T.C.
İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

AKILLI ŞEHİRLER İÇİN LORAWAN AĞLARINDA MAKİNE ÖĞRENİMİ TABANLI DIŞ MEKAN KONUMLANDIRMA TEKNİKLERİNİN PERFORMANS ANALİZİ

Gufran BAYRAKTAR

DANIŞMAN
Dr. Öğr. Üyesi Gülsüm Zeynep GÜRKAŞ AYDIN

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Temmuz, 2022

TEZ KABUL VE ONAYI

Gufran BAYRAKTAR tarafından, **Dr. Öğr. Üyesi Gülsüm Zeynep GÜRKAŞ AYDIN** danışmanlığında hazırlanan **"Akıllı Şehirler için LoRaWAN Ağlarında Makine Öğrenimi Tabanlı Dış Mekan Konumlandırma Tekniklerinin Performans Analizi "** başlıklı bu çalışma, jürimiz tarafından **01/07/2022** tarihinde yapılan sınav sonucunda **oy birliği** ile **BAŞARILI** bulunarak **Yüksek Lisans Tezi** olarak kabul edilmiştir.

Tez Jürisi

		İmza	**Sonuç**
DANIŞMAN	Dr. Öğr. Üyesi Gülsüm Zeynep GÜRKAŞ AYDIN İstanbul Üniversitesi- Cerrahpaşa Bilgisayar Mühendisliği Anabilim Dalı		☒ Kabul ☐ Ret
ÜYE	Doç. Dr. Derya YILTAŞ KAPLAN İstanbul Üniversitesi- Cerrahpaşa Bilgisayar Mühendisliği Anabilim Dalı		☒ Kabul ☐ Ret
ÜYE	Dr. Öğr. Üyesi Ali BOYACI İstanbul Ticaret Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı		☒ Kabul ☐ Ret

BEYAN

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve bilimsel etik kuralları içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını ve her türlü hukuki sorumluluğu aldığımı kabul ederim.

Gufran BAYRAKTAR

(İmza)

Annem/Babam/Kardeşlerim/Hocam/Bana destek veren herkese ithaf ediyorum…

BÜTÇE DESTEKLERİ

AKILLI ŞEHİRLER İÇİN LORAWAN AĞLARINDA MAKİNE ÖĞRENİMİ TABANLI DIŞ MEKAN KONUMLANDIRMA TEKNİKLERİNİN PERFORMANS ANALİZİ

Bu tez çalışması için herhangi bir kurumdan bütçe desteği alınmamıştır.

TEŞEKKÜR

Tez çalışmamın her aşamasında bana yardımcı olan, desteklerini esirgemeyen ve beni yüreklendiren değerli danışman hocam Sayın Dr.Öğr.Üyesi Gülsüm Zeynep GÜRKAŞ AYDIN'a sonsuz teşekkürlerimi sunarım.

Tüm hayatım boyunca benim yanımda olan, aldığım karaları her zaman destekleyen, sabır ve anlayışlarıyla çalışmalarımdaki başarıma büyük katkısı olan, her zaman ümit veren annem Najah BAYRAKTAR'a, babam Ahmed BAYRAKTAR'a, ağabeylerim Yusuf'a, Muhammed'e ve ablam Alanur'a sonsuz şükranlarımı sunar ve teşekkür ederim.

Çalışmalarım sırasında sabır gösteren ve değerli desteklerini esirgemeyen müdürüm Şerif BADAWY Bey'e, ümit veren ve destek olan sevgili dostum Dania ANCERENİ'ye, çalışmam sırasında küçük veya büyük yardımını esirgemeyen herkese çok teşekkür ederim.

Temmuz 2022 **Gufran BAYRAKTAR**

İÇİNDEKİLER

Sayfa No

1. GİRİŞ 1
2. KAVRAMSAL ÇERÇEVE 3
2.1. KONUMLANDIRMA 3
2.1.1. Geleneksel Konumlandırma Teknikleri 3
2.2. LORAWAN 6
2.2.1. LoRaWAN Mimarisi 7
2.2.2. LoRaWAN ile İlgili Çalışmalar 9
2.3. LORAWAN İLE KONUMLANDIRMA 10
2.3.1. LoRaWAN TDOA Tabanlı Konumlandırma 10
2.3.2. LoRaWAN RSSI Tabanlı Konumlandırma 10
2.3.3. LoRaWAN Konumlandırma ile İlgili Çalışmalar 11
3. YÖNTEM 13
3.1. KULLANILAN TOPLULUK ÖĞRENME ALGORİTMALARI (ENSEMBLE LEARNING ALGORITHMS) 13
3.1.1. Torbalama ya da Önyükleme Toplaması, Rastgele Orman (Bagging ya da Bootstrap Aggregation, Random Forest) 13
3.1.2. Artırma (Boosting), AdaBoost ve Gradyan Artırma 15
3.2. KULLANILAN REGRESYON, SINIFLANDIRMA VE KÜMELEME ALGORİTMALARI 18
3.2.1. K- En Yakın Komşu (K-Nearest Neighbors – kNN) Algoritması 18
3.2.2. K-Ortalamalar Kümeleme (K- Means Clustering) Algoritması 19
3.3. KULLANILAN VERİ SETİ VE GELİŞTİRME ORTAMI 21
3.3.1. Kullanılan Veri Seti 21
3.3.2. Veri Setinin Ön İşlemesi 21
3.3.3. Geliştirme Ortamı 22
3.4. ÖNERİLEN METODOLOJİ 23
3.5. PERFORMANS METRİKLERİ 24

4. BULGULAR 26
4.1. KULLANILAN ALGORİTMALAR İLE ELDEN EDİLEN SONUÇLAR 26
4.1.1. kNN Algoritması ile Elde Edilen Sonuçlar 26
4.1.2. Ağırlıklı KNN Algoritması ile Elde Edilen Sonuçlar 28
4.1.3. Rastgele Orman Algoritması ile Elde Edilen Sonuçlar 30
4.1.4. Gradyan Artırma Algoritması ile Elde Edilen Sonuçlar 33
4.1.5. Adaptive Boost Algoritması ile Elde Edilen Sonuçlar 35
.4.1.6 Tüm Algoritmalar Arasındaki Karşılaştırma ile Elde Edilen Sonuçlar 38
4.2. HİBRİT YÖNTEMLER İLE ELDE EDİLEN SONUÇLAR 38
4.3. ÖNERİLEN METODOLOJİ İLE ELDE EDİLEN SONUÇLAR 39
4.3.1. AdaBoost Kullanarak Önerilen Metodoloji ile Elde Edilen Sonuçlar 39
4.3.2. Rastgele Orman Kullanarak Önerilen Yöntem ile Elde Edilen Sonuçlar 40
4.3.3. Hibrit Yöntem Kullanılaarak Önerilen Metodoloji ile Elde Edilen Sonuçlar 41
5. TARTIŞMA 44
6. SONUÇ VE ÖNERİLER 46
KAYNAKLAR 47
EKLER 52

ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1. Üçgenleme Yöntemi....4
Şekil 2.2. Trilaterasyon Yöntemi....5
Şekil 2.3. Multilaterasyon Yöntemi....5
Şekil 2.4. LoRaWAN mimarisi. 7
Şekil 2.5. LoRaWAN Teknolojisinin Katmanlı Mimarisi....8
Şekil 3.1. Torbalama....14
Şekil 3.2. Rastgele Orman 15
Şekil 3.3. Karar Kütükleri Yapısı 16
Şekil 3.4. Dirsek Yöntemi 20
Şekil 3.5. Veri setinin veri noktalarının mekansal dağılımı 21
Şekil 3.6. Önerilen Metodolojinin Akış Şeması....23
Şekil 3.7. K- Ortalama Kümeleme Algoritmasının sonucunda oluşan kümeler 24
Şekil 4.1. Ortalama hata ile k değerleri arasındaki ilişki 28
Şekil 4.2. Ortalama hata ile k değerleri arasındaki ilişki....30
Şekil 4.3. Ortalama hata ile ormandaki ağaç sayısı arasındaki ilişki 32
Şekil 4.4. Medyan hata ile ormandaki ağaç sayısı arasındaki ilişki 32
Şekil 4.5. Öğrenme oranı 0.1 iken, Ortalama hata ile artırma aşamalarının sayısı arasındaki ilişki 34
Şekil 4.6. Öğrenme oranı 0.1 iken, Medyan hata ile artırma aşamalarının sayısı arasındaki ilişki 35
Şekil 4.7. Öğrenme oranı 1.0 iken, ortalama hata ve medyan hata ile karar ağacının maksimum derinliği arasındaki ilişki 37
Şekil 4.8. Öğrenme oranı 0.1 iken, ortalama hata ve medyan hata ile karar ağacının maksimum derinliği arasındaki ilişki 37

