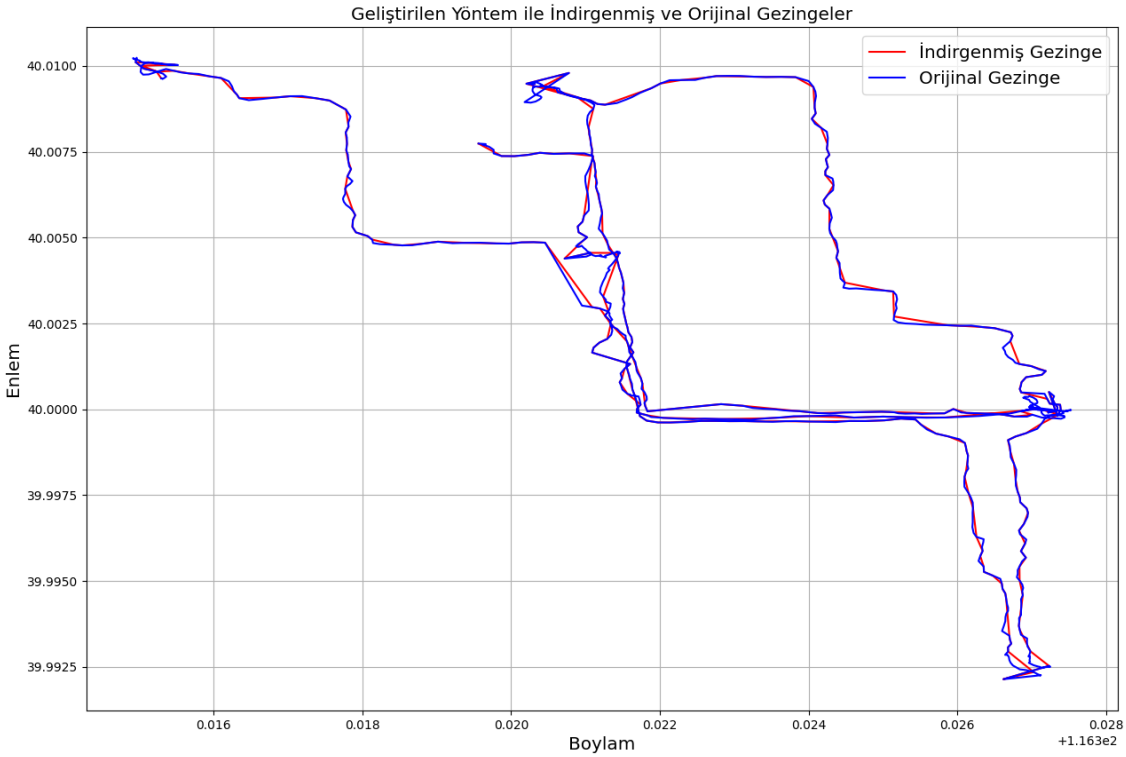
Şekil 3.8.RDP algoritmasının "20081214235553.plt" dosyası üzerinden görsel kıyaslaması

Üçüncü örnekte ise yine Microsoft GeoLife veri kümesinde bulunan "20090608060034.plt" dosyası üzerinde denemeler yapılmıştır. Test sonuçları Tablo 3.4'te gösterilmektedir.

Tablo 3.4.Önerilen yöntemin ve RDP algoritmasının sonuçlarının "20090608060034.plt" dosyası üzerinden kıyaslanması

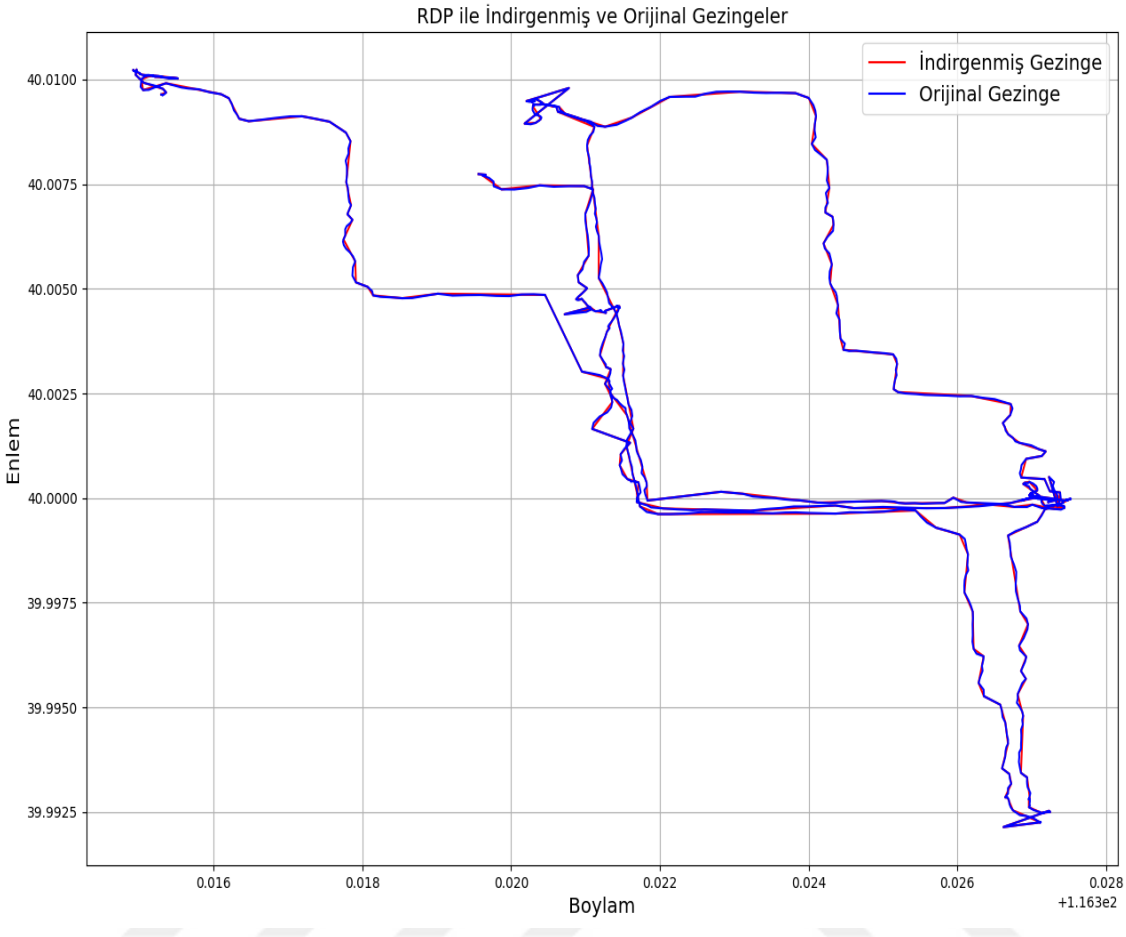
	Önerilen Yöntem	**RDP Algoritması**
Konum Sayısı	875	875
İndirgeme Sonrası Konum Sayısı	290	226
İndirgeme Oranı	%66,86	%74,17
Hesaplama Süresi (ms)	41,43	1005,77
RMSE Değeri	1.11122e-04	1.02031e+01
Eşik Değer	$\theta = 10°$	$\varepsilon = 0{,}00005$
Pencereleme Boyutu	3	-

Bu örnekte pencereleme boyutu 3 ve eşik değeri 10° seçilerek Microsoft GeoLife veri kümesindeki “20090608060034.plt” dosyasına uygulanmıştır. Bu dosyanın içerisinde 875 adet koordinat bulunmaktadır. Bu girdiler kullanılarak yapılan deneme sonucunda 585 adet nokta indirgenerek aynı dosya 290 nokta ile temsil edilmiştir. Bu işlem sonucunda %66,86 oranında indirgeme sağlanmıştır. Ayrıca RMSE değerinin 1.11122e-04 , hesaplama süresinin ise 41,43 ms olduğu gözlemlenmiştir. Orijinal gezinge ve yöntemin çalışması sonucunda elde edilen indirgenmiş gezingenin görsel açıdan kıyaslanabilmesi amacıyla bir arada gösterimi Şekil 3.9’daki gibidir.



Şekil 3.9.Önerilen yöntemin “20090608060034.plt” dosyası üzerinden görsel kıyaslaması

Aynı veri kümesi üzerinde RDP algoritması eşik değeri 0.0005 seçilerek koşulmuştur. Bu girdiler kullanılarak RDP algoritması ile yapılan deneme sonucunda ise 875 adet nokta indirgenerek aynı dosya 226 nokta ile temsil edilmiştir. Bu işlem sonucunda %74,17 oranında indirgeme sağlanmıştır. Ancak RMSE değerinin 1.02031e+01, hesaplama süresinin ise 1005,77 ms olduğu gözlemlenmiştir. Orijinal gezinge ve RDP algoritmasının çalışması sonucunda elde edilen indirgenmiş gezingenin görsel açıdan kıyaslanabilmesi amacıyla bir arada gösterimi Şekil 3.10’daki gibidir.



Şekil 3.10.RDP algoritmasının "20090608060034.plt" dosyası üzerinden görsel kıyaslaması

Geliştirilen yöntemin indirgeme oranı RDP'ye göre düşük olmasına rağmen elde edilen indirgenmiş gezingenin RMSE hatası da düşüktür. Geliştirilen yöntem için kullanılan eşik değeri yükseltilerek indirgeme oranı artırılabilir, ancak bu durumda elde edilen indirgenmiş gezingenin RMSE hatası da artacaktır. Eşik değerinin düşük ya da yüksek olması durumundan bağımsız olarak geliştirilen yöntemin daha hızlı çalıştığı aşikardır. Bu bakımdan geliştirilen yöntem hem daha hızlı hem de uygun eşik değeri kullanılarak kabul edilebilir bir indirgeme oranıyla ve RMSE hatasıyla indirgeme yapmaktadır.

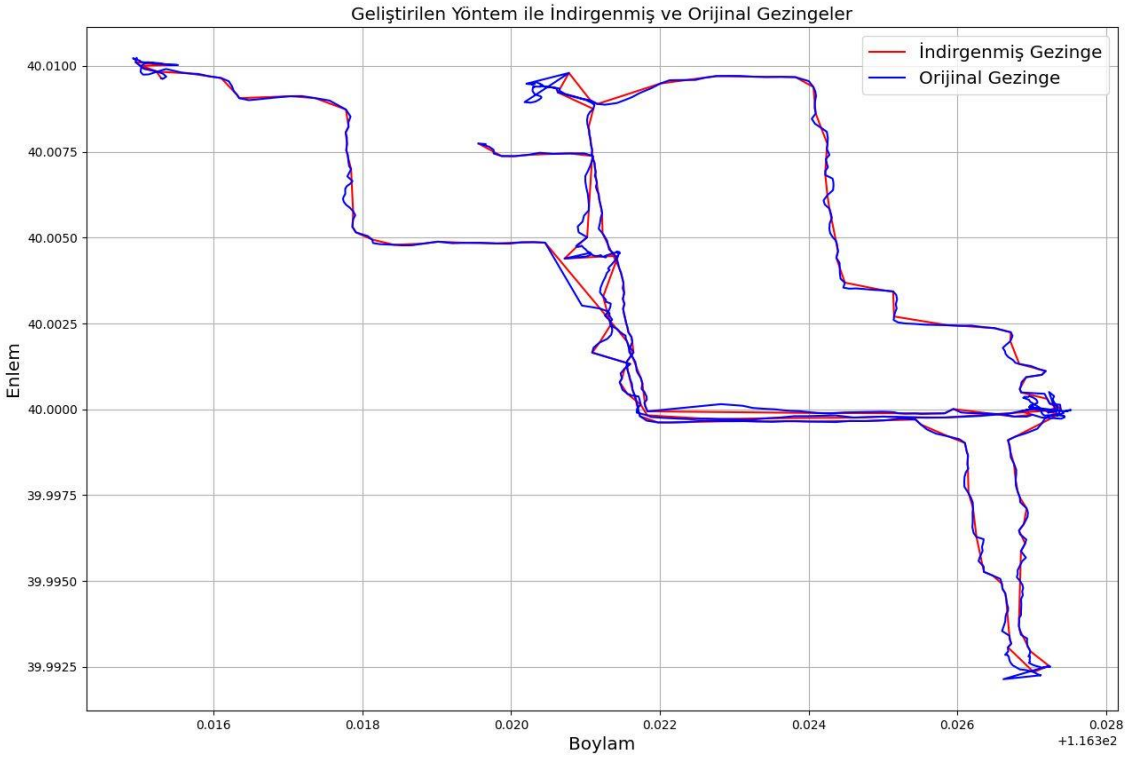
Dördüncü örnekte ise yine Microsoft GeoLife veri kümesinde bulunan "20090608060034.plt" dosyası üzerinde denemeler yapılmıştır. Test sonuçları Tablo 3.4'te gösterilmektedir.

Tablo 3.5.Önerilen yöntemin ve RDP algoritmasının sonuçlarının “20090608060034.plt” dosyası üzerinden kıyaslanması

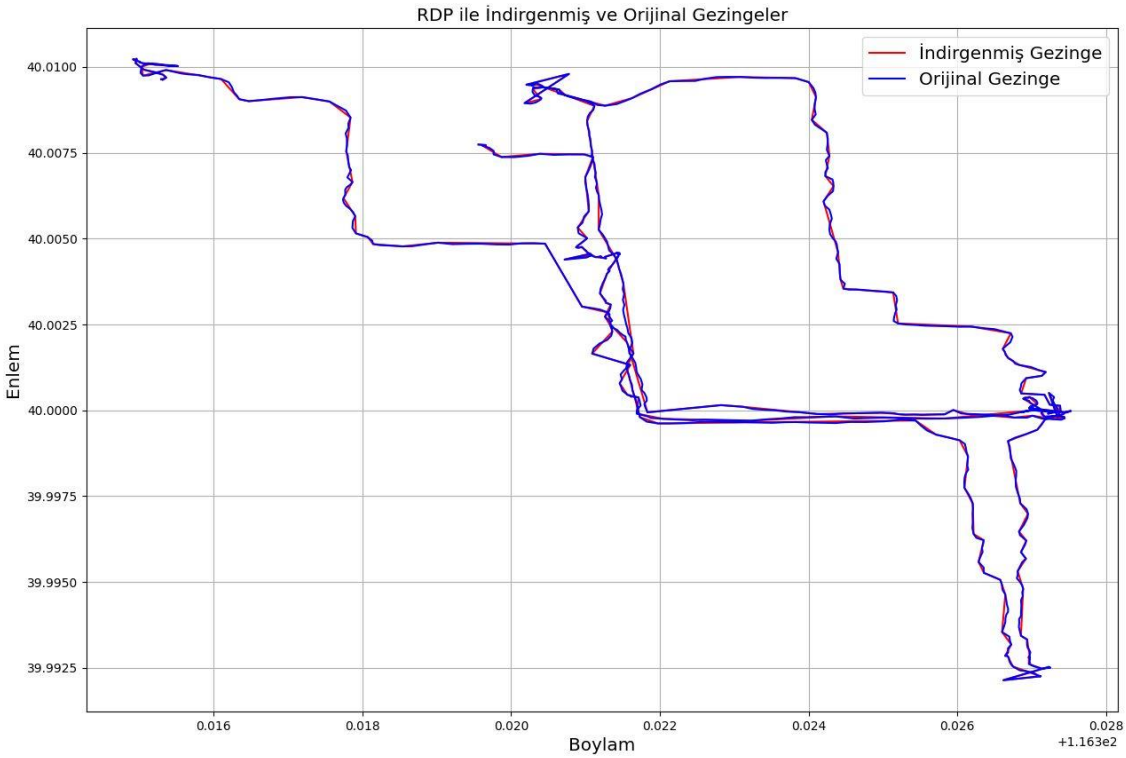
	Önerilen Yöntem	**RDP Algoritması**
Konum Sayısı	875	875
İndirgeme Sonrası Konum Sayısı	199	199
İndirgeme Oranı	%77,26	%77,26
Hesaplama Süresi (ms)	30,37	1121,01
RMSE Değeri	1.64900e-04	1.02031e+01
Eşik Değer	$\theta = 30°$	$\varepsilon = 0{,}00006$
Pencereleme Boyutu	3	-

Bu örnekte önerilen yöntem pencereleme boyutu 3 ve eşik değeri 30°, RDP algoritması ise eşik değeri 0,00006 seçilerek Microsoft GeoLife veri kümesindeki “20090608060034.plt” dosyasına ayrı ayrı uygulanmıştır. Her iki yöntemin çalışması sonucunda da bu dosyanın içerisinde bulunan 875 adet koordinat 676 adet nokta indirgenerek 199 nokta ile temsil edilmiştir. Her iki yöntem için de %77,26 oranında indirgeme sağlanmıştır. Geliştirilen yönteme ait RMSE değerinin 1,64900e-04, RDP algoritmasının RMSE değerinin ise 1,02031e+01 olduğu görülmüştür. Ayrıca geliştirilen yöntemin hesaplama süresinin 30,37 ms, RDP algoritmasının hesaplama süresinin ise 1121,01 ms olduğu gözlemlenmiştir. Orijinal gezinge ve geliştirilen yöntemin çalışması sonucunda elde edilen indirgenmiş gezingenin görsel açıdan kıyaslanabilmesi amacıyla bir arada gösterimi Şekil 3.11’deki gibidir. Ayrıca RDP algoritmasının çalışması sonucunda elde edilen indirgenmiş gezinge ile orijinal gezingenin görsel kıyaslaması ise Şekil 3.12‘deki gibidir.

Geliştirilen yöntemin indirgeme oranının RDP algoritmasının indirgeme oranına eşit olması durumunda bile geliştirilen yöntem ile indirgenmiş gezingenin RMSE hatası RDP algoritmasının RMSE hatasından daha küçük olduğu görülmektedir. Bunun yanı sıra geliştirilen yöntemin aynı indirgeme oranında bile RDP algoritmasından çok daha hızlı çalıştığı açık olarak görülmektedir. Geliştirilmiş olan yöntemin hem daha düşük hata oranıyla indirgeme yaptığı hem de daha hızlı çalıştığı görüldüğünden geliştirilmiş olan yöntemin daha başarılı bir yöntem olduğu aşikardır.