Şekil 4.9. Üç yaklaşımın ortalama hatası ile küme sayısı arasındaki ilişki 42
Şekil 4.10. Üç yaklaşımın Medyan hatası ile küme sayısı arasındaki ilişki 43

TABLO LİSTESİ

Sayfa No

Tablo 3.1. İşlenmiş Veri Seti Örneği 22

Tablo 4.1. kNN algoritmasında kullanılan değişkenler ve değerleri 26

Tablo 4.2. kNN algoritması ile elde edilen ortalama hata, medyan hata ve $R2$ değerleri 27

Tablo 4.3. Ağırlıklı kNN algoritmasında kullanılan değişkenler ve değerleri 28

Tablo 4.4. WKNN algoritması ile elde edilen ortalama hata, medyan hata ve $R2$ değerleri ... 29

Tablo 4.5. Rastgele Orman algoritmasında kullanılan değişken ve değeri 30

Tablo 4.6. Rastgele Orman algoritması ile elde edilen ortalama hata, medyan hata ve $R2$ değerleri 31

Tablo 4.7. Gradyan artırma algoritmasında kullanılan değişkenler ve değerleri 33

Tablo 4.8. Gradyan Artırma algoritması ile elde edilen ortalama hata, medyan hata ve $R2$ değerleri 33

Tablo 4.9. AdaBoost algoritmasında kullanılan değişkenler ve değerleri 35

Tablo 4.10. AdaBoost algoritması ile elde edilen ortalama hata, medyan hata ve $R2$ değerleri 36

Tablo 4.11. Tüm algoritmalar ile elde edilen ortalama hata ve medyan hata karşılaştırılma tablosu 38

Tablo 4.12. Tüm algoritmalarda kullanılan değişkenler ve değerleri 38

Tablo 4.13. Hibrit yöntemler ile elde edilen ortalama hata, medyan hata ve $R2$ değerleri 39

Tablo 4.14. Rastgele Orman ve AdaBoost algoritmalarda kullanılan değişkenler ve değerleri 39

Tablo 4.15. AdaBoost kullanarak önerilen metodoloji ile elde edilen ortalama hata, medyan hata ve $R2$ değerleri 40

Tablo 4.16. Rasgele Orman kullanarak önerilen metodoloji ile elde edilen ortalama hata, medyan hata ve $R2$ değerleri 40

Tablo 4.17. Hibrit yöntem kullanarak önerilen yöntem ile elde edilen ortalama hata, medyan hata ve $R2$ değerleri.....41

SİMGE VE KISALTMA LİSTESİ

Simgeler **Açıklama**

€ : Euro (Euro)
dB : Desibel (Decibel)
d : Uzaklık (distance)
km : Kilometre (Kilo Meter)
m : Metre (Meter)
mA : Miliamper (Milliampere)
mAh : Miliamper-saat (Milliampere hour)
MHz : Megahertz (MegaHertz)

Kısaltmalar **Açıklama**

AoA : Varış Açısı (Angle of Arrival)
BS : Baz İstasyonu (Base Station)
BLE : Bluetooth Düşük Enerji (Bluetooth Low Energy)
GPS : Küresel Konumlama Sistemi (Global Positioning System)
GBM : Gradyan Artırma Makinesi (Gradient Boosting Machine)
HDOP : Yatay Hassasiyet Seyreltmesi (Horizontal Dilution Of Precision)
IoT : Nesnelerin İnterneti (Internet of Things)
ISM : Bilimsel ve Tıbbi Cihaz Bandı (Industrial Scientific Medical Band)
IDE : Tümleşik Geliştirme Ortamı (Integrated Development Environment)
kNN : K-En Yakın Komşu (K-Nearest Neighbor)
LPWAN : Düşük Güçlü Geniş Alan Ağı (Low Power Wide Area Network)
LoRaWAN : Uzun Menzilli Geniş Alan Ağı (Long Range Wide Area Network)
LoRa : Long Range (Uzun Menzilli)
MAC : Ortam Erişim Kontrolü (Medium Access Control)
NB-IoT : Dar Bant Nesnelerin İnterneti (Narrow Band-Internet of Things)
RSSI : Alınan Sinyal Gücü Göstergesi (Received Signal Strength Indicator)
RSS : Alınan Sinyal Gücü (Received Signal Strength)
RX : Alınan Zaman (Received Time)

R^2	: Belirtme Katsayısı (Coefficient of Determination)
RF	: Radyo Frekansı (Radio Frequency)
SF	: Yayılma Faktörü (Spreading Factor)
SVM	: Destek Vektör Makinesi (Support Vector Machine)
ToA	: Varış Zamanı (Time of Arrival)
TDoA	: Varış Zamanında Meydana Gelen Fark (Time Difference of Arrival)
TOF	: Uçuş Süresi (Time of Flight)
TTN	: Nesneler Ağı (The Things Network)
T	: Eşik (Threshold)
WKNN	: Ağırlıklı K-En Yakın Komşu (Weighted K-Nearest Neighbor)
Wi-Fi	: Kablosuz Bağlantı Alanı (Wireless Fidelity)
WCSS	: Küme İçinde Kareler Toplamı (Within-Cluster Sum of Square)
CRC	: Döngüsel Artıklık Denetimi (Cyclic Redundancy Check)

ÖZET

YÜKSEK LİSANS TEZİ

AKILLI ŞEHİRLER İÇİN LORAWAN AĞLARINDA MAKİNE ÖĞRENİMİ TABANLI DIŞ MEKAN KONUMLANDIRMA TEKNİKLERİNİN PERFORMANS ANALİZİ

Gufran BAYRAKTAR

İstanbul Üniversitesi-Cerrahpaşa

Lisansüstü Eğitim Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Danışman: Dr. Öğr. Üyesi Gülsüm Zeynep GÜRKAŞ AYDIN

Konumlandırma, birçok Nesnelerin İnterneti (IoT) uygulamasında temel bir unsurdur. Bu tür uygulamalarda LoRaWAN gibi bir Düşük Güçlü Geniş Alan Ağı (LPWAN) teknolojisinin kullanılması güvenilir mobil iletişim ve düşük güç tüketimi sağlamasına rağmen, GPS alıcılarının fazla güç tüketimi nedeniyle konum belirleme için kullanılması çok elverişli değildir. Bu nedenle LoRaWAN'a dayalı konumlandırma için alınan sinyal gücü (RSSI) ölçümlerinin kullanılması, LoRaWAN'ın hem iletişim hem de konumlandırma için kullanıldığı uygulama ve alanlarda avantaj sağlamıştır. Bu tezin çalışmasının amacı, farklı makine öğrenmesi algoritmaların performansını LoRaWAN tabanlı bir sistemde test etmek ve karşılaştırma yapmaktadır. Ayrıca diğer bir amaç da LoRaWAN ve konumlandırma tabanlı uygulamaların güç tüketimini azaltmak için bir cihazın konumunu tahmin edebilen efektif bir sistem önermektedir. Bu amaçların ışığında bu tez çalışması üç kısımdan oluşmaktadır: Birinci kısımda, konum tahmini için beş makine öğrenmesi algoritması karşılaştırılmıştır. Bu algoritmalar Rastgele Orman, Gradyan Artırma, AdaBoost, K- En Yakın Komşu ve ağırlıklı K-En Yakın Komşu algoritmalarıdır. İkinci kısımda farklı hibrit yöntemler değerlendirilmiş ve karşılaştırılmıştır. Üçüncü kısımda, önceki çalışmalardan yola çıkılarak yeni bir metodoloji önerilmiştir. Yöntemler, Belçika'nın Antwerp kentindeki kentsel alanlarda toplanan halka açık bir LoRaWAN veri seti kullanılarak değerlendirmiştir. Bu tez çalışmasının ilk kısmında elde edilen 255 metre ortalama hata ve 117 metre medyan hata, ikinci kısımda sırasıyla 254 metre ortalama hata ve 137 metre medyan hata ve üçüncü kısımda, önerilen metodolojinin en iyi parametrelerle test edilmesi ve performansının incelenmesi sonucunda, bu çalışmada kullanılan veri setini kullanan önceki çalışmalara kıyasla en düşük medyan hata değeri elde edilmiştir. Önerilen metodolojide 256 metrelik bir ortalama hata ve 97 metrelik bir medyan hata elde edilmiştir.