Şekil 3.11.Önerilen yöntemin "20090608060034.plt" dosyası üzerinden görsel kıyaslaması



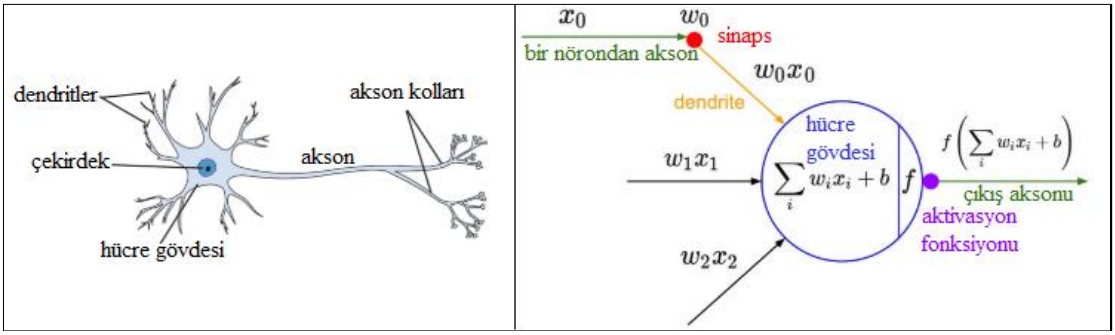
Şekil 3.12.RDP algoritmasının "20090608060034.plt" dosyası üzerinden görsel kıyaslaması

4. YAPAY SİNİR AĞI MODELİ

Bu tez çalışması kapsamında ilk olarak matematiksel bir GNSS verisi indirgeme yöntemi geliştirildiğinden bir önceki bölümde bahsedilmiştir. Geliştirilmiş olan bu matematiksel yöntemin indirgeme sonuçları kullanılarak bir yapay sinir ağı modeli eğitimi gerçekleştirilmiştir. Bu sayede, matematiksel yöntemin katı bir kurala bağlı olan indirgeme işlemi, eğitilmiş olan yapay sinir ağı yardımıyla makineye öğretilmiş ve GNSS'ten elde edilen konum verisi yapay sinir ağı kullanılarak indirgenmiştir. Yapılan literatür araştırmaları sırasında, GNSS verisi indirgemesi konusunda herhangi bir yapay zekâ kullanılmadığı tespit edilmiştir. Bahsedilen bu eksiklik bu tez çalışması ile giderilmiştir. Ayrıca GNSS verisi indirgemesi konusunda yapay zekânın kullanılmasının önü açılmaya çalışılmıştır. Eğitilen yapay sinir ağı modeli için Google'a ait TensorFlow çerçevesi kullanılmıştır.

4.1.Yapay Sinir Ağı Nedir?

Yapay Sinir Ağı (YSA), biyolojik sinir ağı sistemleri modellenerek katmanlar halinde oluşturulmuş bir yapay sinir ağı modelidir. İlk olarak insan sinir hücresi 1943 yılında Warren McCulloch ve Walter Pitts tarafından matematiksel olarak modellenmiştir (McCulloch & Pitts, 1943). Ardından, YSA'lar farklı alanlarda kullanılmak üzere geliştirilmeye devam edilmiştir. YSA, esinlenildiği insan sinir hücresine benzer şekilde hücre dışından verilerin alındığı giriş, bu verilerin toplanarak aktivasyon fonksiyonundan geçirildiği işlem merkezi ya da hücre merkezi ve işlem merkezinden elde edilen sonuçların değerlendirildiği çıkış olacak biçimde modellenmiştir. Örnek bir sinir hücresi ve YSA hücresinin modeli Şekil 4.1'de gösterilmektedir.

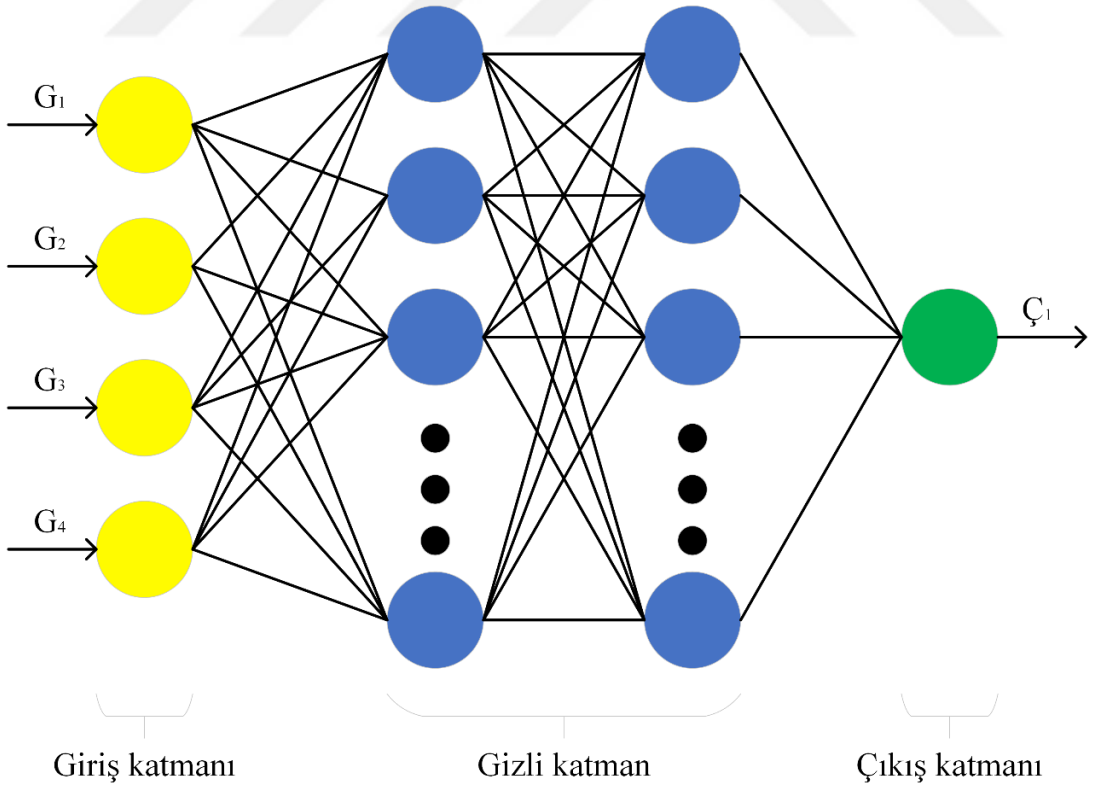


Şekil 4.1. Yapay sinir hücresinin modellenmesi (Cantemir, 2019)

YSA'larda bir hücrenin çıkışı Denklem (4.1)'de gösterildiği gibi ifade edilmektedir. En çok kullanılan YSA modeli olan Geri Yayılımlı YSA modelinin genel yapısı ise Şekil 4.2'de gösterilmektedir. YSA'lar, girdi olarak kullanılan verileri işleyerek sonuçlar üretmektedir. Başlangıçta, ağın katmanlarında bulunan nöronlara rastgele ağırlıklar ve eşik değerleri atanmaktadır. Ağın eğitilmesi ile ağırlıkların olması gereken değerleri alması sağlanmaktadır. Eğitim sonucunda elde edilen bu ağırlıklar ve eşik değerleri, YSA'nın öğrenme yeteneğini ve sonuçların doğruluğunu belirlemektedir. YSA'lar, çeşitli veri kümelerindeki kalıpları öğrenmek amacıyla kullanılmaktadır.

$$y = f(\sum wx + b) \tag{4.1}$$

Burada, y çıkış değerini, f doğrusal olmayan aktivasyon fonksiyonunu, x hücreye dışarıdan alınan verileri, w toplayıcıya iletilen ağırlıkları, b ise eşik değerini temsil etmektedir. Dışarıdan alınan veriler ağırlıklandırılarak toplayıcıya iletilir ve toplayıcıda bias değeri eklenerek elde edilen bu veriler doğrusal olmayan bir aktivasyon fonksiyonundan geçirildikten sonra çıkışa iletilmektedir.



Şekil 4.2. YSA'ya ait örnek ağ modeli

YSA'lar katmanların birbirine bağlanma biçimlerine göre *İleri Beslemeli* ve *Geri Beslemeli Ağlar* olarak iki grupta incelenebilmektedir. Bir katmanın hücre çıkışlarının bir sonraki katmandaki hücrelerin girişleri olduğu ağ yapısı *İleri Beslemeli Ağ* olarak isimlendirilmektedir. Bu tipteki ağlar doğrusal olmayan sabit işlevleri gerçekleştirmektedirler. *Geri Beslemeli Sinir Ağı*, çıktıların önceki katmanlarda üretilen çıktılarla veya o andaki durumlarla geri beslendiği sinir ağı modelidir. Bu geri besleme mekanizması, geçmiş bilgilerin mevcut hesaplamalara dahil edilmesine ve zaman içinde değişen dinamik ilişkilerin modellenmesine olanak sağlamaktadır. Bu sinir ağı türü, geçmiş deneyim ve bilgiyi kullanarak gelecekteki çıktıları tahmin etmek veya mevcut girdilere bağlı olarak davranışları öngörmek gibi problemlerin çözümünde kullanılmaktadır. Bu katmanda ağın yapısına ve çözülmesi istenilen problemin durumuna göre RMSE, Çapraz Entropi gibi farklı hata ölçüm yöntemleri kullanılmaktadır. Çapraz entropi, sınıflandırma problemlerinde YSA'nın çıkış değerleri ile beklenen değerler arasındaki farkın olasılıksal olarak hesaplanmasında yaygın olarak kullanılmaktadır. Denklem (4.2)'deki formül ile hesaplanmaktadır. Hesaplanan olasılık değeri ile beklenen olasılık değerleri arasındaki farkın artması çapraz entropi kaybının arttığını göstermektedir. İdeal durumda bu farkın 0 olması istenir (Koşan ve diğ., 2019; Z. Zhang & Sabuncu, 2018).

$$H(p,q) = -\sum_{i,j} p(i,j) \log q(i,j) \tag{4.2}$$

Yapay sinir ağları; karmaşık verileri sınıflandırma, tahmin yapma, örüntü tanıma, dil işleme, resim tanıma, işaretlerin tanınması, öneri sistemleri, nesne tespiti, kontrol sistemleri ve daha pek çok görevi yerine getirmek için kullanılabilirler. Başarılı sonuçlar elde edebilmeleri için uygun veri kümeleri kullanılarak doğru veriler ile eğitilmiş olmaları gerekmektedir. Yaygın kullanım alanlarından bazıları aşağıdaki gibidir.

1. Doğal Dil İşleme: Yapay sinir ağları, doğal dil işleme alanında yaygın olarak kullanılmaktadır. YSA'lar, öğrenme süreci sırasında, doğal dildeki kelime ve cümle yapılarını tanıyabilmektedir.
2. Örüntü Tanıma: YSA'lar, örüntülerdeki nesne ve desenleri tanımlamak için kullanılabilmektedir. Bu nedenle robotik ve güvenlik sistemleri gibi alanlarda sıklıkla uygulanmaktadır.

3. Ses Tanıma: YSA'lar, ses tanıma sistemlerinde de yaygın olarak kullanılabilmektedir. YSA'lar, bir kişinin konuşmasını tanıyarak, söylenen kelime veya cümleleri yazıya dönüştürebilmektedir.
4. Öneri ve Tahmin Sistemleri: YSA'lar, geliştirilen uygulama alanına göre kullanıcılara özel öneri ve tahminler sunabilmektedir.
5. Kontrol Sistemleri: YSA'lar kontrol sistemleri gibi sistemlerde de kullanılabilmektedir. Örneğin, bir basınç tankı YSA kullanılarak yönetilebilmektedir ya da bir robotun kontrolü sağlanabilmektedir. Bunların yanı sıra YSA desteği ile kontrol sistemlerinin performansı da artırabilmektedir.
6. Finansal Tahmin: YSA'lar, borsa fiyatlarındaki değişiklikleri veya finansal verilerin tahminleri gibi alanlarda da kullanılabilmektedir.

YSA'lar, öğrenme süreci sırasında veri kümelerindeki giriş-çıkış ilişkisini tanımaktadırlar ve bu ilişkileri kullanarak yeni verileri sınıflandırmakta veya tahminler yapabilmektedirler. Örneğin, bir görüntü tanıma YSA'sı, verilen bir görüntüyü sınıflandırmak için görüntünün içerisindeki desenleri ve özellikleri analiz etmektedir. Bu nedenle, YSA'lar, özellikle yapay zekâ ve makine öğrenmesi alanlarında çok yaygın olarak kullanılmaktadırlar. Geleneksel programlama yaklaşımlarına göre daha esnek ve adaptif bir yapıda oldukları için özellikle büyük ve karmaşık veri kümeleri üzerinde işlem yapmak için tercih edilmektedirler. Ayrıca, YSA'lar, insan beyninin işleyişini taklit ettiklerinden, eğitilme amacına göre insan benzeri öğrenme ve karar verme yeteneklerine de sahip olabilmektedir. YSA'larının eğitimleri sırasında genellikle belirli aşamalar bulunmaktadır. Bu adımlar sırasıyla aşağıdaki gibidir.

1. Veri toplama: YSA'nın eğitimi için kullanılacak verilerin toplanması ve düzenlenmesi.
2. Veri önişleme: Verilerin temizlenmesi, normalleştirilmesi, dönüştürülmesi veya düzenlenmesi gibi ön işleme adımlarının uygulanması.
3. Model tasarımı: Yapay sinir ağı mimarisinin tasarlanması, katmanların sayısı ve türü, aktivasyon fonksiyonları gibi parametrelerin seçilmesi.
4. Model eğitimi: Verilerin yapay sinir ağına verilerek ağırlık ve bias değerlerinin hesaplanması ve bu değerlerin hata fonksiyonuna göre güncellenmesi işlemi.

5. Model doğrulama: Eğitilmiş yapay sinir ağı modelinin, ayrı bir doğrulama veri kümesinde test edilmesi ve performansının ölçülmesi.
6. Model ayarlaması: Model performansının doğrulanmasının ardından, modeldeki bazı parametreler ile öğrenme oranı ve optimizasyon algoritması gibi bazı hiperparametrelerin ayarlanarak modelin yeniden eğitimi.
7. Model dağıtımı: Eğitilmiş yapay sinir ağı modelinin kullanıma hazır hale getirilmesi ve gerçek dünya uygulamalarında kullanılması.

Yukarıda bahsedilen bu adımlar eğitilmek istenilen modele göre tekrarlanabilir ve/veya modelin daha da iyileştirilmesi için özelleştirilebilir. YSA'ların; evrişimsel sinir ağı (CNN), yinelemeli sinir ağı (RNN), derin sinir ağı (DNN) ve uzun-kısa süreli bellekli sinir ağı (LSTM) gibi farklı tiplerde özelleştirilmiş modelleri de bulunmaktadır. Her bir tip, farklı görevler için tasarlanmıştır ve farklı yapısal özelliklere sahiptir. Bu farklı yapılar, farklı görevler için daha etkili olabilmektedir ve öğrenme süreci daha verimli hale getirilebilmektedir.

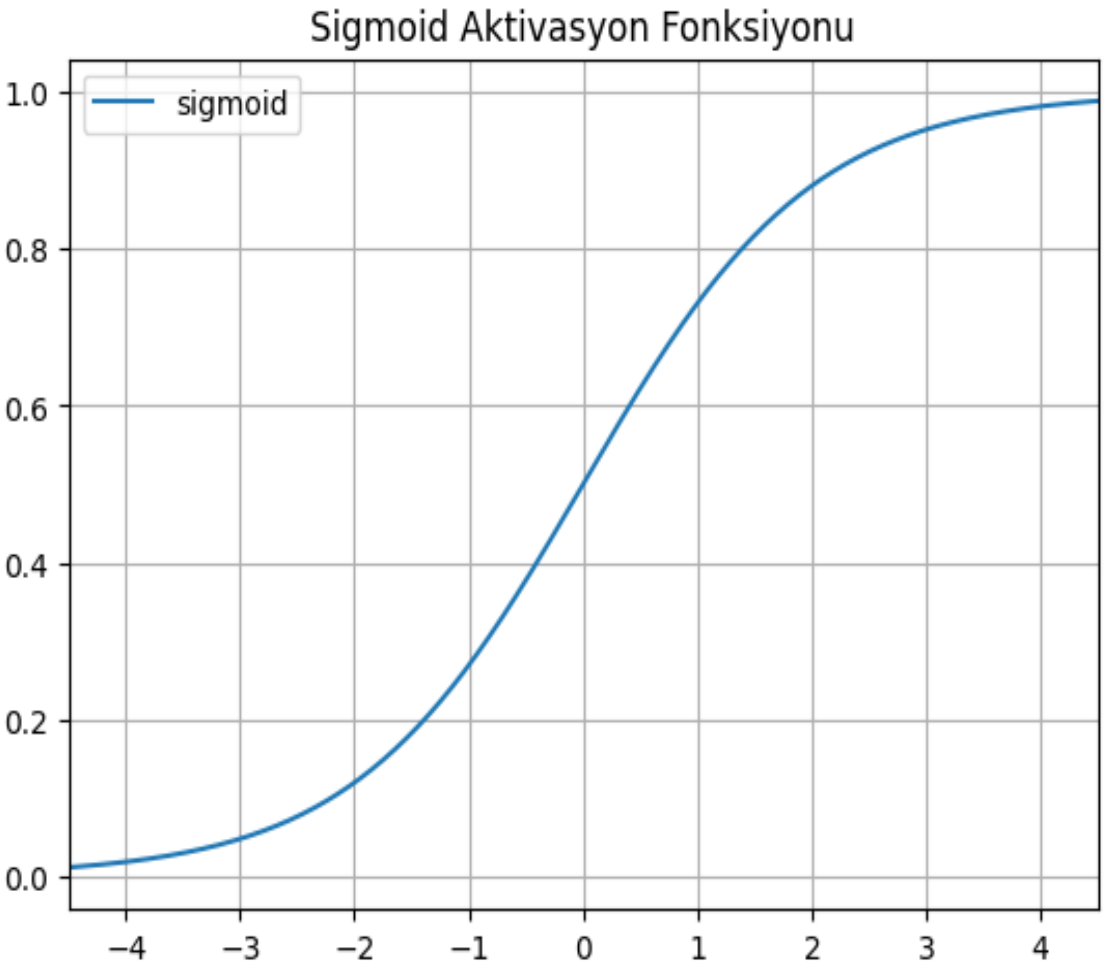
4.1.1.Aktivasyon Fonksiyonları

Aktivasyon fonksiyonları YSA'ların kullanım alanlarına ve problemlere göre değişiklik göstermektedir. Literatürde en yaygın kullanılanılan aktivasyon fonksiyonlarından bazıları Sigmoid, Hiperbolik Tanjant(tanh), Rectified Linear Unit (Relu) ve Softmax'tır (Sharma ve diğ., 2020).