Temmuz 2022 , 74 sayfa.

Anahtar kelimeler: LoRaWAN, Parmak İzi Algoritması, Makine Öğrenmesi, Dış Mekan konumlandırma

ABSTRACT

M.Sc. THESIS

PERFORMANCE ANALYSIS OF MACHINE LEARNING BASED OUTDOOR POSITIONING TECHNIQUES IN LORAWAN NETWORKS FOR SMART CITIES

Gufran BAYRAKTAR

İstanbul University-Cerrahpaşa

Institute of Graduate Studies

Department of Computer Engineering

Computer Engineering Programme

Supervisor: Assist. Prof. Dr. Gülsüm Zeynep GÜRKAŞ AYDIN

Localization is a fundamental element in many Internet of Things (IoT) applications. Although using Low Power Wide Area Network (LPWAN) technology such as LoRaWAN in such applications ensures low-power consumption with reliable mobile communication, using a GPS for localization is considered as impractical due to the power consumption of a GPS receiver. Therefore, the use of received signal strength (RSSI) measurements for LoRaWAN-based localization has proved advantageous in applications and areas where LoRaWAN is used for both communication and localization. The aim of this thesis study is to test and compare the performance of different machine learning algorithms in a LoRaWAN based system. Another aim is to propose an effective system that can predict a device's location to reduce the power consumption of LoRaWAN and localization based applications.

In the light of these aims, this thesis study consists of three parts: In the first part, five machine learning algorithms for location estimation are compared. The algorithms are Random Forest, Gradient Boost, AdaBoost, kNN and WKNN. In the second part different hybrid methods are evaluated and compared. In the third part, a new methodology is proposed based on the previous work. The methods are evaluated using a publicly available LoRaWAN dataset collected in urban areas in Antwerp city in Belgium. The mean and the median error obtained in the first part of this thesis study are 255 meters and 117 meters respectively and in the second part are 254 meters and137 meters respectively and in the third part, as a result of testing the proposed methodology with the best parameters and examining its performance, The lowest median error value was obtained compared to previous studies using the same data set used in this study. In

the proposed methodology, a mean error of 256 meters and a median error of 97 meters were obtained.

July 2022, 74 pages.

Keywords: LoRaWAN, fingerprinting algorithm, Machine Learning, outdoor localization

1. GİRİŞ

Nesnelerin interneti (IoT) ekosistemi en temelde cihazlar ve nesneler arasındaki bağlantıyı ve veri alışverişini ifade eder. Temel teknolojilerdeki hızlı gelişmeler sayesinde nesnelerin interneti, yaşam kalitemizi iyileştirmeyi vaat eden yeni uygulamalar için önemli bir altyapı ve servis sunmaktadır. Nesnelerin interneti, dünya çapındaki uygulayıcılar ve araştırmacılardan büyük ilgi görmüş ve kapsamında çok çeşitli uygulamalar ortaya çıkmıştır.

IoT, nesneler arasındaki bağlantıyı sağlamak için farklı iletişim teknolojileri kullanmaktadır, ancak cihazlarda bulunan kısıtlamalar ve geleneksel kablosuz ağların nesnelerin interneti uygulamalarının tüm ihtiyaçlarını karşılayamaması sebebiyle, nesnelerin interneti uygulamalarının gereksinimleri, düşük güçlü geniş alan ağı (LPWAN) isimli yeni bir kablosuz iletişim teknolojisinin ortaya çıkmasına neden olmuştur.

"LPWAN" terimi 2013 yılı başlarında ortay çıkmıştır [1]. LPWAN, düşük güç, uzun menzil ve düşük maliyetli iletişim özellikleri nedeniyle endüstride hızla daha popüler hale gelmiştir. Kırsal bölgelerde 10-40 km'ye ve kentsel bölgelerde 1-5 km'ye kadar uzun menzilli iletişim sağlamaktadır [2]. Buna ek olarak, yüksek enerji verimlidir (yani 10 yıldan fazla pil ömrü [3]) ve ucuzdur [4]. LPWAN'ın bu umut verici yönleri, LPWAN'ın dış ve iç ortamlardaki performansına ilişkin deneysel çalışmaları harekete geçirmiştir [5], [6]. LPWAN, uzun menzilde yalnızca küçük miktarlarda veri iletmesi gereken nesnelerin interneti uygulamaları için son derece uygun bir teknoloji olarak görülmeye başlanmıştır. Lisanslı ve lisanssız frekans bant genişliğinde birçok LPWAN teknolojisi ortaya çıkmıştır. Bunların arasında Sigfox, LoRa ve NB-IoT, [7, 8, 9] birçok teknik farklılık içeren ve günümüzün önde gelen gelişen teknolojilerindendir [5].

LoRaWAN, LoRa Alliance tarafından standartlaştırılmış bir Uzun Menzilli Geniş Alan ağ protokolüdür. Semtech firmasının LoRa modülasyon şemasıyla kullanılan ağ için sistem mimarisini ve iletişim protokolünü tanımlar [10]. Önde gelen üç LPWAN teknolojisi olan Sigfox, LoRaWAN ve NB-IoT üzerinde yapılan karşılaştırmalı bir çalışmada, LoRaWAN'ın düşük maliyet, uzun pil ömrü ve güvenilir mobil iletişim özellikleriyle konum tabanlı uygulamalarda en iyi seçenek olduğunu göstermektedir [9].

Başka bir deneysel çalışma, LoRaWAN'ın yayılma faktörü (spreading factor - SF), cihaz sınıfı ve iletilen paket boyutu ile ilgili bazı belirli koşullara bağlı olarak, 2000 mAh kapasiteli bir pilin 10 yıl kullanılabileceğini göstermiştir [11]. Bir GPS alıcısının yaklaşık 30-50 mA olan güç tüketimi nedeniyle, GPS'in konum tabanlı uygulamalarda kullanımı pratik değildir [12]. Bu nedenle, düşük güçlü cihazlarda konumlandırma için alternatif yaklaşımlara ihtiyaç bulunmaktadır. Bu amaçla, tek bir teknolojinin kullanımı hem iletişim hem de konumlandırma yönünden daha efektif olacaktır. LoRaWAN'da cihazlar, kentsel ve kırsal alanlarda dağıtılmış sabit baz istasyonları ile radyo frekansı (radio frequency – RF) mesajları aracılığıyla iletişim kurar. Bu mesajlar, uygulamalara özel olarak toplanması gereken bilgileri içerir. Ayrıca, bu sinyal iletimlerinde ve alımlarında, konumlandırma için kullanılabilecek ek bilgiler vardır. Konum tahminleri üretmek için Alınan Sinyal Gücü Göstergesi (RSSI) veya Varış Zaman Farkı (TDoA) gibi ölçümler kullanılabilmektedir. Sinyaller ya geleneksel yöntemlerle (multilaterasyon, trilaterasyon, vb.) ya da parmak izi yöntemleriyle işlenebilmektedir [13]. Son zamanlarda, LoRaWAN'a dayalı alınan sinyal gücü (RSS) parmak izi yaklaşımları popülerlik kazanmıştır [13, 14].

Bu tez çalışmasında, RSSI değerlerine dayalı konum tahmini için beş farklı algoritma arasında bir karşılaştırma yapılmış, ardından hibrit konum tahmin yöntemlerinin bir değerlendirmesi yapılmıştır. Ayrıca, hibrit yöntemlerin sonuçlarına bağlı olarak, konumu tahmin etmek için farklı yaklaşımlar içeren yeni bir metodoloji önerilmektedir.

İkinci bölümde konumlandırma hakkında bir genel bakış verilmiş, LoRaWAN ve LoRaWAN'da konumlandırma ile bu konulardaki ilgili çalışmalar sunulmuştur. Üçüncü bölümde tezde kullanılan algoritmalar ve yöntemler hakkında bilgi verilmektedir ve kullanılan veri seti, veri setinin ön işlemesi, geliştirme ortamı ve önerilen yöntemin iş akışı açıklanmaktadır. Dördüncü bölümde ise tüm sonuçlar üç başlıkta sunulmuştur. Birinci bölümde kullanılan beş algoritma ve aralarında yapılan karşılaştırmanın sonuçları, ikinci bölümde hibrit yöntemlerin sonuçları ve üçüncü bölümde ise önerilen metodolojinin sonuçları sunulmaktadır. Beşinci bölümde sonuçlar değerlendirmesini ve altıncı bölüm de çalışmanın sonucunu ve gelecek çalışmalara dair önerileri sunmaktadır.

2. KAVRAMSAL ÇERÇEVE

Kablosuz ağlarda konumlandırma, kablosuz iletişim teknolojilerini kullanarak otomatik bir şekilde fiziksel konum bulma sürecidir. Bu bölümde genel olarak konumlandırma, LoRaWAN ve LoRaWAN'da konumlandırma ile ilgili bazı çalışmalar sunulmuş ve bu tez çalışmasında kullanılan yöntem ve teknikler anlatılmıştır.

Bölüm 2.1'de geleneksel konumlandırma teknikleri hakkında bilgi verilmiş ve farklı teknolojiler kullanılarak konumlandırma ile ilgili bazı çalışmalar tartışılmıştır. Ardından Bölüm 2.2'de LoRaWAN teknolojisi ve mimarisi hakkında genel bir bakış sunulmuş ve ilgili çalışmalara yer verilmiştir. Son olarak, Bölüm 2.3'te LoRaWAN'da konumlandırma hakkında bir genel bakış sunulmuş ve LoRaWAN'da geleneksel konumlandırma yöntemleri ve parmak izi yöntemleri kullanılarak konumlandırma ile ilgili çalışmalar tartışılmıştır.

2.1. KONUMLANDIRMA

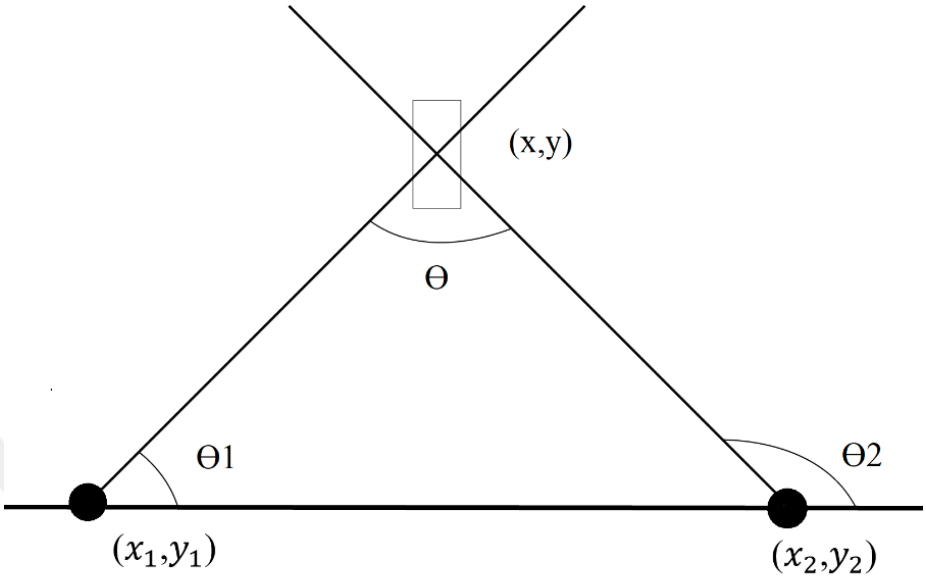
Konumlandırma denilince akla ilk olarak küresel konumlandırma sistemi olan GPS (Global Positioning System) gelmektedir ancak zamanla GPS'e bağlı yaşanan zorluklar nedeniyle bir nesnenin yerini belirlemek için geliştirilmiş birçok yeni teknoloji ve teknik geliştirilmiştir. Bu bölümde konumlandırmanın bazı temel ilkeleri ve en sık kullanılan geleneksel konumlandırma teknikleri anlatılmaktadır.

2.1.1. Geleneksel Konumlandırma Teknikleri

Bir cihazın konumunu belirlemek için öncelikle, çeşitli ölçümlere ek olarak dayanak noktaları (anchor points) olarak adlandırılan bilinen koordinatlara sahip bazı referans noktalarına ihtiyacımız bulunmaktadır. Bu dayanak noktalarını ve ölçümleri kullanarak, bir cihazın konumunu mümkün olduğunca kesin olarak tahmin etmek için kullanılabilecek çeşitli teknikler vardır. Bu tekniklerden en temel olanları üçgenleme, trilaterasyon, multilaterasyon ve parmak izi yöntemleridir.

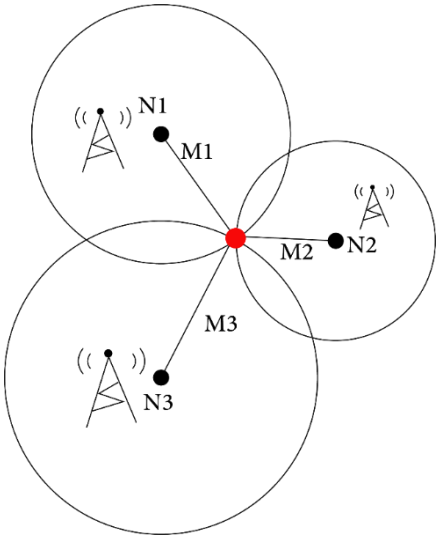
Üçgenleme yönteminde, bir cihazın konumu, cihaz ile iki dayanak noktası arasında oluşturulan üçgenler kullanılarak tahmin edilebilir. Her dayanak noktasından, Şekil 2.1 'da gösterildiği gibi

cihaza olan açıyı hesaplanmaktadır. Bu iki açı Varış Açısı (AoA) kullanılarak hesaplanabilmekte ve mesafe de bu iki açı ve iki dayanak noktası arasındaki mesafe ile trigonometrik formüller kullanılarak hesaplanabilmektedir.