4.1.1.1. Sigmoid Aktivasyon Fonksiyonu

YSA'da kullanılan aktivasyon fonksiyonları arasında en yaygın olarak kullanılan fonksiyondur. Bu fonksiyon giriş değeri olarak aldığı değeri 0 ile 1 arasında normalize etmektedir. Fonksiyon Denklem (4.3)'deki formül ile tanımlanmaktadır ve Şekil 4.3'de gösterilmektedir (Sharma ve diğ., 2020).

$$S(x)= \frac{1}{1+e^{-x}} \tag{4.3}$$

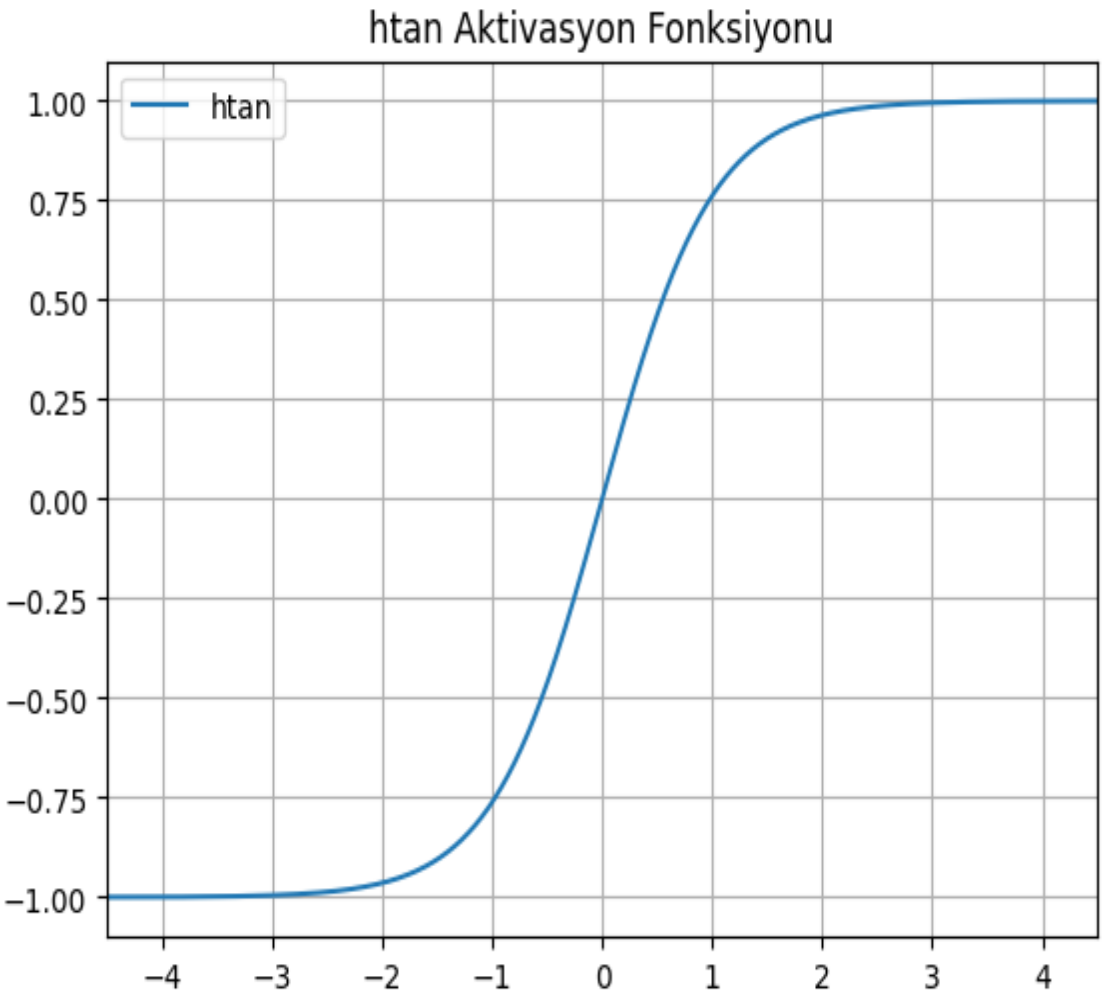


Şekil 4.3.Sigmoid fonksiyonu

4.1.1.2. Hiperbolik Tanjant Aktivasyon Fonksiyonu

Hiperbolik Tanjant (tanh) aktivasyon fonksiyonu da YSA'da yaygın olarak kullanılan aktivasyon fonksiyonlarındandır. Sigmoid aktivasyon fonksiyonuna benzer bir şekilde çalışmaktadır. Hiperbolik Tanjant fonksiyonunun Sigmoid aktivasyon fonksiyonundan farkı giriş değeri olarak aldığı değeri -1 ile 1 aralığında normalize etmesidir. Fonksiyon Denklem (4.4)'deki formül ile tanımlanmaktadır ve Şekil 4.4'te gösterilmektedir (Sharma ve diğ., 2020).

$$\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1} \tag{4.4}$$

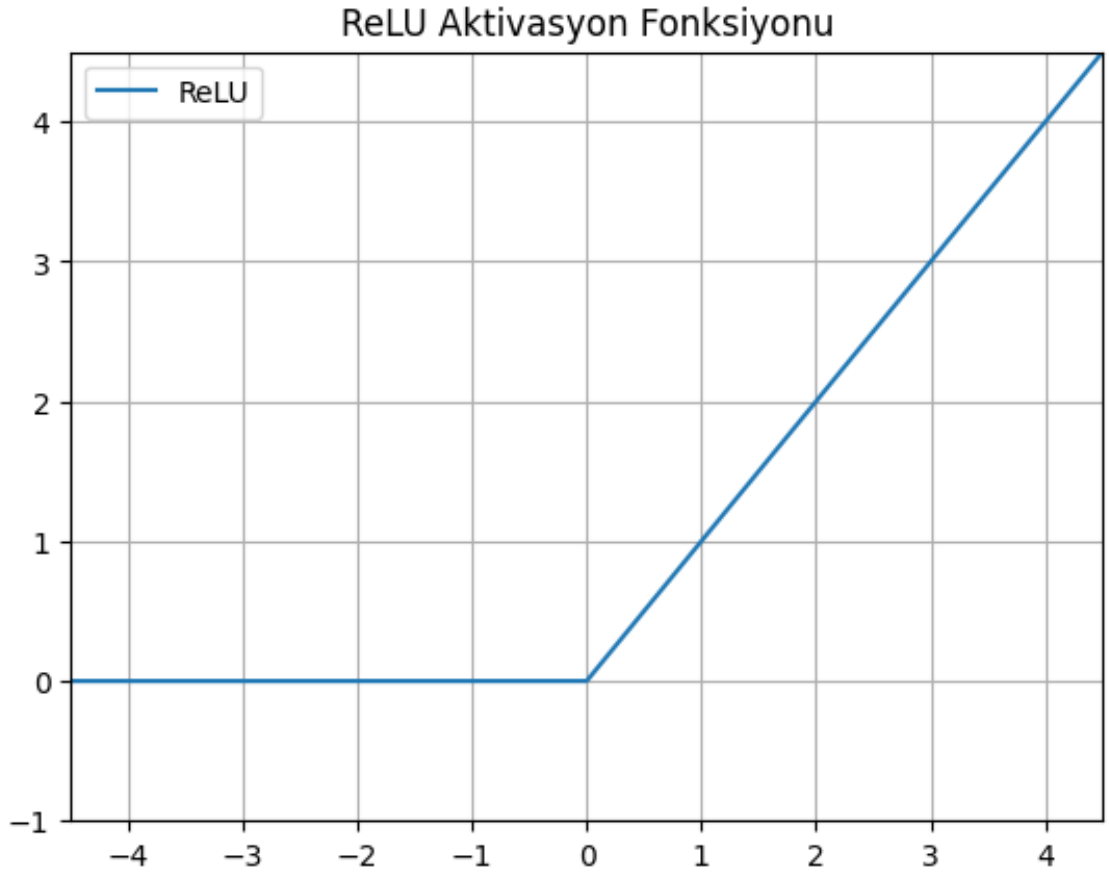


Şekil 4.4.Hiperbolik tanjant fonksiyonu (Sharma ve diğ., 2020)

4.1.1.3. ReLU (Rectified Linear Unit) Aktivasyon Fonksiyonu

Rectified Linear Unit (ReLU) aktivasyon fonksiyonu ağdaki tüm nöronların aktive olmasını engelleyerek hesaplama yükünü azaltmak ve ağın daha hızlı çalışmasını sağlamak gerektiğinde yaygın olarak kullanılmaktadır. ReLU fonksiyonu üretilen çıkış değerini 0 ile giriş değeri olarak verilen değer aralığında normalize etmektedir. Fonksiyon Denklem (4.5)'teki formül ile tanımlanmaktadır ve Şekil 4.5'te gösterilmektedir (Sharma ve diğ., 2020).

$$\text{Relu}\,(x) = \max(0,x) \tag{4.5}$$



Şekil 4.5.Relu fonksiyonu (Sharma ve diğ., 2020)

4.1.1.4. Softmax Aktivasyon Fonksiyonu

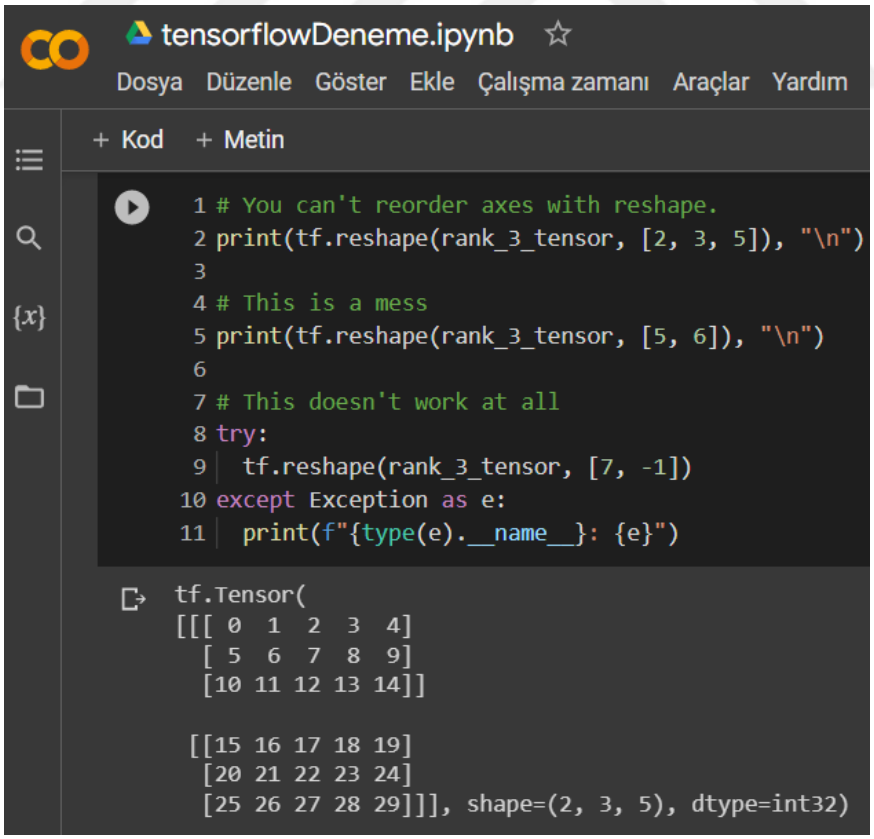
YSA'da ikiden fazla sınıflandırma yapılması gerekli olan problemlerde yaygın olarak kullanılmaktadır. Giriş değeri olarak almış olduğu değerin hangi sınıfa ait olduğunun olasılığını hesaplamaktadır. 0 ile 1 aralığında değer üretmektedir. Çıkış katmanı aktivasyon fonksiyonu olarak genellikle derin öğrenme modellerinde kullanılmaktadır. Fonksiyon Denklem (4.6)'daki formül ile tanımlanmaktadır (Sharma ve diğ., 2020).

$$\vartheta(\text{m. sınıf}) = \frac{e^{W_m A}}{\sum_{i=1}^{N} e^{W_i A}} \tag{4.6}$$

4.2.TensorFlow

TensorFlow (TF), Google tarafından Theano çerçevesinin yerine geliştirilmiş olan açık kaynaklı bir makine öğrenmesi ve derin öğrenme yazılımı kütüphanesidir. TF, yalnızca derin öğrenme değil aynı zamanda pekiştirmeli öğrenmeyi, yapay sinir ağları ve diğer

makine öğrenmesi modellerinin oluşturulması, eğitimi ve tahminlerinin yapılması için bir dizi araç ve kaynak sağlamaktadır. Bu modeller, görüntü işleme, doğal dil işleme, ses işleme, tahmin yapma ve diğer birçok görev için kullanılabilmektedir. Temelde TF, yapılacak olan hesaplamanın graflara ayrılarak yapılmasına dayanmaktadır. Bu graflar, bir veri akış grafiği olarak adlandırılmaktadır. Ayrıca modelin katmanları, işlevleri ve verileri arasındaki bağlantıları göstermektedir. TF, grafik hesaplama sayesinde, işlemleri verimli bir şekilde dağıtabilmektedir. Dahası GPU ve TPU gibi donanım hızlandırıcıları kullanarak işlemci gücünü artırabilmektedir. Çoğu makine öğrenmesi çerçevesi gibi, daha hızlı çalışmasını sağlayan bir C/C++ motoru üzerinden, bir Python API ile geliştirilmiştir. Bunun yanı sıra, Python, C++, Java ve Go gibi birçok programlama diliyle ve Google Colab gibi bulut tabanlı geliştirme ortamlarında da kullanılabilmektedir. Java API'si için deneysel destek bulunmasına rağmen, mevcut sürümlerde kararlı kabul edilmemektedir. Bu nedenle mevcut sürümleri Java ve Scala geliştiricileri için pek uygun olmayabilir (URL-18). Şekil 4.6'da Google Colab üzerinde çalışan ve tensörleri yeniden şekillendiren TensorFlow kodu örneği gösterilmektedir.



Şekil 4.6.Google Colab üzerinde Python dili kullanarak tensörleri yeniden şekillendiren kod örneği

4.3.Yapay Sinir Ağı Modeli Eğitimi

Bu tez çalışması kapsamında geliştirilmiş olan matematiksel gezinge indirgeme yöntemi Microsoft GeoLife veri kümesindeki bazı dosyalar üzerinde koşularak bir veri kümesi oluşturulmuştur. Literatürde eksik olduğu tespit edilen ve bir yapay zekâ yöntemi kullanılarak GNSS verisi indirgemesi yapabilmek amacıyla, bu veri kümesi kullanılarak bir YSA modeli eğitimi gerçekleştirilmiştir. Eğitilen bu YSA modeli kullanılarak mobil uygulama üzerinde verinin elde edildiği anda çevrim içi olarak GNSS verisi indirgemesi yapılabilmesi sağlanmıştır.

4.3.1.Kullanılan Veri Kümesi

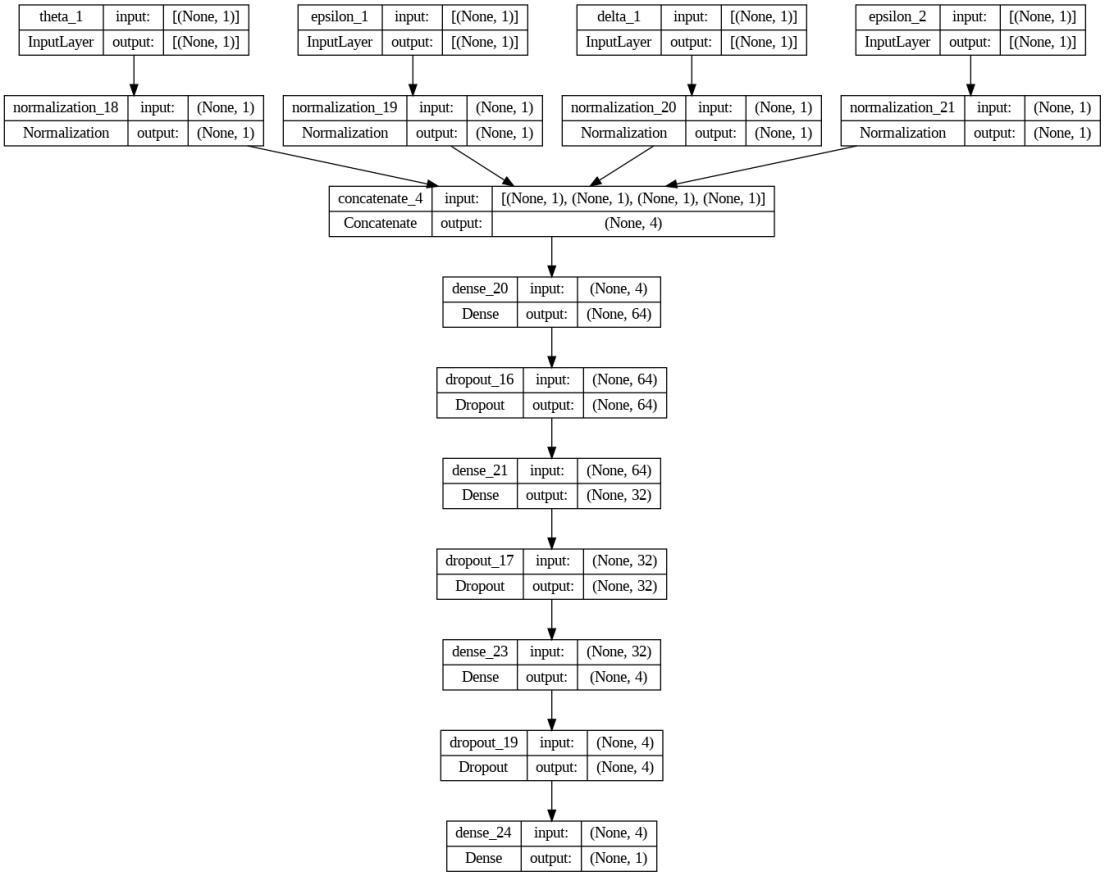
Eğitim için kullanılan veri kümesinin oluşturulması amacıyla Microsoft GeoLife veri kümesi içerisindeki farklı dosyalardan üretilmiş olan 14022 tane konum verisi kullanılmıştır. Veri kümesi oluşturulurken geliştirilmiş olan matematiksel indirgeme yöntemi θ = 8° - 10° arasında değişecek biçimde ve pencereleme boyutu 3 olacak şekilde koşulmuştur. Hassasiyetin artırılması amacıyla θ değeri belirli bir aralıkta değiştirilmektedir. Pencereleme boyutunun 3 olarak seçilmesinin nedeni ise, mobil cihaz üzerinde çevrim içi hesaplama yapılacağı düşünüldüğünden, modelin çalışması sırasındaki performansını artırmak içindir. Pencereleme boyutu değiştiğinde yöntemin ek hesaplamalar yapması gerekeceğinden, bu çalışmada 3 hücreli pencereleme kullanılmıştır.

4.3.2.Model Eğitimi

Modelin eğitimi, İnternet'te geliştirme ve yürütme kolaylığı sağladığı için Google Colab üzerinde Python dilinde yapılmıştır. Eğitim işlemi; TF, Keras, Numpy ve Pandas kütüphaneleri kullanılarak gerçekleştirilmiştir.

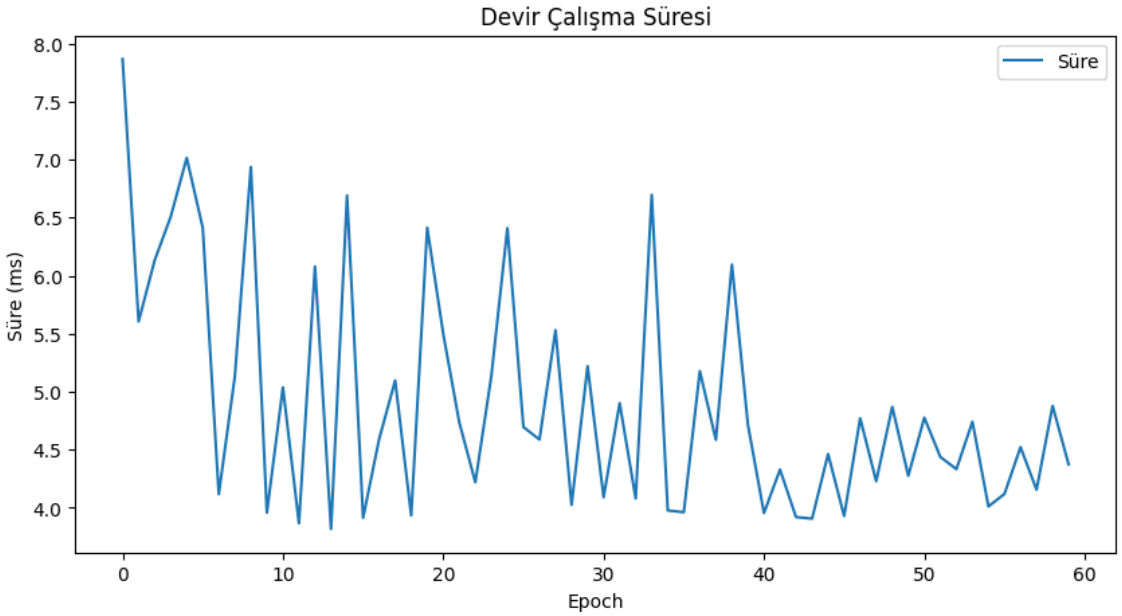
Modelin eğitimi için, ilk olarak gerekli olan kütüphaneler çalışma ortamına alınır. Ardından, eğitim için kullanılacak olan veri kümesi çalışma ortamına yüklenerek dosya işlenir. Bu aşamada, dosyada bulunan 14022 tane konum verisine ait $\theta(P_1P_3)$ (theta_1) ve $\Delta(P_2P_3)$ (delta_1) açıları ile ε_1 (epsilon_1) ve ε_2 (epsilon_2) eşik değerleri ve dönüş noktası olma durumları okunmuştur. Okunan tüm bu veriler içerisinden 12620 tanesi yani %90'ı eğitim, 1402 tanesi yani %10'u ise test için veri çerçevelerine ayırılmıştır. Bu veri

çerçeveleri uygun veri kümeleri haline dönüştürülerek 8'erli gruplara ayrılmıştır. Bu girdilerin tamamı eğitimin başarımını artırmak için normalize edilmiştir ve giriş katmanına uygulanacak biçimde bir dizi haline getirilmiştir. Eğitilen model 4 katmadan meydana gelen ve 60 döngü boyunca eğitimi gerçekleştirilmiş ileri beslemeli yapay sinir ağıdır. Çıkış katmanı hariç aktivasyon fonksiyonu olarak bu modelde daha iyi sonuçlar verdiğinden ReLU kullanılmış ve her bir katmanda öğrenmeyi tamamlayamadığı için atılma oranı 0,5 olarak seçilmiştir. Çıkış katmanında ise sigmoid aktivasyon fonksiyonu kullanılmıştır. Yoğunluk; ilk katmanda 64, ikinci katmanda 32, üçüncü katmanda 16, dördündü katmanda 4, çıkış katmanında ise 1 olarak belirlenmiştir. Modelin derlemesi sırasında ağırlıklandırma için az bellek gereksinimi bulunan, hesaplama açısından verimli olan, gradyanların ve kare gradyanların üstel hareketli ortalamasını hesaplayan *adam* (Kingma & Ba, 2014) algoritması kullanılmıştır. Hata fonksiyonu olarak da MSE kullanılmıştır. Modelin iyileşmesini sağlamak amacıyla doğrulama kaybı monitörize edilmiş ve en iyileşen eğitimlerin sonuçları saklanmıştır. Oluşturulan YSA modeli Şekil 4.7'de gösterilmektedir.

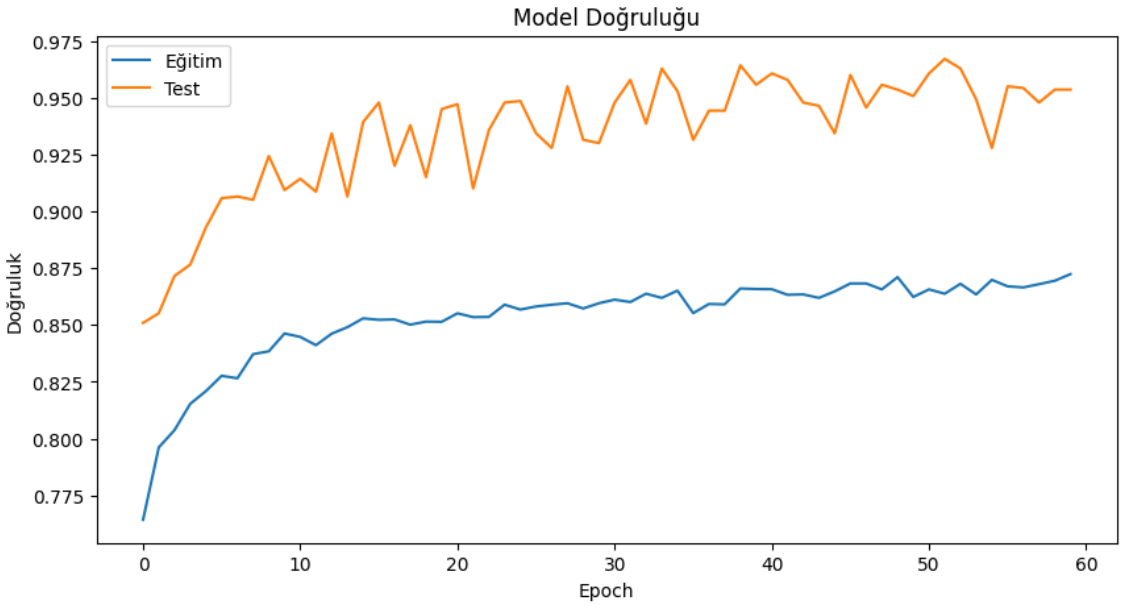


Şekil 4.7.Eğitilen ağın modeli

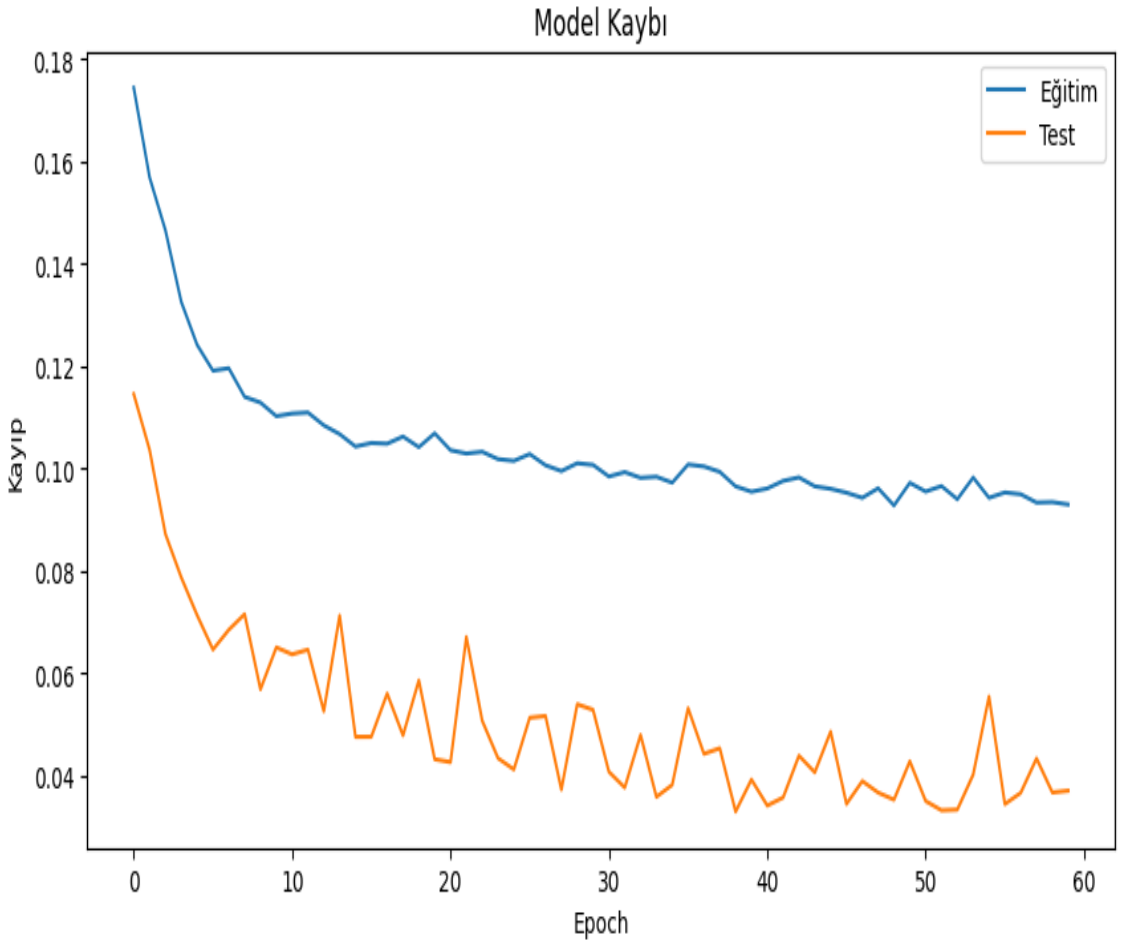
Modelin eğitimi 60 devir boyunca yapılmıştır. Eğitim sonucunda elde edilen ortalama doğruluk değeri 0,8531, ortalama kayıp değeri 0,1049, ortalama geçerleme doğruluğu değeri 0,9347, ortalama geçerleme kaybı değeri 0,0503, her bir devrin ortalama çalışma süresi ise 4,9081 ms olmuştur. Modelin eğitimi sırasındaki her bir devire ait çalışma süreleri Şekil 4.8'deki grafikte, eğitim ve test doğruluk değerleri Şekil 4.9'daki grafikte, eğitim ve test kayıp değerleri Şekil 4.10'daki grafikte gösterilmektedir.



Şekil 4.8.Eğitilen modelin her bir devir için çalışma süresi grafiği



Şekil 4.9.Eğitilen modelin doğruluk grafiği



Şekil 4.10.Eğitilen modelin kayıp grafiği

Modelin eğitimi tamamlandıktan sonra mobil uygulamada kullanılabilmesi amacıyla TensorFlow Lite (TFLite) modeline dönüştürülerek ".tflite" uzantılı dosyaya yazılmıştır. Bu dosya geliştirilmiş olan mobil uygulamaya eklenerek model yüklenmiş ve uygulama üzerinde koşulmuştur.

4.4.Performans Değerlendirmesi

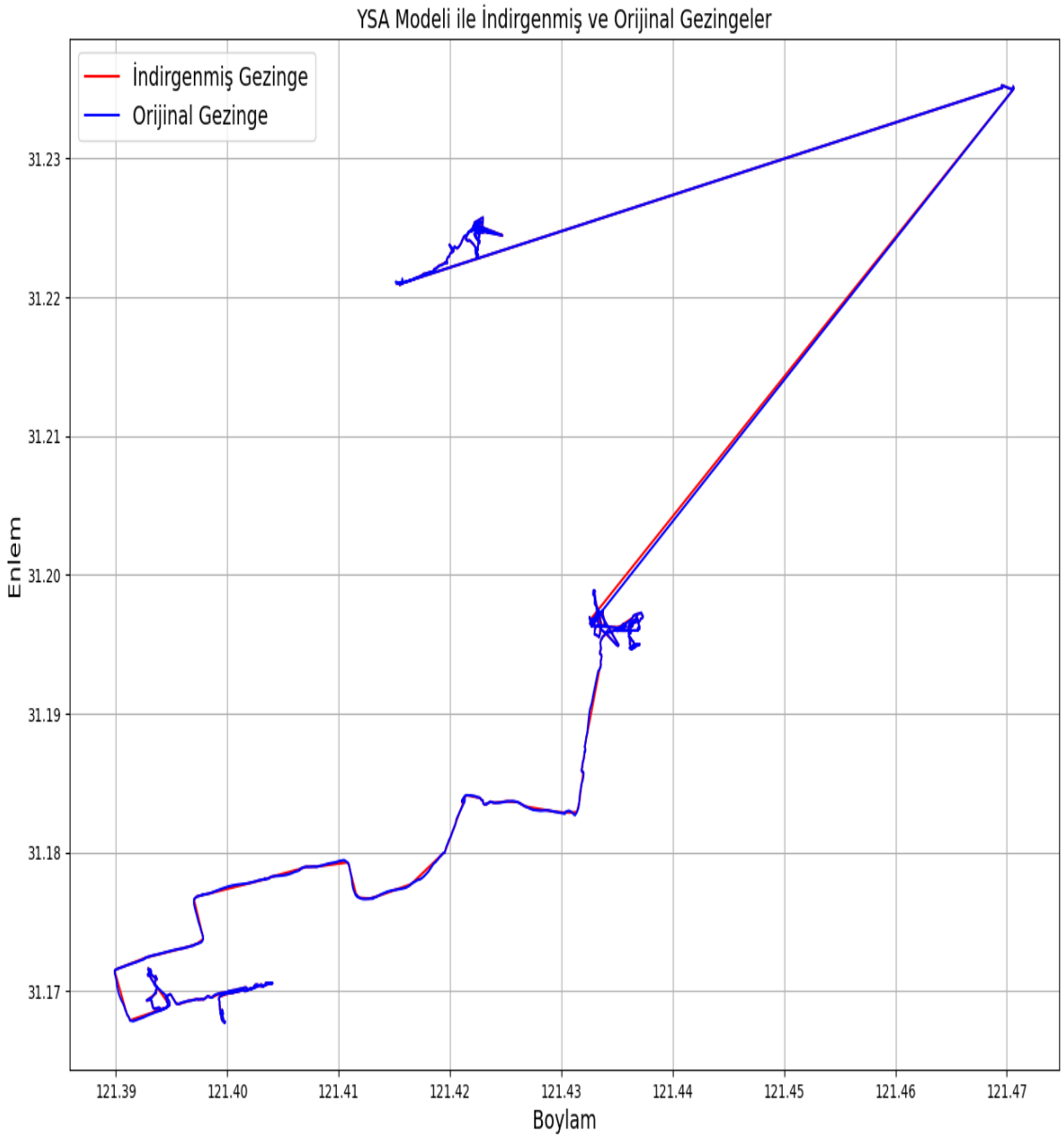
Bu tez çalışması kapsamında eğitilmiş olan YSA modelinin performansını değerlendirebilmek amacıyla bazı denemeler yapılmıştır. Testler Microsoft GeoLife veri kümesindeki bazı dosyalar üzerinde gerçekleştirilmiştir. Elde edilen sonuçların anlamlandırılabilmesi amacıyla YSA modeli ve bu tez çalışması kapsamında geliştirilen yöntemin sonuçları karşılaştırılmaktadır.

İlk örnekte Microsoft GeoLife veri kümesindeki “20090522094234.plt” dosyası üzerinde denemeler yapılmıştır. Test sonuçları Tablo 4.1’de gösterilmektedir.