Şekil 2.1. Üçgenleme Yöntemi

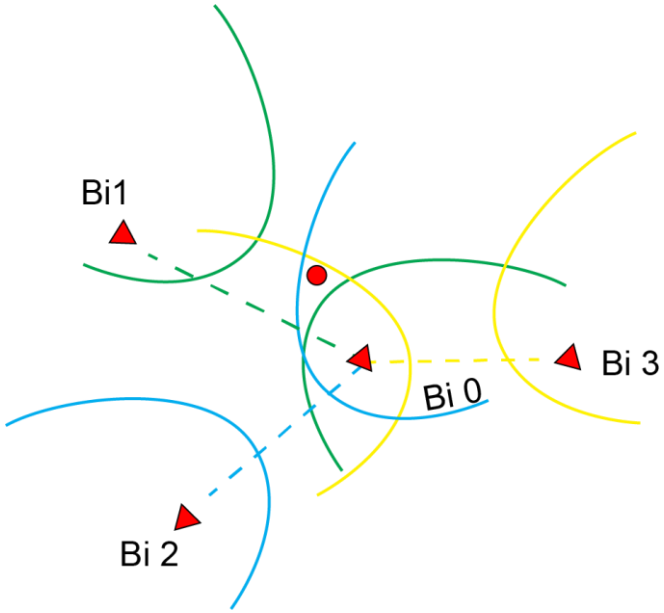
Trilaterasyon yöntemi, konumu belirlenmeye çalışılan cihaz ile üç veya daha fazla dayanak noktası arasındaki mesafeyi kullanan bir tekniktir. Bu yöntemde Şekil 2.2'de gösterildiği gibi üç adet daire olduğu kabul edilir. Bu dairelerin merkezleri dayanak noktaları olarak kullanılır. Her dairenin yarıçapı dayanak noktası ile cihaz arasındaki mesafedir. Cihazın konumu, Şekil 2.2'de gösterildiği gibi farklı mesafelerden elde edilen üç dairenin kesişim noktasıdır. Trilaterasyon yönteminde mesafeler çeşitli ölçüm türleri kullanılarak tahmin edilebilir: Bunlar Varış Zamanı (ToA) ölçümleri, Alınan İşaret Güç Göstergesi (RSSI) veya Uçuş Süresi (TOF) ölçümleridir. Bu nedenle, verici ve alıcı arasında senkronizasyon gerektirmektedirler.



Şekil 2.2. Trilaterasyon Yöntemi

Uygulanmasının basit olması trilaterasyon yöntemini en popüler konumlandırma tekniklerinden biri haline getirmiştir ancak gürültülü ortamlarda yüksek gürültü mesafesi ölçüm hataları kaçınılmazdır ve bu da yöntemin kullanımında ve ölçümlerde dezavantaj haline gelebilmektedir.

Hiperbolik Lateralasyon olarak da bilinen Multilaterasyon yöntemi, bir cihazdan üç veya daha fazla alıcıya yayılan bir sinyalin Şekil 2.3'te gösterildiği gibi TDOA varış zaman farkını doğru bir şekilde hesaplayarak bir cihazın konumunu belirleme tekniğidir [15].



Şekil 2.3. Multilaterasyon Yöntemi

Parmak İzi yöntemi, bir makine öğrenimi tekniğidir. Çevrimdışı eğitim ve çevrimiçi konum belirleme olmak üzere iki aşamada çalışmaktadır. Eğitim aşaması sırasında, bilinen konumlardaki sinyal gücü ölçümleri bir veri setinde toplanır. Çevrimiçi aşamasında, bir cihazın konumu, bir konum algoritması kullanılarak eğitim veri seti ile gerçek zamanlı sinyal gücü ölçümlerinin karşılaştırılmasıyla belirlenir [16]. En çok bilinen Parmak İzi teknikleri üçüncü bölümde ayrıntılı bir şekilde sunulmuştur.

Yukarıda bahsedildiği gibi, konum belirleme özet olarak, önce dayanak noktalarına ve çeşitli ölçümlere sahip olarak ve ardından konumu tahmin etmek için bir konumlandırma algoritması kullanılarak yapılır. Konum tahmininin performansı, ölçüm özelliklerine ve doğruluğuna büyük ölçüde bağlıdır.

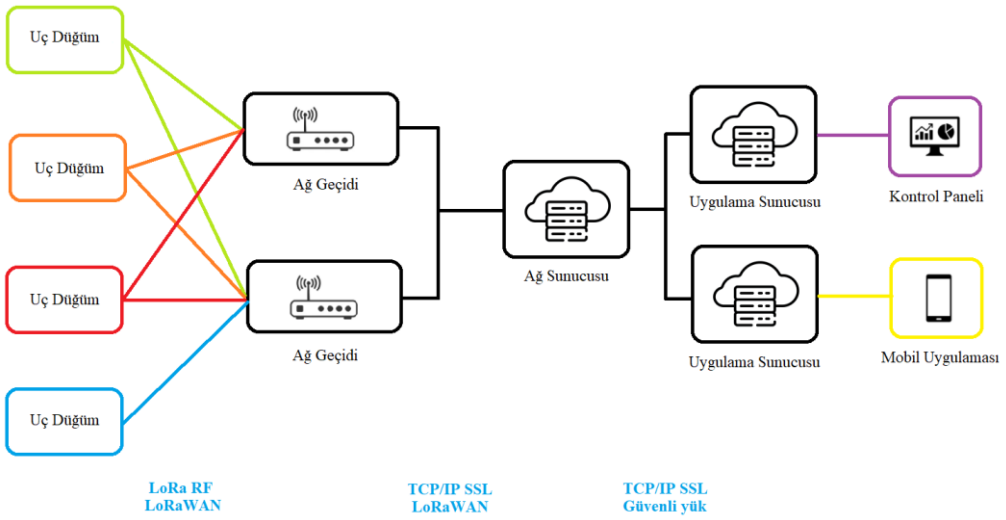
Geleneksel algoritmalar ve Parmak İzi teknikleri ister iç mekan ister dış mekan konum tahmininde çok çeşitli çalışmalarda kullanılmıştır. [17]'de Haznedar, iç mekan konum tahmini için bir yöntem önermek için Wi-Fi ve BLE teknolojilerini kullanmıştır. Bu tez çalışmasının ilk kısmında geleneksel doğrusal olmayan en küçük kare algoritması kullanılarak doğruluk oranları ortalama %72- %83, ikinci kısımda makine öğrenmesinde kNN ve SVM kullanıldığında %87 ile %88 arasında elde edilmiştir. [18]'de Anagnostopoulos ve arkadaşları ön işleme adımlarının uygun seçimini analiz etmek ve bir kNN parmak izi yönteminin hiper parametrelerini ayarlamak için bir Sigfox ağında yayınlanmış bir RSSI ölçümleri veri seti kullanmıştır. Sonuç olarak, ortalama ve medyan hataları sırasıyla 298 metre ve 109 metre olarak elde etmişlerdir.

2.2. LORAWAN

LPWAN, düşük güçlü geniş alan ağı anlamına gelmektedir. Küçük boyutlu veri paketlerini düşük güç tüketimi ile uzun mesafelere göndermek amacıyla internete bağlı ve batarya ile çalışan cihazlar için tasarlanmış bir kablosuz haberleşme ağı türüdür. Bu alanda en yaygın teknolojiler LoRa, Sigfox ve Nb-IoT'dir. LoRaWAN, teknoloji sektöründeki en büyük ittifaklardan biri haline gelen açık bir birlik olan LoRa Alliance tarafından standartlaştırılmış bir uzun menzilli geniş alan ağ protokolüdür [19]. Semtech firmasının LoRa modülasyon şemasıyla kullanılan ağ için sistem mimarisini ve iletişim protokolünü tanımlar [10] uygun maliyetli ağ dağıtımları için lisanssız ISM radyo bantları kullanır. LoRaWAN ağlarında kullanılan frekans planı, yerel düzenlemelere göre farklılık gösterir. Örneğin; Türkiye'de lisanssız 433 MHz ve 868 MHz frekans bantlarında faaliyet göstermektedir.

2.2.1. LoRaWAN Mimarisi

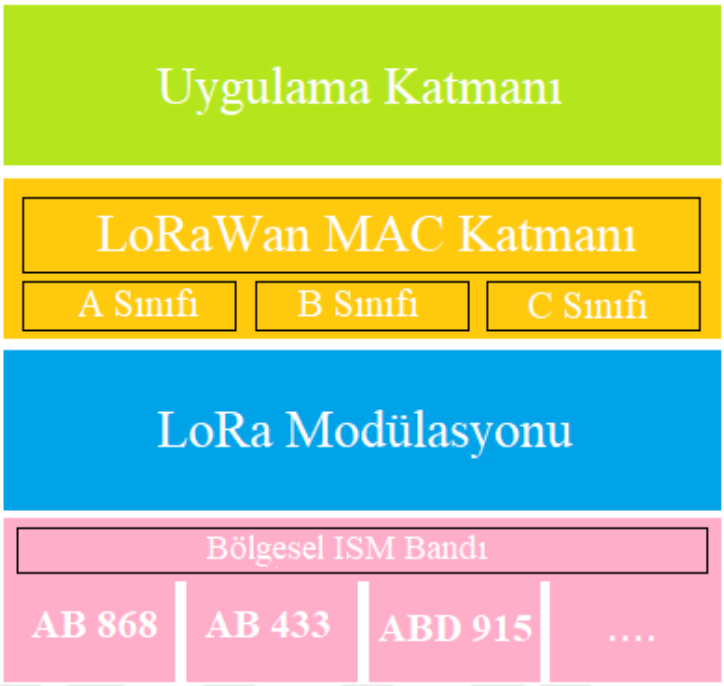
LoRaWAN, uçtan uca güvenlik, hizmet kalitesi ve hareketlilik dahil olmak üzere temel nesnelerin interneti gereksinimlerini kapsamaktadır.



Şekil 2.4. LoRaWAN mimarisi.

LoRaWAN, Şekil 2.4'te gösterildiği gibi yıldızların yıldızı (star of stars) topolojisini kullanır. Ağ geçitleri; düğümler ve bulut tabanlı ağ sunucusu arasında aracı olarak davranır. Bir düğüm tarafından bir veri iletildiğinde, o düğümün menzilindeki tüm ağ geçitleri verilerini alır ve ağ sunucusuna iletir. Düğümler, LoRaWAN protokolü aracılığıyla LoRaWAN Ağ Geçitleri ile iletişim kurar. Ağ geçitleri ve ağ sunucusu arasındaki bağlantı, hücresel, Ethernet, uydu veya Wi-Fi gibi daha yüksek bir bant genişliği iletişim protokolü aracılığıyla gerçekleştirilir. Bir ağ geçidi bir mesaj aldığında, yalnızca verilerin bütünlüğünü kontrol etmektedir. Döngüsel artıklık denetiminin (cyclic redundancy check - CRC) değeri yanlışsa, mesaj ihmal edilir. Aksi durumda ağ geçidi, mesajı, mesajın alınan sinyal gücü göstergesi seviyesini ve bir zaman damgasını içeren meta verileri ağ sunucusuna iletecektir.

Birden fazla ağ geçidi, bir LoRa düğümünden aynı mesajı alabildiği için, ağ sunucusu çoğaltılan paketleri filtreler, ardından mesaj ağ sunucusundan uygun uygulama sunucusuna gönderilir [10]. LoRaWAN, farklı uygulama türlerine hizmet edebilmek için Şekil 2.5'te gösterildiği gibi üç cihaz sınıfını desteklemektedir: A Sınıfı, B Sınıfı ve C Sınıfı. Tüm sınıflar çift yönlü iletişimi (yukarı bağlantı (uplink) ve aşağı bağlantı (downlink)) desteklemektedir



Şekil 2.5. LoRaWAN Teknolojisinin Katmanlı Mimarisi

A Sınıfı Cihazlar: Tüm LoRaWAN uç cihazları bu sınıf tarafından desteklenmelidir. B sınıfı ve C Sınıfı cihazlar, A Sınıfı cihazların uzantılarıdır. En basit sınıftır ve zamanının çoğunu uyku modunda geçirdiği için üç sınıfın içinde enerji verimliliği en yüksek olan sınıftır. A Sınıfı cihaz herhangi bir zamanda bir yukarı bağlantı mesajı gönderebilir ve bunu iki kısa aşağı bağlantı alma penceresi takip eder. Genellikle, alma penceresinin süresi bir saniyedir. Sunucu, iki kısa alım penceresinden herhangi biri sırasında bir aşağı bağlantı göndermediyse, uyku moduna geri döner. Bir sonraki aşağı bağlantı sadece bir sonraki yukarı bağlantı iletiminden sonra olacaktır, bu nedenle A Sınıfı cihazlar aktüatör rolünde uygun değildir.

B Sınıfı Cihazlar: B Sınıfı cihazlar, ağ geçitleri tarafından gönderilen, zamanla senkronize edilmiş beaconları kullanarak A sınıfındaki ilk mesajlara ek olarak sunucudan aşağı bağlantı mesajlarını almak için alma pencerelerini planlar. B sınıfı cihazlarda pil tüketimi A Sınıfına göre daha fazladır çünkü bu cihazlar beacon ve ping slotları sırasında uyku modunda daha az zaman harcar. Ancak, güç tüketimi, pille çalışan uygulamalar için hala yeterince düşüktür.

C Sınıfı Cihazlar: Bu sınıftaki cihazlar A Sınıfı yapıya ek olarak, cihazın alıcısını iletim süresi dışında herhangi bir zamanda sunucudan aşağı bağlantı mesajı almaya uygun şekilde açık tutar. Bu özellik, düşük gecikmeli iletişime izin verir, ancak yüksek güç tüketimine neden olabilir.

LoRa, Chirp Spread Spectrum (CSS) teknolojisinden türetilen yayılmış spektrum modülasyon teknolojisini kullanır. Ek olarak, LoRa ortogonal yayılma faktörleri kullanır ve SF7'den SF12'ye kadar altı yayılma faktörüne sahiptir [8].

Yayılma faktörü ne kadar yüksek kullanılırsa, sinyaller o kadar uzağa gidebilmektedir. Bir ağ geçidine yakın uç düğümler daha düşük bir yayılma faktöründe veri iletmelidir, ancak bir ağ geçidinden uzaktaki uç düğümlerin daha yüksek bir yayılma faktöründe veri iletmesi gerekir [19]. Daha yüksek yayılma faktörleri, bir uç düğümde daha az veri boyutu ve daha kısa pil ömrü sağlar.

2.2.2. LoRaWAN ile İlgili Çalışmalar

LoRaWAN, Endüstriyel Nesnelerin İnterneti (Industrial Internet of Things- IIoT) uygulamalarının çoğu için uygun bir teknolojidir çünkü Wi-Fi gibi daha az yaygın bağlantı çözümleri ve hücresel bağlantılar gibi daha pahalı bağlantı türleri için bir alternatif sunmaktadır. LoRaWAN'ın en yaygın kullanıldığı alanlar akıllı şehir, akıllı bina, akıllı sağlık, akıllı tarım ve akıllı ulaşım uygulamalarıdır.

Akıllı şehirlerde, sistemdeki farklı parametreleri izlemek ve kontrol etmek için çeşitli uygulamalar mevcuttur. Örneğin [20]'de Chaudhari ve arkadaşları kanalizasyon sistemlerindeki rögar kapaklarının konumunu, sökülmesini veya zarar görmesini izlemek için bir ivme ölçer sensörü kullanmıştır. Bu kapaklar hareket ettirildiğinde veya zarar gördüğünde, LoRa kartı, bulut tabanlı bir açık kaynaklı LoRaWAN platformu olan The Things Network (TTN) ile ilişkili en yakın ağ geçidini uyararak sinyali iletir. TTN, sivil yetkililerin destek bölümüne hataların düzeltilmesini talep etmek amacıyla bir alarm gönderir.

Tarım alanında, Zhao ve arkadaşları [21] 'de LoRaWAN tabanlı bir akıllı sulama sistemi önermişlerdir. Önerdikleri sistemde, bir hidroelektrik jeneratörü ile donatılmış sulama düğümü onlarca yıl çalışabilmektedir. Sulama düğümü ile ağ geçidi arasındaki iletişim mesafesi 8 km'dir, bu nedenle önerdikleri sistem 200 hektara kadar geniş bir alanı kapsayabilmektedir.