Tablo 4.1.YSA modeli ve geliştirilen yöntemin sonuçlarının “20090522094234.plt” dosyası üzerinden kıyaslanması

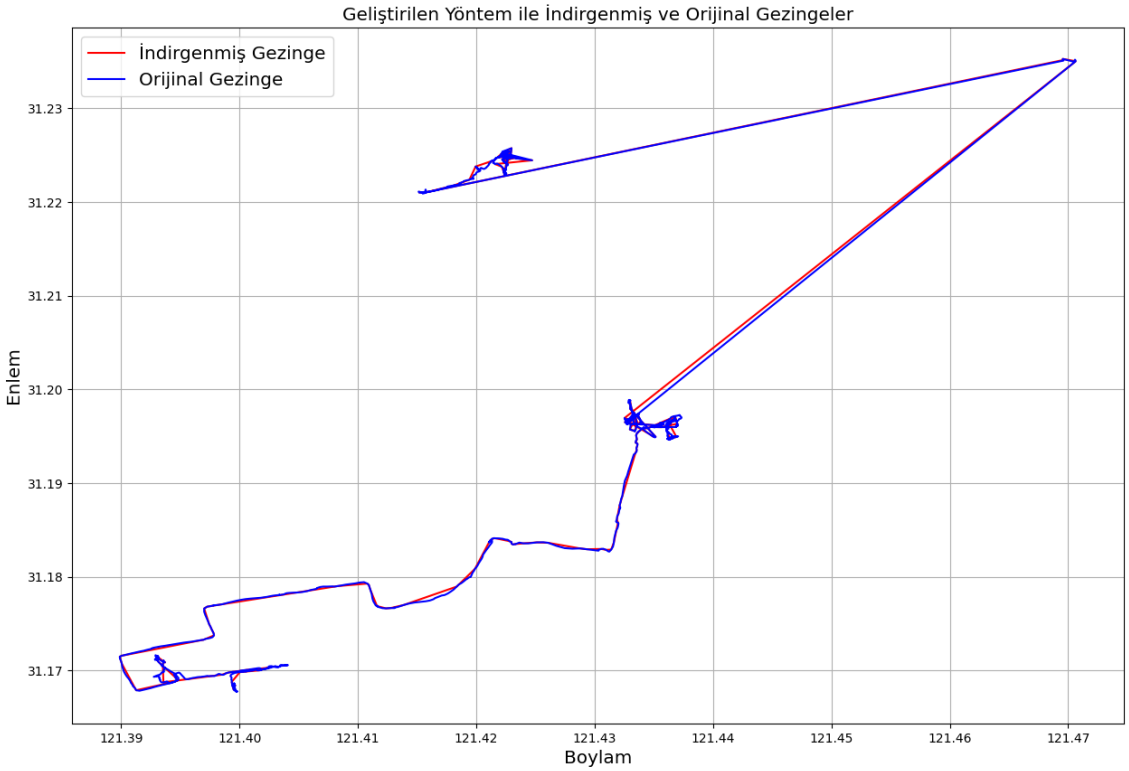
	YSA Modeli	**Önerilen Yöntem**
Konum Sayısı	2560	2560
İndirgeme Sonrası Konum Sayısı	1494	170
İndirgeme Oranı	%41,64	%93,36
Hesaplama Süresi (ms)	20,89	14,41
RMSE Değeri	8,30117e-4	1,44114e -03
Eşik Değer	$\theta > 8° \ \& \ \theta < 10°$	$\theta = 5°$
Pencereleme Boyutu	3	3

Bu örnekte eğitimi gerçekleştirilen YSA modeli Microsoft GeoLife veri kümesindeki “20090522094234.plt” dosyasına uygulanmıştır. Bu dosyanın içerisinde 2560 adet koordinat bulunmaktadır. Bu Tablo 4.1’de listelenen girdiler kullanılarak yapılan deneme sonucunda dosya içerisinde bulunan 2560 adet noktadan 1066 adet nokta indirgenmiştir ve indirgeme sonucunda aynı dosya 1494 adet nokta ile temsil edilmiştir. Bu işlem sonucunda %41,64 oranında nokta indirgemesi sağlanmıştır. Ayrıca indirgenmiş gezingeye ait RMSE değerinin 8,30117e-4 olduğu görülmüştür. Bunun yanı sıra yöntemin hesaplama süresinin ise 20,89 ms olduğu gözlemlenmiştir. Microsoft GeoLife veri kümesindeki “20090522094234.plt” dosyasındaki orijinal gezinge ve yöntemin çalışması sonucunda elde edilen indirgenmiş gezingenin görsel açıdan kıyaslanabilmesi amacıyla bir arada gösterimi Şekil 4.11’deki gibidir.



Şekil 4.11.YSA modelinin sonuçlarının "20090522094234.plt" dosyası üzerinden görsel kıyaslaması

Geliştirilen matematiksel yöntem aynı veri kümesi üzerinde pencereleme boyutu 3 ve eşik değeri 5° seçilerek koşulmuştur. Bu girdiler kullanılarak yapılan deneme sonucunda ise 2390 adet nokta indirgenerek aynı dosya 170 nokta ile temsil edilmiştir. Bu işlem sonucunda %93,36 oranında indirgeme sağlanmıştır. Ancak RMSE değerinin 1,44114e-03, hesaplama süresinin ise 14,41 ms olduğu gözlemlenmiştir. Orijinal gezinge ve yöntemin çalışması sonucunda elde edilen indirgenmiş gezingenin görsel açıdan kıyaslanabilmesi amacıyla bir arada gösterimi Şekil 4.12'deki gibidir.



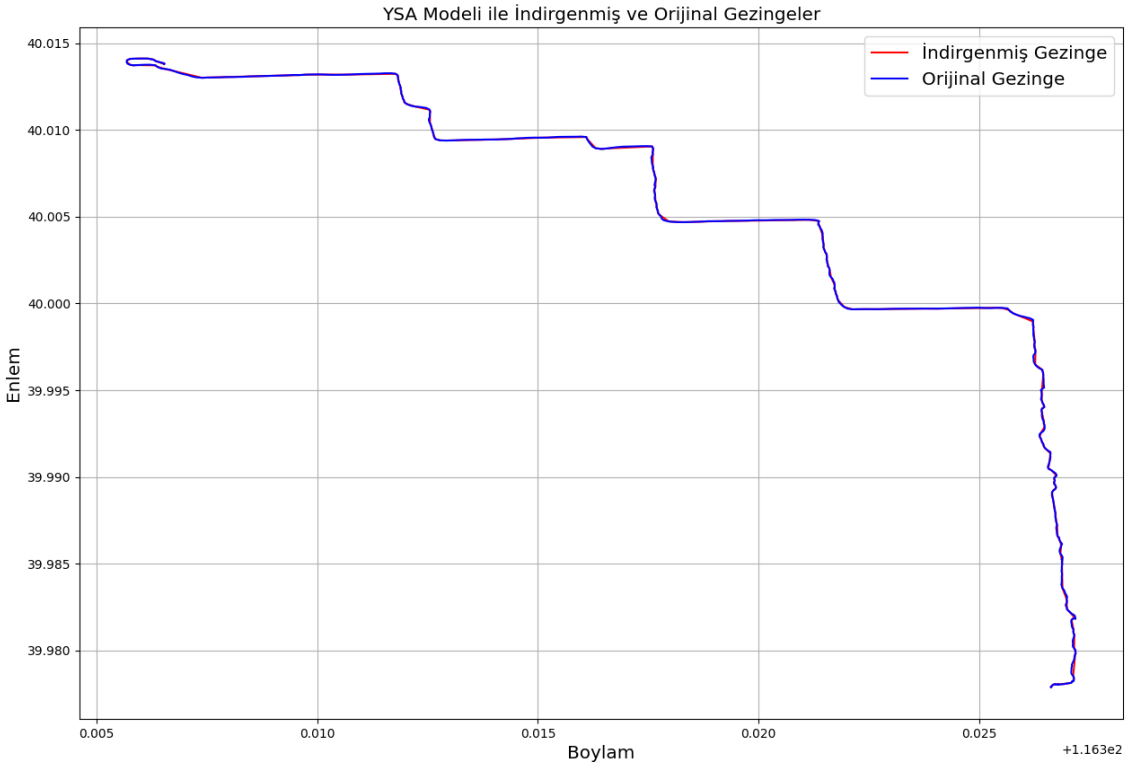
Şekil 4.12.Önerilen yöntemin sonuçlarının "20090522094234.plt" dosyası üzerinden görsel kıyaslaması

İkinci örnekte Microsoft GeoLife veri kümesindeki "20081214235553.plt" dosyası üzerinde denemeler yapılmıştır. Test sonuçları Tablo 4.2'de gösterilmektedir.

Tablo 4.2.YSA modeli ve geliştirilen yöntemin sonuçlarının "20081214235553.plt" dosyası üzerinden kıyaslanması

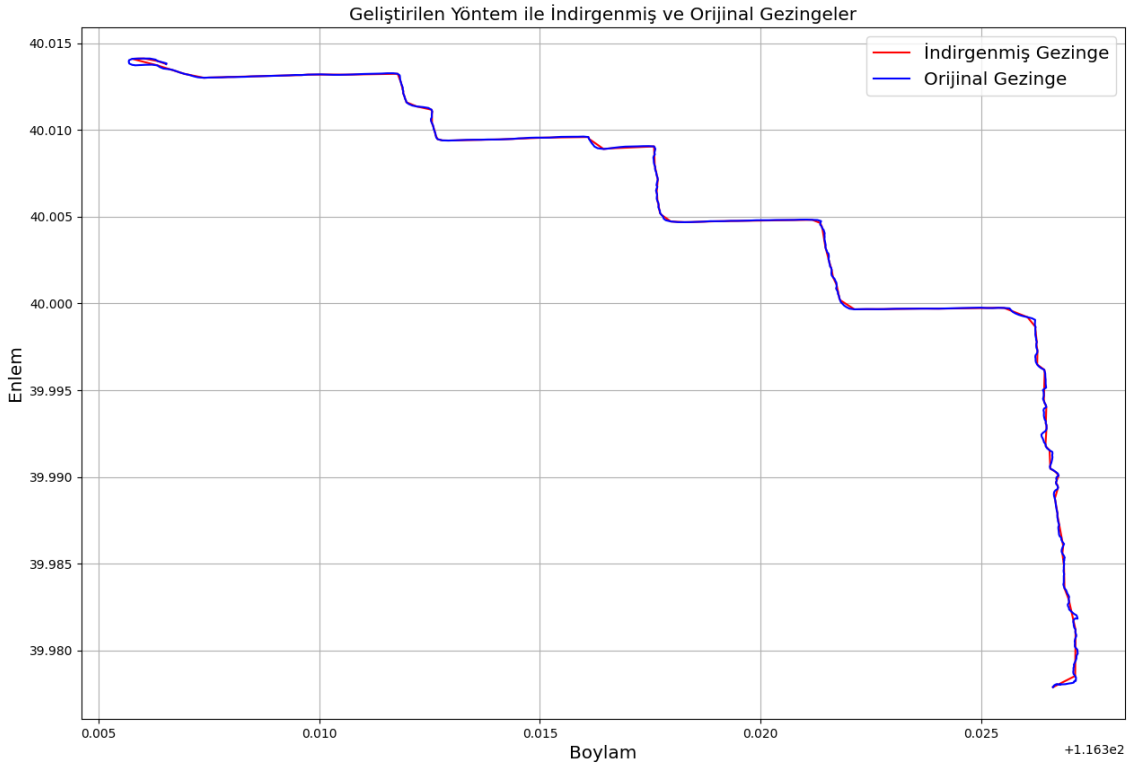
	YSA Modeli	**Önerilen Yöntem**
Konum Sayısı	539	539
İndirgeme Sonrası Konum Sayısı	220	113
İndirgeme Oranı	%59,18	%79,04
Hesaplama Süresi (ms)	42,17	10,4
RMSE Değeri	1,50339e-05	5.08088e-05
Eşik Değer	-	$\theta = 20°$
Pencereleme Boyutu	-	3

Bu örnekte eğitimi gerçekleştirilen YSA modeli Microsoft GeoLife veri kümesindeki “20081214235553.plt” dosyasına uygulanmıştır. Bu dosyanın içerisinde 539 adet koordinat bulunmaktadır. Bu girdiler kullanılarak yapılan deneme sonucunda 319 adet nokta indirgenerek aynı dosya 220 adet nokta ile temsil edilmiştir. Bu işlem sonucunda %59,18 oranında indirgeme sağlanmıştır. Ayrıca RMSE değerinin 1,50339e-05, hesaplama süresinin ise 42,17 ms olduğu gözlemlenmiştir. Orijinal gezinge ve yöntemin çalışması sonucunda elde edilen indirgenmiş gezingenin görsel açıdan kıyaslanabilmesi amacıyla bir arada gösterimi Şekil 4.13’teki gibidir.



Şekil 4.13. YSA modelinin sonuçlarının “20081214235553.plt” dosyası üzerinden görsel kıyaslaması

Geliştirilen matematiksel yöntem aynı veri kümesi üzerinde pencereleme boyutu 3 ve eşik değeri 20° seçilerek koşulmuştur. Bu girdiler kullanılarak yapılan deneme sonucunda ise 539 adet nokta indirgenerek aynı dosya 113 nokta ile temsil edilmiştir. Bu işlem sonucunda %79,04 oranında indirgeme sağlanmıştır. Ancak RMSE değerinin 5.08088e-05, hesaplama süresinin ise 10,4 ms olduğu gözlemlenmiştir. Orijinal gezinge ve yöntemin çalışması sonucunda elde edilen indirgenmiş gezingenin görsel açıdan kıyaslanabilmesi amacıyla bir arada gösterimi Şekil 4.14’teki gibidir.



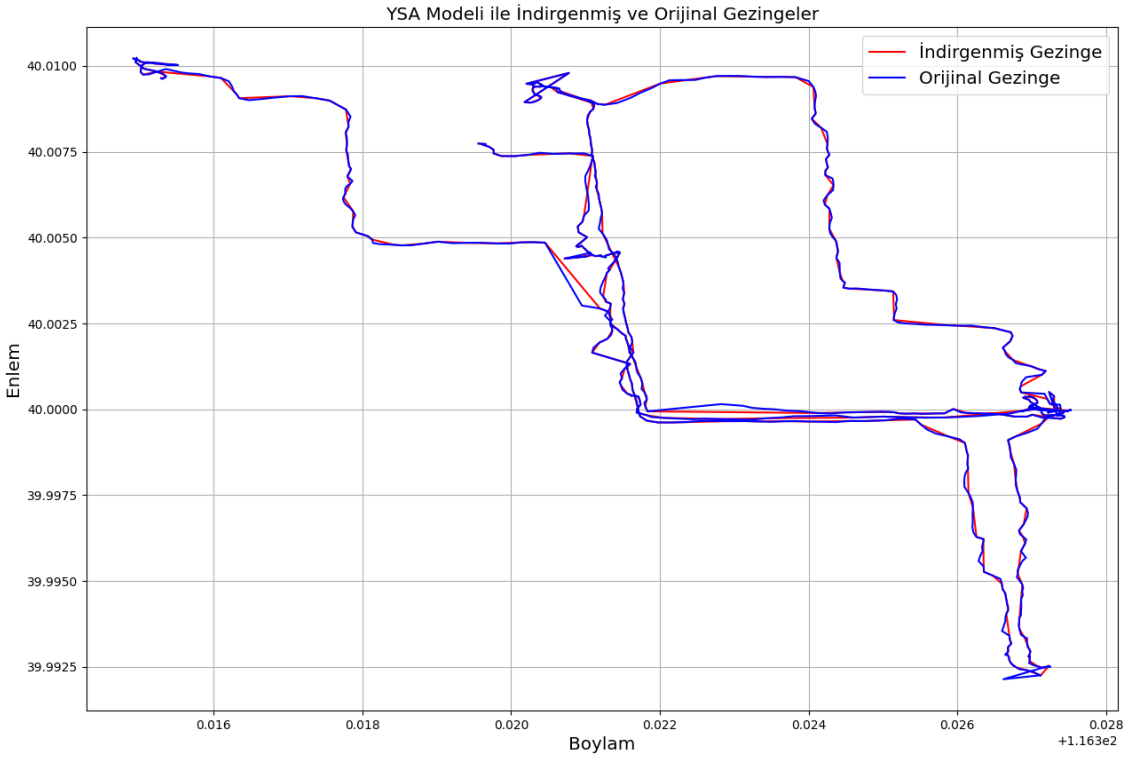
Şekil 4.14. Önerilen yöntemin sonuçlarının "20081214235553.plt" dosyası üzerinden görsel kıyaslaması

Üçüncü örnekte ise Microsoft GeoLife veri kümesindeki "20090608060034.plt" dosyası üzerinde denemeler yapılmıştır. Test sonuçları Tablo 4.3'te gösterilmektedir.

Tablo 4.3.YSA modeli ve geliştirilen yöntemin sonuçlarının "20090608060034.plt" dosyası üzerinden kıyaslanması

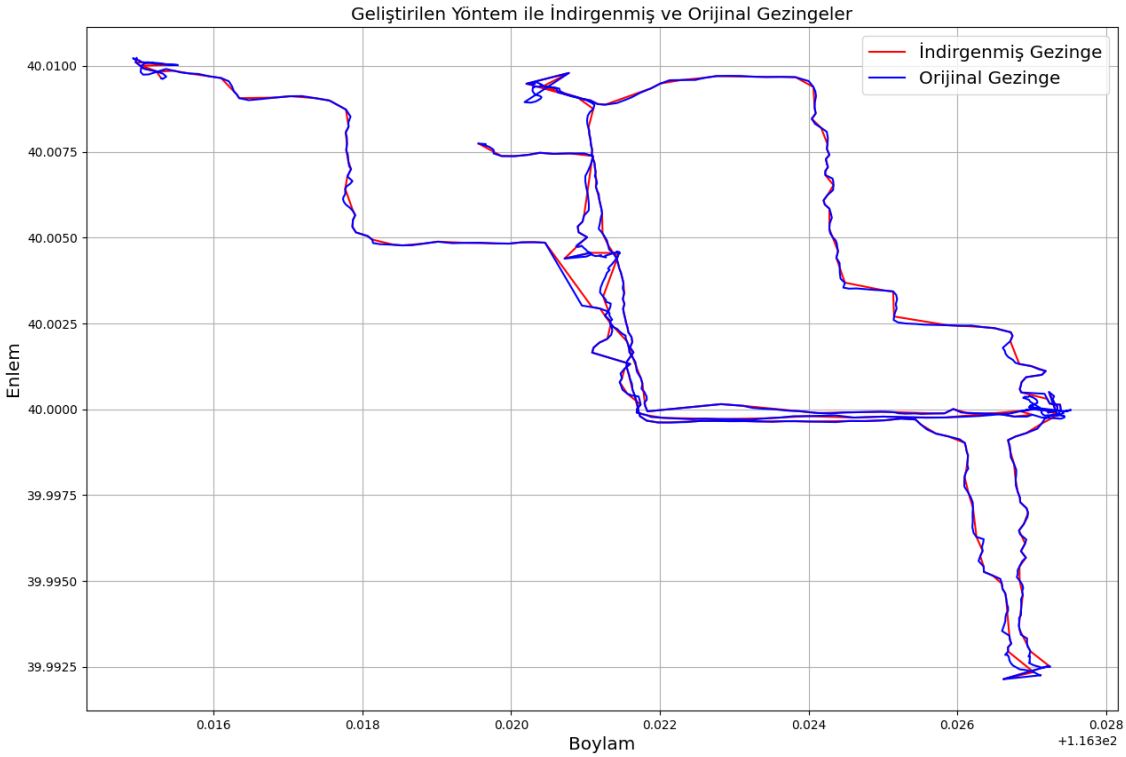
	YSA Modeli	**Önerilen Yöntem**
Konum Sayısı	875	875
İndirgeme Sonrası Konum Sayısı	515	290
İndirgeme Oranı	%41,14	%66,86
Hesaplama Süresi (ms)	70,89	41,43
RMSE Değeri	5.87826e-05	1.11122e-04
Eşik Değer	-	$\theta = 10°$
Pencereleme Boyutu	-	3

Bu örnekte eğitimi gerçekleştirilen YSA modeli Microsoft GeoLife veri kümesindeki “20090608060034.plt” dosyasına uygulanmıştır. Bu dosyanın içerisinde 875 adet koordinat bulunmaktadır. Bu girdiler kullanılarak yapılan deneme sonucunda 360 adet nokta indirgenerek aynı dosya 515 adet nokta ile temsil edilmiştir. Bu işlem sonucunda %41,14 oranında indirgeme sağlanmıştır. Ayrıca RMSE değerinin 5.87826e-05, hesaplama süresinin ise 70,89 ms olduğu gözlemlenmiştir. Orijinal gezinge ve yöntemin çalışması sonucunda elde edilen indirgenmiş gezingenin görsel açıdan kıyaslanabilmesi amacıyla bir arada gösterimi Şekil 4.15‘teki gibidir.