Kodali ve arkadaşları [22] 'de bir akıllı çiftlik izleme modeli sunmuşlarıdır. Bu model, verici düğümden alıcı düğüme sıcaklık, nem ve toprak nemi bilgilerini göndermek için LoRaWAN kullanmıştır. Önerdikleri sistem, LoRaWAN teknolojisinin benimsenmesinin geleneksel ağların tüm dezavantajlarının üstesinden gelmek için en uygun olduğunu başarılı bir şekilde göstermektedir.

Madencilerin güvenliği düşüncesinin yanında madenciliğin maliyetini azaltmak için Porselvi ve arkadaşları [23]'de maden işçisinin durumunu ve madendeki çeşitli duman seviyelerini izleyen bir sistem önermiştir. Bu sistemde kriz durumunda LoRaWAN modülü yöneticiye alarm verebilmektedir.

2.3. LORAWAN İLE KONUMLANDIRMA

LoRaWAN, çok çeşitli konum tabanlı uygulamaları kapsayan LPWAN ağları için bir coğrafi konum çözümünü desteklemektedir. Tüm LoRaWAN uç cihazları, ek maliyet veya işlem gücü olmadan bu yeteneğe sahiptir.

LoRaWAN'da coğrafi konumu belirlemek için iki tür yaklaşım vardır:

- Alınan Sinyal Gücü Göstergesi (RSSI) tabanlı
- Daha iyi doğruluk için Varış Zaman Farkı (TDOA) tabanlı

2.3.1. LoRaWAN TDOA Tabanlı Konumlandırma

Bir LoRaWAN ağ geçidi, bir yukarı bağlantı mesajı aldığında, ona meta verinin bir parçası olarak bir zaman damgası verir ve ardından mesajı ağ sunucusuna iletir. Ağ geçitleri, doğru bir zaman senkronizasyonu sağlamalıdır ve bu, her ağ geçidinde bir GPS modülünün bulunmasıyla sağlanmaktadır.

Bir uç cihazın konumunu belirlemek için, yukarı bağlantı mesajının en az üç veya daha fazla ağ geçidine ulaşması gerekmektedir. Mesajı alan bir çift ağ geçidi için varış zamanı farkını hesapladıktan sonra, uç cihazın konumu, multilaterasyon tekniği kullanılarak belirlenebilir. Her varış zaman farkı değeri bir hiperbolü temsil eder ve uç cihaz bu hiperbol üzerine yerleştirilebilir. Bu hiperbollerin kesişiminin bulunmasıyla uç cihazın konumu belirlenebilir. Yayılma ortamı, çoklu yol, ağ geçidi yoğunluğu ve kullanılan yerelleştirme algoritması gibi konumun doğruluğu üzerinde önemli rol oynayan birkaç faktör bulunmaktadır [24].

2.3.2. LoRaWAN RSSI Tabanlı Konumlandırma

Yalnızca RSSI ölçümlerinin mevcut olduğu durumlarda, hedef düğümün konumunu tahmin etmek için RSSI tabanlı bir algoritma kullanılabilmektedir. Wi-Fi, BLE ve GPS gibi diğer teknolojilerle karşılaştırıldığında, LoRaWAN, geleneksel algoritmaları kullanırken TDOA kullanarak 20 ila 200 metre ve RSSI kullanarak 1000 ila 2000 metre arasında bir aralıkta

doğruluğun yanı sıra çevre koşullarına da bağlı olarak etkin bir maliyet güç verimliliği sağlar [24].

2.3.3. LoRaWAN Konumlandırma ile İlgili Çalışmalar

LoRaWAN'da GPS kullanarak konumlandırma ve izleme geçekleştiren çalışmalar da mevcuttur. [25]'te Silva ve arkadaşları iletilecek koordinatları sağlamak için yalnızca bir baz istasyonu ve bir GPS cihazı kullanarak konumlandırma ve izleme sağlamak amacıyla Uzun Menzilli (LoRa) modülasyon protokolüne dayalı bir çözüm önermiştir. Başka bir çalışmada, Guan ve arkadaşları otobüslerin dinamik konumunu ve beklenen varış zamanını belirlemede yardımcı olabilecek LoRaWAN'a dayalı gerçek zamanlı bir konumlandırma sistemi tasarlamış ve uygulamışlardır [26].

GPS yüksek doğrulukta sonuçlar verebilmekte ancak daha yüksek güç tüketimine neden olmaktadır. Bu nedenle, LPWAN teknolojisini kullanmanın temel avantajlarından biri konumlandırma ve iletişimde enerji tüketimini azaltmaktır.

2.3.3.1. Geleneksel Konumlandırma tekniklerini kullanarak LoRaWAN ile Konumlandırma

Fargas ve arkadaşları [12]'da GPS kullanmadan bir cihazın şimdiki konumunu hesaplayabilen bir LoRaWAN takip sistemi önermiştir. Konumu tahmin etmek için, yinelemeli olmayan ve yinelemeli olmak üzere TDoA tabanlı iki konumlandırma algoritması önermişlerdir. Bu iki algoritma ile bir cihazın statik bir noktadaki konumunu yaklaşık 100 metrelik bir doğrulukla tahmin edebilmişlerdir.

Plet ve arkadaşları [27] 'de, konum tahmini için beş farklı algoritma kullanmış ve ardından bir yol haritası filtresi uygulamışlardır. RSSI tabanlı algoritmalar sırasıyla 1000 m ve 2000 m medyan ve maksimum hata elde ederken, TDoA tabanlı algoritmalar sırasıyla 150 m ve 350 m civarında bir medyan ve maksimum hata elde etmişlerdir.

200 m'lik bir menzilde konum doğruluğu bazı durumlarda yeterli olmayabilmektedir. [28] 'da Podevijn ve arkadaşları LoRa düğümüne bir e-pusula eklenmiş ve ağ geçitlerine yol tarifi göndererek ve bu verileri TDoA değerleriyle bir harita eşleştirme algoritmasında birleştirerek 8 kat daha doğru tahminler elde etmeyi başarmışlardır. Fakat, e-pusula kullanımı daha fazla güç tüketebilmektedir ve TDOA konumlandırmada, tüm ağ geçitleri, ekstra donanım gerektiren doğru bir zaman senkronizasyonu sağlamalıdır.

2.3.3.2. Makine öğrenmesi kullanarak LoRaWAN ile Konumlandırma

Dış mekan konumlandırma temelli yaptıkları çalışmalarında Janssen ve arkadaşları, on farklı makine öğrenmesi algoritmasını değerlendirerek bu tez çalışmasında da kullanılan veri setini [14] kullanmışlardır. KNN ve rastgele orman regresyon yöntemleri ile 350 metre civarında bir ortalama konum tahmini hatası elde etmişlerdir [29].

Li ve arkadaşları [30]'de RSSI tabanlı konumlandırmada hiyerarşik kümeleme tabanlı bir yöntem önermiştir. Önerilen yöntem iki katmanlı hiyerarşiden oluşmaktadır. İlk katmanda %95 doğruluk elde edilmiş ve ikinci katmanda 158,41 metre medyan hata ve 346,03 metre ortalama hata elde edilmiştir. Başka bir çalışma olan [13] 'de Anagnostopoulos ve arkadaşları optimize edilmiş bir kNN yöntemini, ekstra ağaçlar yöntemini ve bir sinir ağı yöntemini, bu tez çalışmasında da kullanılan veri setini [14] kullanarak karşılaştırmıştır. Sinir ağı yaklaşımı, ortalamada 357 metre hata ve 206 metre medyan hata ile en iyi doğruluğa sahip olan yöntem olarak sunulmuştur.