Şekil 4.15. YSA modelinin sonuçlarının “20090608060034.plt” dosyası üzerinden görsel kıyaslaması

Geliştirilen matematiksel yöntem aynı veri kümesi üzerinde pencereleme boyutu 3 ve eşik değeri 10° seçilerek koşulmuştur. Bu girdiler kullanılarak yapılan deneme sonucunda ise 875 adet nokta indirgenerek aynı dosya 290 nokta ile temsil edilmiştir. Bu işlem sonucunda %66,86 oranında indirgeme sağlanmıştır. Ancak RMSE değerinin 1.11122e-04, hesaplama süresinin ise 41,43 ms olduğu gözlemlenmiştir. Orijinal gezinge ve yöntemin çalışması sonucunda elde edilen indirgenmiş gezingenin görsel açıdan kıyaslanabilmesi amacıyla bir arada gösterimi Şekil 4.16‘deki gibidir.

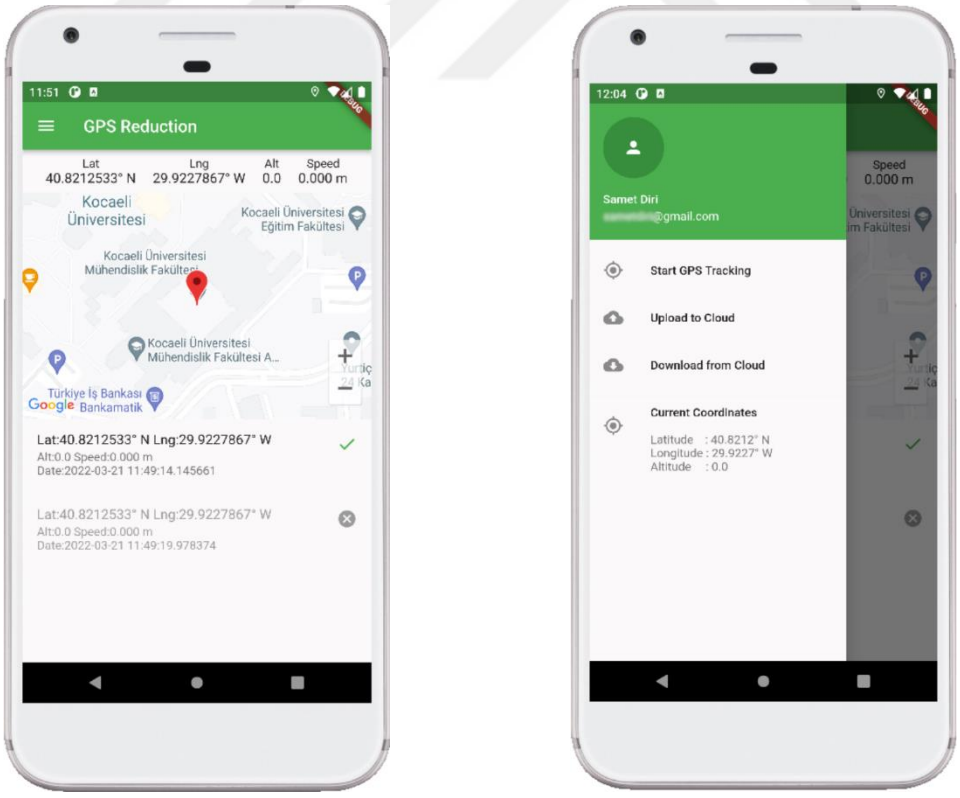


Şekil 4.16. Önerilen yöntemin sonuçlarının “20090608060034.plt” dosyası üzerinden görsel kıyaslaması

5.GELİŞTİRİLEN MOBİL UYGULAMA

5.1.Mobil Uygulamada Kullanılan Teknolojiler

Bu tez çalışması kapsamında hareketli nesnelerin konum verilerinin indirgenmesi için geliştirilen yöntemin mobil cihazlarda da çevrim içi bir şekilde çalıştığını ispatlayabilmek, test edebilmek ve görselleştirebilmek amacıyla bir mobil uygulama geliştirilmiştir. Uygulama; Android Studio v.2020.3.1 (URL-19) IDE'si, Dart v.211.7808 (URL-20) dili ve Flutter v.65.2.2 (URL-21) SDK'sı kullanılarak geliştirilmiştir. Arayüzleri Şekil 5.1'de gösterilen bu uygulama Samsung Galaxy M51 (SM-M515F) Android 11 işletim sistemli mobil cihazı üzerinde gerçek ortamlarda test edilmiştir. Uygulama kullanılarak indirgenen noktaların saklanması ve işlenebilmesi amacıyla Google'a ait Firebase platformundaki gerçek zamanlı bulut veri tabanı olan "Realtime Database" kullanılmıştır (Fitri ve diğ., 2022; Khawas & Shah, 2018; Reis ve diğ., 2022). Bulut veri tabanı sayesinde hareketli nesnelerden gerçek zamanlı gezinge akışı sağlanmaktadır.

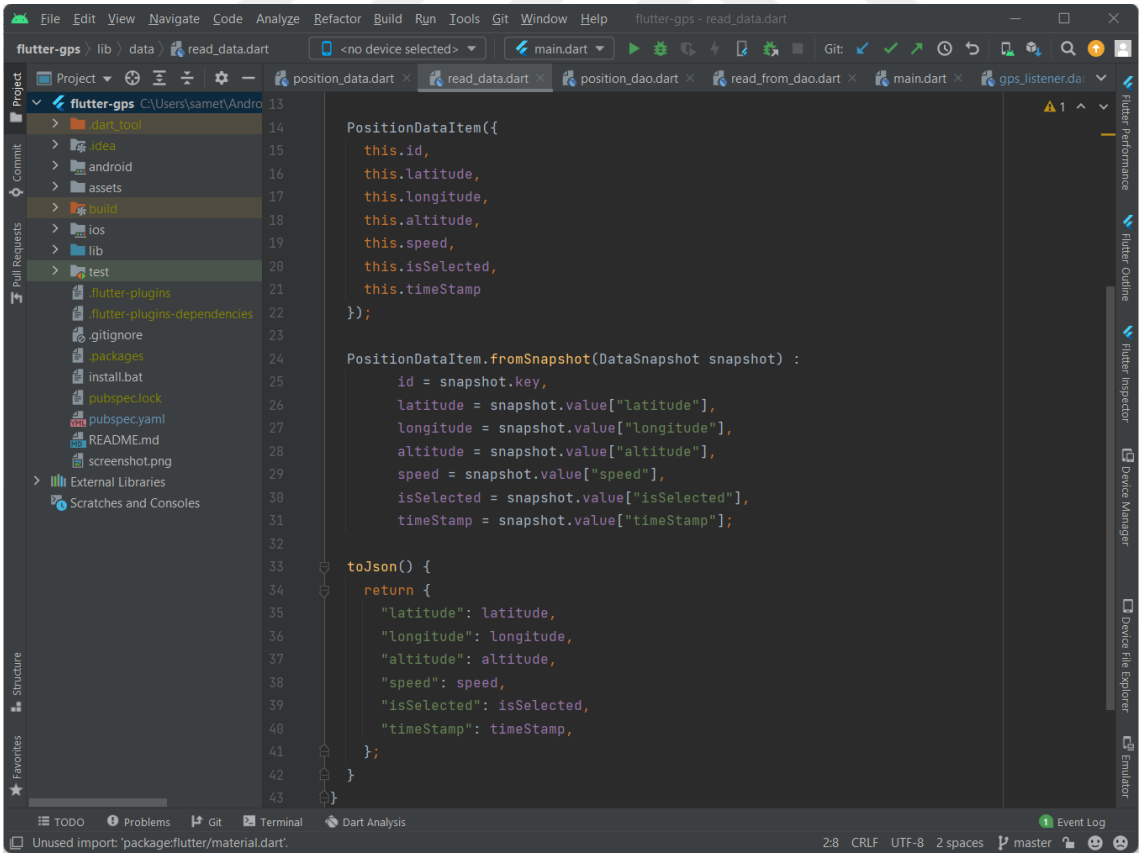


(a) (b)

Şekil 5.1. (a) Geliştirilen mobil uygulamanın ana ekranı, (b) geliştirilen mobil uygulamanın ana menüsü

5.1.1.Android Studio

Android Studio, Google tarafından Java, Kotlin, Dart veya C/C++ programlama dillerini kullanarak Android uygulamaları geliştirmek için sağlanan resmi bir entegre geliştirme ortamıdır (ing: Integreted Development Enviroment - IDE). Android Studio, kullanıcı arayüzü tasarımı, kodlama, derleme, hata ayıklama, test etme ve diğer birçok geliştirme işlemini tek bir arayüzde birleştirmektedir. Ayrıca zengin bir kullanıcı arayüzü tasarım aracına ve hata ayıklama özelliklerine sahiptir. Bunların yanı sıra, hata ayıklama işlemini kolaylaştırmak amacıyla, kodda var olan hataları bulmak ve düzeltmek için hata ayıklama araçlarına sahiptir. Android Studio, birçok farklı Android cihazı ve sürümü için geliştirilen uygulamaların farklı cihazlarda doğru şekilde çalıştığını doğrulamaya yardımcı olması amacıyla kendi içerisinde birkaç farklı test çerçevesi barındırmaktadır. Bu, uygulamaların yayınlanmadan önce hataların ve sorunların tespit edilmesine olanak sağlamaktadır (URL-19). Android Studio'ya ait ekran görüntüsü Şekil 5.2'de gösterilmektedir.



Şekil 5.2. Android Studio ekran örneği

5.1.2.Dart Dili ve Flutter Çerçevesi

Dart; Google tarafından geliştirilen, özellikle web ve mobil uygulama geliştirme için basit, modern ve güçlü bir dil olarak tasarlanmış, açık kaynaklı hem derlenebilen hem de yorumlanabilen bir programlama dilidir. JavaScript'e benzer bir sözdizimine sahip olduğundan yazılım geliştiriciler tarafından kolayca öğrenilebilmektedir. Ayrıca yüksek performanslı, hafif, güvenli ve geniş kütüphane yelpazesine sahip bir dil olarak öne çıkmaktadır. Bunların yanı sıra, JavaScript'ten farklı olarak Dart, veri tipleri ve statik tipleme özelliği olan bir dildir. Dart, ayrıca boş güvenliği (ing: null safety) özelliği barındırmaktadır. Boş güvenliği, geliştirici tarafından değişkenlerin ve/veya nesnelerin değerlerinin boş olabileceğini belirtmedikçe boş olmayacağıdır. Bu özellikler hata ayıklamaya, kod kalitesini artırmaya ve çalışma zamanında meydana gelebilecek olan boş istisnalarından korunmaya yardımcı olmaktadır. Dart dili, özellikle Flutter çerçevesi için geliştirilmiştir. (URL-20). Dart dili kullanılarak mobil cihazın konum bilgisine ait ilk durum ataması yapan kod örneği Şekil 5.3'te gösterilmiştir.

```
@override
void initState() {
  _positionStream = _geolocator.getPositionStream(locationOptions);
  _positionStreamSubscription = _positionStream.listen((Position position) {
    setState(() {
      double latitude = position.latitude;
      double longitude = position.longitude;
      double altitude = position.altitude;
      double speed = position.speed;

      latitudeAsString = ((latitude.abs() * sensitivity).floor() / sensitivity).toString() +
(latitude > 0 ? '° N' : '° S');
      longitudeAsString = ((longitude.abs() * sensitivity).floor() / sensitivity).toString()
+ (longitude > 180 ? '° E' : '° W');
      altitudeAsString = ((altitude.abs() * sensitivity).floor() / sensitivity).toString() +
(altitude > 0 ? ' m' : '');
      speedAsString = (speed).toStringAsFixed(2) + ' m';
    });
  });
  super.initState();
}
```

Şekil 5.3. Dart dili kullanılarak yazılmış bir ilk durum kodu örneği.

Flutter; Dart dili ile geliştirilmiş, çok platformlu, hızlı, hafif ve yüksek performanslı mobil uygulamalar geliştirmek için kullanılan açık kaynaklı bir kullanıcı arayüzü çerçevesidir. Bu çerçeve genişletilebilir bir durumdadır, böylece yeni özellikler ve işlevlerin eklenmesi mümkündür. Temel olarak bir Flutter uygulaması geliştirilirken her bir görsel bileşen için parçacıklar (ing: widget) oluşturulmaktadır. Bu parçacıklar bir araya getirilerek arayüz oluşturulmaktadır. Flutter'da, düğmeler veya metinler gibi Durum Bilgisi Olmayan (ing: stateless) parçacıklar ile metin kutuları, animasyon denetleyicileri gibi Durum Bilgisi Olan (ing: stateful) parçacıklar olmak üzere iki tür parçacık bulunmaktadır. Dart programlama dili kullanılarak kolayca hem Durum Bilgisi Olmayan hem de Durum Bilgisi Olan parçacıklar oluşturulabilmektedir. Bu sayede, uygun parçacıklara uygun işlevler tanımlaması yaparak bir mobil uygulamayı geliştirmek oldukça kolay hale gelebilmektedir (URL-21). Flutter çerçevesi ve Dart programlama dili kullanarak oluşturulmuş olan mobil cihaza ait konum bilgisinin ve detaylarının gösterildiği örnek bir liste parçacığı Şekil 5.4'te gösterilmektedir.

```
Widget _buildListItem() {
 return ListView.builder(
   padding: new EdgeInsets.all(1.0),
   itemCount: _latList.length,
   itemBuilder: (BuildContext context, int index) {
    return new ListTile(
      title: Text('Lat:' + _latList[index] + ' Lng:' + _lngList[index]),
      subtitle: Text('Alt:' +
          _altList[index] +
          ' Speed:' +
          _speedList[index] +
          '\nDate:' +
          _timeStampList[index].toString()),
      trailing: _iconList[index],
      enabled: _selectionList[index],
      isThreeLine: true,
      onTap: () => {},
    );
   },
 );
}
```

Şekil 5.4. Flutter çerçevesi kullanılarak yazılmış bir liste parçacığı örneği.

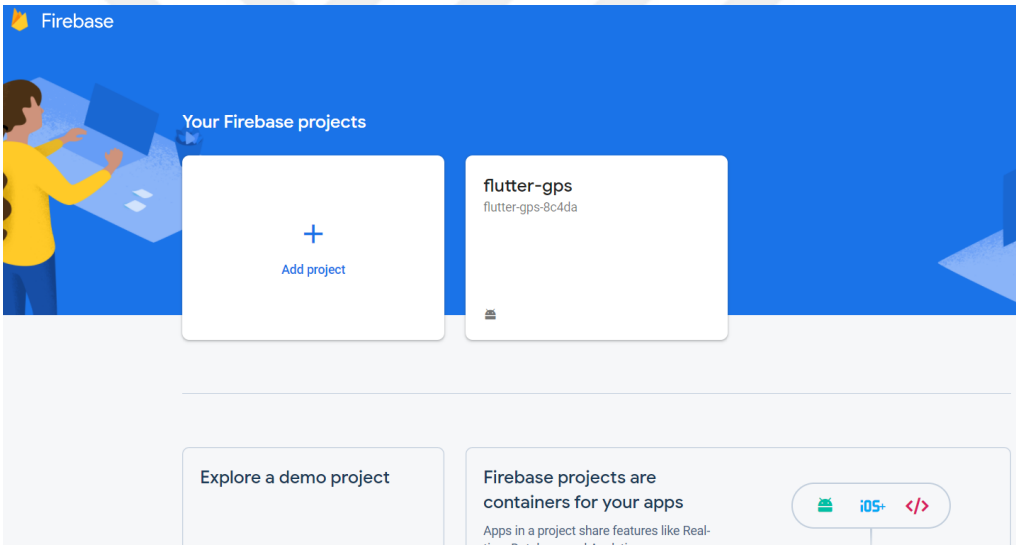
5.2.Mobil Uygulamanın İşlevleri

Tez çalışması kapsamında geliştirilmiş ve arayüz örnekleri Şekil 5.1'de gösterilen mobil uygulama bir giriş ekranı, menü ve ana ekrandan oluşmaktadır. Kullanıcı girişi yapılarak Firebase üzerinde sadece oturum açan kullanıcının verilerine ulaşması ve eklemesi sağlanmaktadır. Ana menüden "Start GPS Tracking" düğmesi tıklandığında GNSS konum verisi alma ve indirgeme işlemi başlatılmaktadır. Aynı şekilde, uygulama ilk çalıştırıldığı anda da GNSS konum verisi alma ve indirgeme işlemi otomatik olarak başlamaktadır. "Upload to Cloud" butonu ile indirgenen ancak internet bağlantısı olmadığı ya da zayıf olduğu ortamlarda bulut veri tabanına aktarılamayan verilerin aktarılması sağlanmaktadır. "Download from Cloud" butonu ile kullanıcı tarafından bulut veri tabanına kaydedilmiş olan verilerin mobil cihaz üzerine indirilmesi ve harita üzerinde gösterilmesi sağlanmaktadır. Alınan ve indirgenen GNSS konum verileri Şekil 5.1(a)'da gösterilen ekrandaki gibi Google Maps API'si (URL-22) kodlanarak görselleştirilmektedir. Ayrıca, yine ana ekran üzerinde mevcut konuma ait enlem, boylam, yükseklik ve hız bilgileri ile geçmiş konumlara ait enlem, boylam, yükseklik, hız ve zaman etiketi bilgileri listelenmektedir. Harita üzerinde; başlangıç noktası mavi, indirgenen noktalar kırmızı, seçilen noktalar ise yeşil işaretçiler ile gösterilmektedir. Aynı şekilde, geçmiş konumların olduğu listede; seçilen noktalar belirgin ve yanlarında yeşil im ile, seçilmeyen noktalar soluk ve gri çarpı ile gösterilmektedir.

5.3.Mobil Uygulamada Kullanılan Veri Modeli

Veri tabanı, verilerin saklanması ve gelecekte işlenmesi için kullanılmaktadır. Bulut veri tabanı kullanılmasının amacı ise mobil uyumluluğu yüksek, güvenilir, hızlı ve güvenli olmasıdır. Şekil 5.5'te konsol ekranı örneği gösterilen Firebase, Google tarafından sunulan bir bulut bilişim platformudur. Bu platform gerçek zamanlı veri senkronizasyonunu destekleyen, bulut tabanlı, istemci tarafından erişilebilir iki veri tabanı çözümü sunmaktadır. Platform üzerinde Realtime Database ve Cloud Firestore Database olmak üzere iki tür veri tabanı hizmeti bulunmaktadır (URL-23). Geliştirilen mobil uygulamada veri tabanı olarak Google Firebase Realtime Database kullanılmaktadır.

Realtime Database, Firebase'in orijinal veri tabanıdır. Gerçek zamanlı olarak istemciler arasında eşitlenmiş durumlar gerektiren mobil uygulamalar için verimli, düşük gecikmeli bir çözüm sunmaktadır. Realtime Database, bulut tabanlı uygulamaların hızlı bir şekilde geliştirilmesi için kullanılan bir *Servis Olarak Arka Uç* (ing. Backend as a Service - BaaS) hizmetidir. Bu hizmet, gerçek zamanlı veri senkronizasyonu sağlamak ve uygulama verilerini gerçek zamanlı olarak senkronize etmek için kullanılmaktadır. Realtime Database, belge tabanlı bir NoSQL veri tabanıdır. Saklanacak veriler JSON formatında depolanmakta ve uygulama kullanıcıları arasında veri senkronizasyonunu sağlamak için kullanılmaktadır. Bu nedenle, basit verilerin saklanması çok kolaydır. Varsayılan durumda 32 düzeye kadar verilerin iç içe yerleştirilmesine imkân vermektedir. Ayrıca, uygulama geliştiricilerinin daha hızlı ve daha kolay bir şekilde gerçek zamanlı uygulamalar oluşturmalarına olanak sağlamaktadır.



Şekil 5.5. Firebase konsolu ekranı

Bu tez çalışmasında kullanılan veri modeline ait örnek Şekil 5.6'da gösterilmiştir. Burada, her bir kullanıcının kendisine ait iki gezingeyi birbirinden ayırmak için ise gezinge kaydı her başlatıldığında veri tabanına eklenen ilk kaydın içerisinde "isStart" isminde true değeri saklayan bir alan eklenmektedir. Bu sayede, tek bir veri tabanı içerisinde birden fazla kullanıcıya ait birden fazla gezinge verisi saklanabilmekte ve işlenebilmektedir. Her bir kullanıcının her bir gezingesindeki seçilen her bir noktaya ait kaydedilen girdi için "altitude", "isSelected", "isStart", "latitude", "longitude", "speed", "timeStamp" alanları saklanmaktadır. "isSelected" alanı yöntemin testi sırasında sadece seçilen değil, elde

edilen tüm noktalar bulut veri tabanına aktarıldığı sırada yapılan seçim işleminin tutarlılığını kontrol etmek amacıyla oluşturulmuştur.


```
{
  "positions": {
    "-NTJJXc7UGr0RkPb_rBX": {
      "altitude": "410.79998779296875 m",
      "isSelected": "true",
      "isStart": "true",
      "latitude": "40.8212594° N",
      "longitude": "29.9291252° W",
      "speed": "0.000 m",
      "timeStamp": "2023-03-18 15:23:45.412155"
    },
    "-NTJJYbRjPp3gepHlxQ2": {
      "altitude": "410.79998779296875 m",
      "isSelected": "false",
      "isStart": "false",
      "latitude": "40.8212612° N",
      "longitude": "29.9291254° W",
      "speed": "1.117 m",
      "timeStamp": "2023-03-18 15:23:49.468266"
    },
    "-NTJJaXOsbP9aADo9VlV": {
      "altitude": "410.79998779296875 m",
      "isSelected": "true",
      "isStart": "false",
      "latitude": "40.8212437° N",
      "longitude": "29.929069° W",
      "speed": "2.508 m",
      "timeStamp": "2023-03-18 15:24:01.433152"
    }
  }
}
```


Şekil 5.6. Google Firebase Gerçek Zamanlı Bulut Veri Tabanına ait girdi örnekleri.

6. SONUÇLAR VE ÖNERİLER

Hareketliliğin artması ve teknolojinin gelişmesi sonucunda günlük yaşamda GNSS uydularından elde edilen konum verilerinin kullanımı da artmıştır. Biriken bu GNSS verilerinin indirgenmesi işlemi iletme, depolama ve işleme maliyetlerini azaltmaktadır. İndirgeme işlemi, orijinal gezinge içerisindeki noktalardan gezingenin hatlarını bozmayan noktaların atılarak önem derecesi yüksek, yeri doldurulamaz veya olmazsa olmaz olan noktaların seçilerek saklanması ile yapılmaktadır. Bu sayede, gereksiz noktalar eksiltilmekte iken, iletim ve depolama gibi maliyetleri azaltılmaktadır. Ayrıca, gezinge verileri içerisinde zaman etiketi barındırdığı için eksiltilen noktaların geri döndürülmesi istenildiğinde yüksek doğruluk oranı ile bu işlem gerçekleştirilebilmektedir.

Bu tez çalışmasında, GNSS'lerden en yaygın kullanılanı olan GPS verilerini indirgeyen bir matematiksel yöntem geliştirilmiştir. Ayrıca, geliştirilen bu yöntem kullanılarak yapay sinir ağı eğitimi için bir veri kümesi oluşturulmuştur. Oluşturulan bu veri kümesi kullanılarak yapay sinir ağı eğitimi gerçekleştirilmiştir. Eğitim sonucunda elde edilen model dışarıya aktarılmıştır. Matematiksel yöntemin ve yapay sinir ağı modelinin çevrim içi çalışmasının test edilmesi amacıyla mobil uygulama geliştirilmiştir. Dışa aktarılan yapay sinir ağı modeli bu mobil uygulama üzerinde çalıştırılmıştır. Mobil uygulama, verinin elde edildiği anda GNSS verisinin indirgenme durumuna karar vermektedir. Elde edilen nokta, saklanmasına karar verilen bir nokta ise mobil uygulama üzerinden cihazın internet bağlantısı kullanılarak gerçek zamanlı bulut veri tabanına aktararak saklamaktadır. Eğer elde edilen noktanın indirgenmesine karar verilmesi halinde ise, herhangi bir ek işlem uygulanmaksızın bu nokta atılmaktadır.

Çalışma kapsamında geliştirilen matematiksel indirgeme yöntemi; pencereleme boyutu 3 ve eşik değeri 5° seçilerek Microsoft GeoLife veri kümesindeki "20090522094234.plt" dosyasına uygulanmıştır. Bu dosyanın içerisinde 2560 adet koordinat bulunmaktadır. Bu girdiler kullanılarak yapılan deneme sonucunda 2390 adet nokta indirgenerek aynı dosya 170 nokta ile temsil edilmiştir. Bu işlem sonucunda %93,36 oranında indirgeme sağlanmıştır. Ayrıca RMSE değerinin 1,44114e-03, hesaplama süresinin ise 14,5 1ms olduğu gözlemlenmiştir.

Geliştirilen yöntemin başarımının anlaşılması bakımından bu değerlerin tek başına bir anlam ifade etmeyeceği düşünüldüğünden aynı veri kümesi üzerinde RDP algoritması eşik değeri 0.001 seçilerek koşulmuştur. Bu girdiler kullanılarak RDP algoritması ile yapılan deneme sonucunda ise 2525 adet nokta indirgenerek aynı dosya 55 nokta ile temsil edilmiştir. Bu işlem sonucunda %97,85 oranında indirgeme sağlanmıştır. Ancak RMSE değerinin 1,60097e+01, hesaplama süresinin ise 113,19 ms olduğu gözlemlenmiştir. Bu veriler ışığında, her ne kadar RDP algoritmasının indirgeme oranının yüksek olduğu görülse de RMSE değerinin bu denli yüksek olması özellikle hassasiyet gerektiren bölgelerin hiç dikkate alınmadığı ve gezingenin ciddi miktarda kabalaştırıldığı aşikardır. Burada beklenen indirgeme işlemi, sahip olunan gezingenin temel hatlarının gösterilmesi ise daha yüksek eşik değeri ve/veya pencere boyutu kullanılarak geliştirilen yöntem ile de elde edilebilir. RMSE değerine bakılarak hassasiyet gerektiren indirgeme işlemleri için geliştirilen yöntemin daha başarılı olduğu söylenebilir. Ayrıca, indirgeme işlemi RDP algoritmasında yalnızca çevrim dışı veri kümeleri üzerinde yapılabilirken, önerilen yöntem ile hem çevrim dışı hem de çevrim içi yapılabilmektedir. Bunların yanı sıra, önerilen yöntemin algoritma karmaşıklığı daha düşük olduğu için aynı veriyi aynı cihaz üzerinde çok daha hızlı bir şekilde işlemektedir.

Geliştirilen yöntemde pencere boyutu arttıkça indirgeme oranı da azalabilmektedir, ancak hata miktarı da azalmaktadır, yani hassasiyet artmaktadır. Yöntemin sonuçları gezingenin şekline bağlı olarak değişmektedir. Farklı veriler üzerinde yapılan denemelerde pencere boyutunun 3 olduğu ve 10° – 30° arasında ilk eşik değer kullanılan durumlarda istenilen indirgeme oranı, hata değeri ve hesaplama süresi ile uygun sonuçlar elde edildiği gözlemlenmiştir.

Çalışma kapsamında, eğitilen yapay sinir ağı modeli kullanılarak indirgeme işlemi; pencere boyutu 3 ve eşik değeri 10° seçilerek yine Microsoft GeoLife veri kümesindeki "20090522094234.plt" dosyasına uygulanmıştır. Bu girdiler kullanılarak yapılan deneme sonucunda veri kümesindeki 1066 adet nokta indirgenerek aynı gezinge 1494 nokta ile temsil edilmiştir. Bu işlem sonucunda %41,64 oranında indirgeme sağlanmıştır. Ayrıca RMSE değerinin 8,30117e-4, hesaplama süresinin ise 20,89 ms olduğu gözlemlenmiştir. Bu veriler ışığında, önerilen matematiksel indirgeme yönteminde olduğu gibi burada da her ne kadar RDP algoritmasının indirgeme oranının yüksek olduğu görülse de RMSE

değerinin bu denli yüksek olması özellikle hassasiyet gerektiren bölgelerin hiç dikkate alınmadığı ve gezingenin ciddi miktarda kabalaştırıldığı açıkça görülmektedir. Aynı durum önerilen matematiksel indirgeme yöntemi ile eğitilen modelin kıyaslanması durumunda da geçerlidir. Hassasiyet gerektiren indirgeme işlemleri için eğitilen modelin çok daha başarılı olduğu RMSE değeri kıyaslaması referans alınarak kolaylıkla söylenebilir. Ayrıca, indirgeme işlemi önerilen yapay sinir ağı modeli ile de hem çevrim dışı hem de çevrim içi yapılabilmektedir. Bunların yanı sıra, önerilen yapay sinir ağı modeli aynı veriyi aynı cihaz üzerinde çok hızlı bir biçimde gerçekleştirmektedir. Hem önerilen matematiksel indirgeme yönteminde hem de eğitilen YSA modelinde indirgeme işlemi daha hızlı bir biçimde yapılabilmektedir.

Çalışmanın önemli noktalarından biri, önerilen matematiksel indirgeme yöntemi ile önem derecesi yüksek noktaların elde edildiği anda seçilmesine karar verilerek ek bir maliyet gerektirmeden indirgemenin yapılabilmesidir. Diğer bir husus ise, GNSS verisi indirgemesi için yapay sinir ağı kullanılmasıdır. GNSS verisi indirgemesi amacıyla yapay zekanın kullanılması ile literatürdeki önemli eksikliklerden birisinin giderildiği değerlendirilmektedir. Bu çalışma ile, GNSS verisi indirgemesi alanında eksik olduğu görülen hızlı, hassas ve hem çevrim içi hem de çevrim dışı indirgeme yapabilen yeni iki yöntem önerilmiştir.

İndirgeme sonrasında elde edilen gezingenin geri döndürülmesi bu tezin konusu olmadığından orijinal gezingenin elde edilmesi ile ilgili herhangi bir çalışma yapılmamıştır. Fakat, literatürde var olan spline, akima, makima vb. eğri uydurma yöntemleri kullanılarak orijinal gezingenin elde edilmesinin sağlanabileceği düşünülmektedir.

KAYNAKLAR

Abam, M. A., Daneshpajouh, S., Deleuran, L., Ehsani, S., & Ghodsi, M. (2014). Computing homotopic line simplification. *Computational Geometry: Theory and Applications*, *47*(7), 728-739. https://doi.org/10.1016/j.comgeo.2014.02.002

Bao, L., & Du, M. (2018). A distance-based trajectory outlier detection method on maritime traffic data. *Proceedings - 2018 4th International Conference on Control, Automation and Robotics, ICCAR 2018*, 340-343. https://doi.org/10.1109/ICCAR.2018.8384697

Bekhet, S., & Ahmed, A. (2020). Evaluation of similarity measures for video retrieval. *Multimedia Tools and Applications*, *79*(9-10), 6265-6278. https://doi.org/10.1007/s11042-019-08539-4

Bian, J., Tian, D., Tang, Y., & Tao, D. (2019). Trajectory data classification: a review. *ACM Transactions on Intelligent Systems and Technology*, *10*(4). https://doi.org/10.1145/3330138

Byon, S., Kwon, E., Jung, E. S., & Lee, Y. T. (2018). An Implementation of Location Trajectory Data Reduction. *9th International Conference on Information and Communication Technology Convergence: ICT Convergence Powered by Smart Intelligence, ICTC 2018*, 1270-1272. https://doi.org/10.1109/ICTC.2018.8539709

Cantemir, Z. (2019). *Derin Öğrenme Yöntemleriyle Medikal Görüntü Sınıflandırma*. Gazi Üniversitesi.

Chen, C., Ding, Y., Xie, X., Zhang, S., Wang, Z., & Feng, L. (2019). TrajCompressor: An Online Map-matching-based Trajectory Compression Framework Leveraging Vehicle Heading Direction and Change. *IEEE Transactions on Intelligent Transportation Systems*, 1-17. https://doi.org/10.1109/tits.2019.2910591

Chen, J., Wang, D., Xiao, Z., & Havyarimanay, V. (2018). *OCT: A Novel Opportunistic Compression and Transmission Approach for Private Car Trajectory Data*. 401-401. https://doi.org/10.1109/dcc.2018.00054

Chen, L., Özsu, M. T., & Oria, V. (2005). Robust and fast similarity search for moving object trajectories. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 491-502. https://doi.org/10.1145/1066157.1066213

Chen, M., Xu, M., & Franti, P. (2012). Compression of GPS trajectories. *Data Compression Conference Proceedings*, 62-71. https://doi.org/10.1109/DCC.2012.14

Cojocaru, S., Birsan, E., Batrinca, G., & Arsenie, P. (2009). GPS-GLONASS-GALILEO: A dynamical comparison. *Journal of Navigation*, *62*(1), 135-150. https://doi.org/10.1017/S0373463308004980

Deng, Z., Han, W., Wang, L., Ranjan, R., Zomaya, A. Y., & Jie, W. (2017). An efficient online direction-preserving compression approach for trajectory streaming data. *Future Generation Computer Systems*. https://doi.org/10.1016/j.future.2016.09.019

Douglas, D. H., & Peucker, T. K. (1973). Algorithms for The Reduction of The Number of Points Required to Represent A Digitized Line or Its Caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, *10*(2), 112-122. https://doi.org/10.3138/fm57-6770-u75u-7727

Fernández-García, N. L., Del-Moral Martínez, L., Carmona-Poyato, A., Madrid-Cuevas, F. J., & Medina-Carnicer, R. (2016). A new thresholding approach for automatic generation of polygonal approximations. *Journal of Visual Communication and Image Representation*, *35*, 155-168. https://doi.org/10.1016/j.jvcir.2015.12.013

Fitri, N. A., Emba, R. Z., Mufid, M. R., Fiyanto, A., Wajib, W., & Shofyan, A. (2022). Kediri City Tourism Object Application Using Firebase Realtime Database Technology. *Proceedings of the International Conference on Applied Science and Technology on Social Science 2021 (iCAST-SS 2021)*, *647*, 892-897. https://doi.org/10.2991/ASSEHR.K.220301.147

Gasparetta, A., Boscario, P., Lanzutti, A., & Vidoni, R. (2015). *Motion and Operation Planning of Robot Systems, Chapter 1 - Path Planning and Trajectory Planning Algorithms: A General Overview* (C. 29, Sayı September 2017). https://doi.org/10.1007/978-3-319-14705-5

Gültekin, A., Diri, S., & Becerikli, Y. (2023). Simplified and Smoothed Rapidly-Exploring Random Tree Algorithm for Robot Path Planning. *Tehnički vjesnik*, *30*(3), 891-898. https://doi.org/10.17559/TV-20221015080721

Hao, P., Yao, C., Meng, Q., Yu, X., & Li, X. (2019). A heading maintaining oriented compression algorithm for GPS trajectory data. *Informatica (Netherlands)*, *30*(1), 33-52. https://doi.org/10.15388/Informatica.2019.196

Hershberger, J., Ave, L., & Snoeyink, J. (1992). *Speeding Up the Douglas-Peucker Line-Simplication Algorithm*. 1-16. https://doi.org/10.5555/902273

Hyeon, J., Jang, B., Choi, H., Kim, J., Kim, D., & Doh, N. (2023). Photo-realistic 3D model based accurate visual positioning system for large-scale indoor spaces. *Engineering Applications of Artificial Intelligence*, *123*, 106256. https://doi.org/10.1016/J.ENGAPPAI.2023.106256

Katsikouli, P., Sarkar, R., & Gao, J. (2014). Persistence based online signal and trajectory simplification for mobile devices. *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*, *04-07-Nove*, 371-380. https://doi.org/10.1145/2666310.2666388

Khawas, C., & Shah, P. (2018). Application of Firebase in Android App Development-A Study. *Article in International Journal of Computer Applications*, *179*(46), 975-8887. https://doi.org/10.5120/ijca2018917200

Kingma, D. P., & Ba, J. L. (2014). Adam: A Method for Stochastic Optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. https://arxiv.org/abs/1412.6980v9

Koşan, M. A., Coşkun, A., & Karacan, H. (2019). Yapay Zekâ Yöntemlerinde Entropi. Içinde *Journal of Information Systems and Management Research* (C. 1, Sayı 1, ss. 15-22). M. Hanefi CALP.

Kwon, N., Kim, J., Lim, S., Chae, J., & Park, H. (2018). T-reduce: Route-aware mobile trajectory data reduction in transportation management systems. *IEEE Access*, *6*, 38934-38946. https://doi.org/10.1109/ACCESS.2018.2852787

Lee, J.-G., Han, J., & Whang, K.-Y. (2007). *Trajectory clustering: a partition-and-group framework*. 593. https://doi.org/10.1145/1247480.1247546

Lee, M. J. L., Ho, H. Y., Hsu, L. T., & Au, S. L. M. (2021). BIPS: Building Information Positioning System. *2021 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2021*. https://doi.org/10.1109/IPIN51156.2021.9662575

Leondes, C. T., & Yonezawa, K. (1978). Evaluation of Geometric Performance of Global Positioning System. *IEEE Transactions on Aerospace and Electronic Systems*, *AES-14*(3), 533-539. https://doi.org/10.1109/TAES.1978.308616

Li, H., Kulik, L., & Ramamohanarao, K. (2014). Spatio-temporal trajectory simplification for inferring travel paths. *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*, *04-07-Nove*, 63-72. https://doi.org/10.1145/2666310.2666409

Lin, C. Y., Hung, C. C., & Lei, P. R. (2017). A velocity-preserving trajectory simplification approach. *TAAI 2016 - 2016 Conference on Technologies and Applications of Artificial Intelligence, Proceedings*, 58-65. https://doi.org/10.1109/TAAI.2016.7880172

Liu, B., Liu, X., Li, D., Shi, Y., Fernandez, G., & Wang, Y. (2020). A vector line simplification algorithm based on the douglas-peucker algorithm, monotonic chains and dichotomy. *ISPRS International Journal of Geo-Information*, *9*(4). https://doi.org/10.3390/ijgi9040251

Liu, J., Zhao, K., Sommer, P., Shang, S., Kusy, B., & Jurdak, R. (2015). Bounded Quadrant System: Error-bounded trajectory compression on the go. *Proceedings - International Conference on Data Engineering*, *2015-May*, 987-998. https://doi.org/10.1109/ICDE.2015.7113350

Liu, J., Zhao, K., Sommer, P., Shang, S., Kusy, B., Lee, J.-G., & Jurdak, R. (2016). A Novel Framework for Online Amnesic Trajectory Compression in Resource-constrained Environments. *IEEE Transactions on Knowledge and Data Engineering*, *28*(11), 2827-2841.

Long, C., Wong, R. C. W., & Jagadish, H. V. (2013). Direction-preserving trajectory simplification. *Proceedings of the VLDB Endowment*, *6*(10), 949-960. https://doi.org/10.14778/2536206.2536221

Mazimpaka, J. D., & Timpf, S. (2016). Trajectory data mining: A review of methods and applications. *Journal of Spatial Information Science*, *13*(2016), 61-99. https://doi.org/10.5311/josis.2016.13.263

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115-133. https://doi.org/10.1007/BF02478259/METRICS

McMaster, R. B. (1987). Automated Line Generalization. *Cartographica: The International Journal for Geographic Information and Geovisualization*, *24*(2), 74-111. https://doi.org/10.3138/3535-7609-781G-4L20

McNally, J. L., & Breithaupt, R. W. (1987). Mobile satellite systems. A review. *Acta Astronautica*, *16*(C), 279-286. https://doi.org/10.1016/0094-5765(87)90115-9

Meng, Q., Yu, X., Yao, C., Li, X., Li, P., & Zhao, X. (2017). Improvement of OPW-TR algorithm for compressing GPS trajectory data. Içinde *Journal of Information Processing Systems* (C. 13, Sayı 3, ss. 533-545). https://doi.org/10.3745/JIPS.03.0073

Muckell, J., Hwang, J.-H., Lawson, C. T., & Ravi, S. S. (2010). Algorithms for compressing GPS trajectory data. *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, *6*(1), 402-405. https://doi.org/10.1145/1869790.1869847

Nibali, A., & He, Z. (2015). Trajic: An Effective Compression System for Trajectory Data. *IEEE Transactions on Knowledge and Data Engineering*, *27*(11), 3138-3151. https://doi.org/10.1109/TKDE.2015.2436932

Noe, P. S., & Myers, K. A. (1976). A Position Fixing Algorithm for the Low-Cost GPS Receiver. *IEEE Transactions on Aerospace and Electronic Systems*, *AES-12*(2), 295-297. https://doi.org/10.1109/TAES.1976.308310

Pelekis, N., & Theodoridis, Y. (2014). Mobility data management and exploration. Içinde *Mobility Data Management and Exploration*. Springer New York. https://doi.org/10.1007/978-1-4939-0392-4

Potamias, M., Patroumpas, K., & Sellis, T. (2006). Sampling trajectory streams with spatiotemporal criteria. *Proceedings of the International Conference on Scientific*

and Statistical Database Management, SSDBM, 275-284. https://doi.org/10.1109/SSDBM.2006.45

Reis, L. A., Pereira, S. L., Dias, E. M., & Scoton, M. L. R. P. D. (2022). *A Smartphone-Based Solution to Manage Hazardous Materials Transportation: A Review*. 23-33. https://doi.org/10.1007/978-3-031-08545-1_2

Samsonov, T. E., & Yakimova, O. P. (2017). Shape-adaptive geometric simplification of heterogeneous line datasets. *International Journal of Geographical Information Science*, *31*(8), 1485-1520. https://doi.org/10.1080/13658816.2017.1306864

Sandu Popa, I., Zeitouni, K., Oria, V., & Kharrat, A. (2014). Spatio-temporal compression of trajectories in road networks. *GeoInformatica*, *19*(1), 117-145. https://doi.org/10.1007/s10707-014-0208-4

Shan-shi, Z. H. O. U., Xiao-gong, H. U., Li, L. I. U., Feng, H. E., Cheng-pan, T. A. N. G., & Jun-yang, P. A. N. G. (2019). Status of Satellite Orbit Determination and Time Synchronization Technology for Global Navigation Satellite Systems. *Chinese Astronomy and Astrophysics*, *43*(4), 479-492. https://doi.org/10.1016/j.chinastron.2019.11.003

Sharma, S., Sharma, S., & Athaiya, A. (2020). ACTIVATION FUNCTIONS IN NEURAL NETWORKS. *International Journal of Engineering Applied Sciences and Technology*, *04*(12), 310-316. https://doi.org/10.33564/IJEAST.2020.V04I12.054

Spinsanti, L., Berlingerio, M., & Pappalardo, L. (2013). Mobility and Geo-Social Networks. Içinde C. Renso, S. Spaccapietra, & E. Zimanyi (Ed.), *Mobility Data* (ss. 315-333). Cambridge University Press. https://doi.org/10.1017/CBO9781139128926.017

United Nations Office for Outer Space Affairs. (2010). *Current and Planned Global and Regional Navigation Satellite Systems and Satellite-based Augmentations Systems*. *ST/SPACE/50*, 70. http://www.unoosa.org/pdf/publications/icg_ebook.pdf

URL-1. http://www.gps.gov/ (Ziyaret tarihi: 6 Temmuz 2020)

URL-2. http://www.unoosa.org/pdf/publications/icg_ebook.pdf (Ziyaret tarihi: 6 Temmuz 2020)

URL-3. https://github.com/barbeau/gpstest/blob/master/FAQ.md (Ziyaret tarihi: 6 Temmuz 2020)

URL-4. http://en.beidou.gov.cn/SYSTEMS/System/ (Ziyaret tarihi: 6 Temmuz 2020)

URL-5. https://www.gsc-europa.eu/ (Ziyaret tarihi: 6 Temmuz 2020)

URL-6. https://www.glonass-iac.ru/en/ (Ziyaret tarihi: 6 Temmuz 2020)

URL-7. https://www.isro.gov.in/ (Ziyaret tarihi: 6 Temmuz 2020)

URL-8. https://qzss.go.jp/en/ (Ziyaret tarihi: 6 Temmuz 2020)

URL-9. https://ti.arc.nasa.gov/m/profile/adegani/Crash of Korean Air Lines Flight 007.pdf (Ziyaret tarihi: 6 Temmuz 2020)

URL-10. https://www.microsoft.com/en-us/research/publication/geolife-gps-trajectory-dataset-user-guide/ (Ziyaret tarihi: 6 Temmuz 2020)

URL-11. https://www.nhc.noaa.gov/data/ (Ziyaret tarihi: 3 Kasım 2020)

URL-12. https://www.movebank.org/ (Ziyaret tarihi: 3 Kasım 2020)

URL-13. https://www.crcv.ucf.edu/ (Ziyaret tarihi: 3 Kasım 2020)

URL-14. https://www.faa.gov/ (Ziyaret tarihi: 16 Haziran 2021)

URL-15. https://www.flightradar24.com/ (Ziyaret tarihi: 16 Haziran 2021)

URL-16. https://gpsbabel.com (Ziyaret tarihi: 6 Temmuz 2020)

URL-17. https://gpsvisualizer.com (Ziyaret tarihi: 6 Temmuz 2020)

URL-18. https://www.tensorflow.org/ (Ziyaret tarihi: 22 Aralık 2022)

URL-19. https://developer.android.com/docs (Ziyaret tarihi: 23 Aralık 2021)

URL-20. https://dart.dev/guides (Ziyaret tarihi: 23 Mart 2023)

URL-21. https://docs.flutter.dev/ (Ziyaret tarihi: 23Mart 2023)

URL-22. https://developers.google.com/maps/ (Ziyaret tarihi: 16 Haziran 2021)

URL-23. https://firebase.google.com/docs (Ziyaret tarihi: 23 Mart 2023)

Vlachos, M., Kollios, G., & Gunopulos, D. (2002). Discovering similar multidimensional trajectories. *Proceedings - International Conference on Data Engineering*, 673-684. https://doi.org/10.1109/ICDE.2002.994784

Wu, S. T., & Márquez, M. R. G. (2003). A non-self-intersection Douglas-Peucker algorithm. *Brazilian Symposium of Computer Graphic and Image Processing*, *2003-January*, 60-66. https://doi.org/10.1109/SIBGRA.2003.1240992

Xu, M., Wu, J., Liu, M., Xiao, Y., Wang, H., & Hu, D. (2019). Discovery of Critical Nodes in Road Networks Through Mining from Vehicle Trajectories. *IEEE*

Transactions on Intelligent Transportation Systems, *20*(2), 583-593. https://doi.org/10.1109/TITS.2018.2817282

Xuan, Y., & Xu, J. (2011). GPS location data reducing based on turning point judement method. *2011 2nd International Conference on Mechanic Automation and Control Engineering, MACE 2011 - Proceedings*, 7395-7398. https://doi.org/10.1109/MACE.2011.5988759

Zhang, H., Hu, X., Bai, W., Zhang, H., Zhou, Y., & Lin, Y. (2019). An Efficient IoV Trajectory Compression Method in Vehicle Terminals Using Width-Direction-Angle. *IEEE Access*, *7*, 71447-71458. https://doi.org/10.1109/ACCESS.2019.2919800

Zhang, S. K., Liu, Z. J., Cai, Y., Wu, Z. L., & Shi, G. Y. (2016). AIS Trajectories Simplification and Threshold Determination. *Journal of Navigation*, *69*(4), 729-744. https://doi.org/10.1017/S0373463315000831

Zhang, Z., & Sabuncu, M. R. (2018). Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels. *Advances in Neural Information Processing Systems*, *2018-December*, 8778-8788. https://arxiv.org/abs/1805.07836v4

Zhao, H., Chen, W., Zhou, S., Zheng, F., & Liu, Y.-H. (2023). Localization and Motion Planning of Industrial Tractor–Trailers Vehicles. *IEEE Transactions on Control Systems Technology*, 1-9. https://doi.org/10.1109/TCST.2023.3275497

Zhao, L., & Shi, G. (2018). A method for simplifying ship trajectory based on improved Douglas–Peucker algorithm. *Ocean Engineering*, *166*, 37-46. https://doi.org/10.1016/j.oceaneng.2018.08.005

Zheng, Y. (2015). Trajectory data mining: An overview. *ACM Transactions on Intelligent Systems and Technology*, *6*(3), 1-41. https://doi.org/10.1145/2743025

Zheng, Y., Li, Q., Chen, Y., Xie, X., & Ma, W. Y. (2008). Understanding mobility based on GPS data. *UbiComp 2008 - Proceedings of the 10th International Conference on Ubiquitous Computing*, 312-321. https://doi.org/10.1145/1409635.1409677

Zheng, Y., Xie, X., & Ma, W. (2010). GeoLife: A Collaborative Social Networking Service among User, Location and Trajectory. *IEEE Data Eng. Bulletin*, *33*(2), 32-40. https://doi.org/10.1.1.165.4216

Zheng, Y., Zhang, L., Xie, X., & Ma, W. Y. (2009). Mining interesting locations and travel sequences from GPS trajectories. *WWW'09 - Proceedings of the 18th International World Wide Web Conference*, 791-800. https://doi.org/10.1145/1526709.1526816

KİŞİSEL YAYIN VE ESERLER

Akcakaya, A., **Diri, S.**, Şahin, S., Uygulama ve Araştırma Merkezi, B., Üniversitesi, K., Yazar, S., & Fakültesi Bilgisayar Mühendisliği, M. (2023). Kocaeli Üniversitesi Öğrenci Bilgi Sistemi Mobil Uygulamasının Kullanılabilirliği. *Yalvaç Akademi Dergisi, 8*(1), 65-74. https://doi.org/10.5712/YALVAC.1257634

Akcakaya, A., **Diri, S.**, Şahin, S., & Yıkılmazçınar, R. S. (2022). Panic Button Mobile Application Usability Study. *International Journal of Engineering and Innovative Research, 4*(2), 104-113. https://doi.org/10.47933/IJEIR.1085846

Altıntaş, L., Alimoğlu, M. K., Alvur, T. M., Yıldız, G., & **Diri, S.** (2012). Ulusal Çekirdek Eğitim Programı Nın Tıp Eğitimi Programına Entegrasyonunda Yazılım Destekli Uygulama Örneği Kocaeli Üniversitesi Tıp Fakültesi Deneyimi. *Tıp Eğitimi Dünyası*, 34, 6-12.

Altıntaş, L., Alimoğlu, M. K., Alvur, T. M., Yıldız, G., & **Diri, S.** (2012). Ulusal Çekirdek Eğitim Programının Tıp Eğitimi Programına Entegrasyonunda Yazılım Destekli Uygulama Örneği Kocaeli Tıp Fakültesi Deneyimi. VIII. *Ulusal Tıp Eğitimi Kongresi*.

Denizhan, S., **Diri, S.**, Öğütcü, S., & Alkan, A. (2019). Product Warehouse Ramp Assignment Using Artificial Bee Colony. *10th International Symposium on Intelligent Manufacturing and Service Systems*, 838-845.

Diri, S., Gültekin, A., Şahin, S. A., & Kayapınar, Ö. B. (2022). Classification of Server Network Traffic for DDoS Attacks Using MapReduce. *7th International Scientific Conference "Telecommunications, Informatics, Energy and Management"*, 139-142.

Diri, S., Öğütcü, S., & Yıldırım, M. (2018). Genetic Algorithm and Fuzzy Logic Based Auto-Test System for Distance Education. *4th International Conference on Engineering and Natural Sciences (ICENS 2018)*, 734-738.

Diri, S., Yıldırım, M. (2023). Hareketli Nesnelerin Konum Verilerinde İndirgeme Yöntemleri Üzerine Bir İnceleme. *International Congress of New Horizons in Sciences (INCOHIS 2023 Spring)*, 211-215.

Gültekin, A., **Diri, S.**, Altuntaş, F., & Yerlikaya, Z. (2018). Rol Tabanlı Erişim Kontrolü Kullanılarak Öğrenci Bilgi Sistemi Modellenmesi. *Uluslararası Marmara Fen ve Sosyal Bilimler Kongresi*.

Gültekin, A., **Diri, S.**, & Becerikli, Y. (2022). Çoklu İHA'nın Haberleşmesi ile Düşman Ortamının Matematiksel Modellemesi. *3rd International Conference on Applied Engineering and Natural Sciences ICAENS 2022*, 877-882.

Gültekin, A., **Diri, S.** & Becerikli, Y. (2023). Simplified and Smoothed Rapidly-Exploring Random Tree Algorithm for Robot Path Planning. *Tehnički vjesnik, 30*(3), 891-898. https://doi.org/10.17559/TV-20221015080721.

Maden, E., Yıldırım, M., & **Diri, S.** (2020). Kocaeli Üniversitesi Mühendislik Fakültesi Mezunlarının Akademik Başarılarının Veri Madenciliği Yöntemleri Kullanılarak İncelenmesi. Içinde H. Kahveci (Ed.), *2. Uluslararası Akademik Araştırmalar Kongresi* (ss. 132-132).

Öğütcü, S., **Diri, S.**, & Altıntaş, L. (2018). Kocaeli Üniversitesi Sınav Süreç Yönetimi Çalışması. 20. *Akademik Bilişim 2018 Konferansı*, 71-76.

Öğütcü, S., **Diri, S.**, & Yıldırım, M. (2018). Text Preprocessing for an Experiment Code on the Internet Assisted Laboratory System. *4th International Conference on Engineering and Natural Sciences (ICENS 2018)*, 727-733.

ÖZGEÇMİŞ

İlk ve orta öğrenimini Bursa'da tamamladı. 2011 yılında Kocaeli Üniversitesi Teknik Eğitim Fakültesi Bilgisayar Öğretmenliği Bölümünden mezun oldu. 2018 yılında Kocaeli Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümünde lisans tamamlama yaparak mühendis unvanı aldı. Yine aynı yıl Kocaeli Üniversitesi Bilgisayar Mühendisliği ana bilim dalı yüksek lisans programından "Sezgisel Yöntemler Kullanarak Akıllı Soru Bankası Sistemi Tasarımı" konulu tezini başarıyla savunarak mezun oldu. Ardından Kocaeli Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı'nda doktora eğitimine başladı. Doktora eğitimi sırasında başta konum verisi indirgeme konusu olmak üzere robot yol planlama, veri madenciliği, yapay zekâ ve insan-bilgisayar etkileşimi konularında çeşitli çalışmaları bulunmaktadır.

2011-2017 yılları arasında Kocaeli Üniversitesi Tıp Fakültesi ve Öğrenci İşleri Daire Başkanlığı'nda yazılım geliştirici olarak 2017-2018 yılları arasında Kocaeli Üniversitesi Öğrenci İşleri Daire Başkanlığında Uzman olarak 2018 yılından beri Kocaeli Üniversitesi Bilgisayar Uygulama ve Araştırma Merkezinde Öğretim Görevlisi olarak çalışmaktadır. Evli ve bir kız çocuğu babasıdır.