3. YÖNTEM

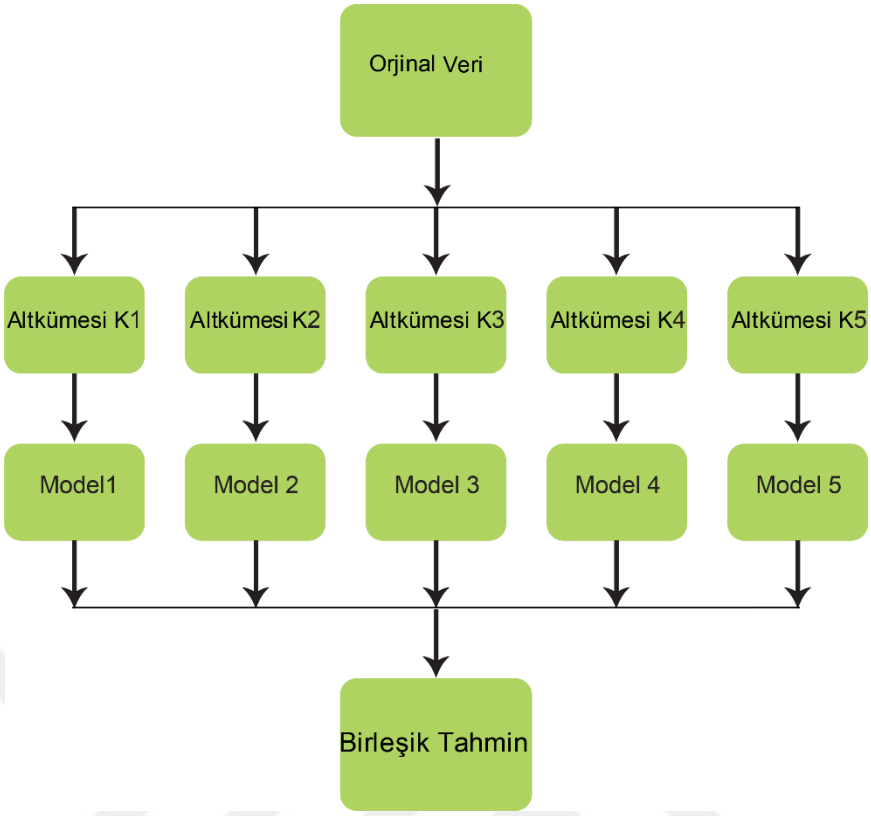
Bu bölümde tezde kullanılan algoritmalar hakkında bir genel bakış sunulmaktadır. Bölüm 3.1'de kullanılan topluluk öğrenme algoritmaları Rastgele Orman, AdaBoost ve Gradyan artırma algoritmalarına yer verilmiştir. Bölüm 3.2'de regresyon, sınıflandırma ve kümeleme için kullanılan kNN, Ağırlıklı kNN, K-ortalama kümeleme algoritmalarına genel bir bakış sunulmuştur Ardından Bölüm 3.3'te kullanılan veri seti, veri setinde ön işleme ve geliştirme ortamı açıklanmıştır ve Bölüm 3.4'te iş akışı ve önerilen yöntem, Bölüm 3.5'te performans metrikleri verilmiştir.

3.1. KULLANILAN TOPLULUK ÖĞRENME ALGORİTMALARI (ENSEMBLE LEARNING ALGORITHMS)

Topluluk, tek bir modelden daha iyi tahmin performansı üretmek için birden fazla modelin aynı öğrenme algoritması kullanılarak eğitildiği bir makine öğrenimi kavramıdır. Topluluk modelinin arkasındaki ana ilke, bir grup zayıf öğrencinin güçlü bir öğrenci oluşturmak için bir araya gelmesi ve böylece modelin doğruluğunu ve genellenebilirliğini artırmasıdır. En iyi bilinen topluluk yöntemleri, önyükleme toplaması olarak da bilinen torbalama (bagging) ve artırmadır (boosting). Torbalama yöntemi en genelde, modelin varyansını azaltmaya yardımcı olurken, artırma yöntemi modelin önyargısını azaltmaya yardımcı olmaktadır.

3.1.1. Torbalama ya da Önyükleme Toplaması, Rastgele Orman (Bagging ya da Bootstrap Aggregation, Random Forest)

Torbalama, Önyükleme prosedürünün yüksek varyanslı bir makine öğrenme algoritmasına, genellikle karar ağaçlarına (decision trees) uygulanmasıdır. Torbalama tekniğinde ilk olarak, orijinal veri setinden çoklu alt setler oluşturulur, değiştirme ile gözlemler seçilir. Bu tekniğe önyükleme adı verilir. Bundan sonra bu alt kümelerin her biri üzerinde bir temel model (zayıf model) oluşturulur. Bu modeller paralel çalışır ve birbirinden bağımsızdır. Son olarak, tüm modellerden gelen tahminlerin birleştirilmesiyle tahminler belirlenir [31]. Torbalama yönteminin çalışma prensibi Şekil 3.1'de gösterilmektedir.



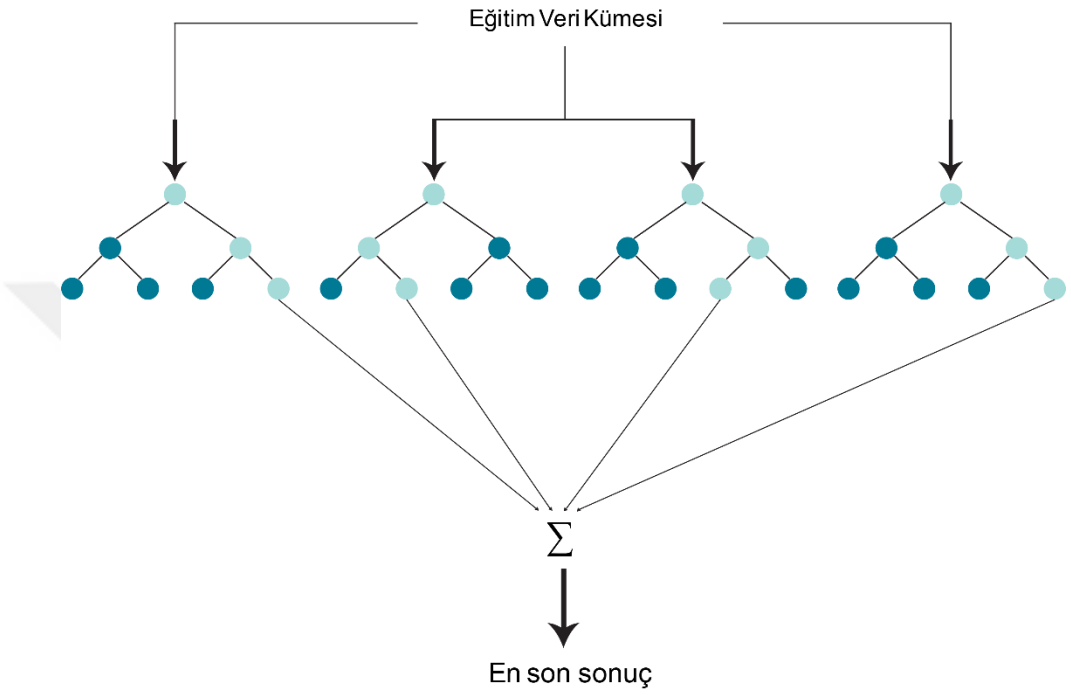
Şekil 3.1. Torbalama.

3.1.1.1. Rastgele Orman (Random Forest)

Torbalama kullanan en ünlü algoritmalardan biri Rastgele Orman algoritmasıdır. Rastgele Orman hem denetimli bir öğrenme algoritması hem de bir topluluk algoritmasıdır. İlişkisiz bir karar ağaçları ormanı oluşturmak için hem torbalama hem de rastgelelik özelliğini kullandığı için torbalama yönteminin bir uzantısıdır. Leo Breiman ve Adele Cutler tarafından geliştirilen ve yaygın olarak kullanılan bir makine öğrenme algoritmasıdır. Tek bir sonuca ulaşmak için Şekil 3.2'de gösterildiği gibi birden çok karar ağacının çıktısını birleştirir. Kullanım kolaylığı ve esnekliği hem sınıflandırma hem de regresyon problemlerini ele aldığı için yaygın şekilde benimsenmiştir. Karar ağaçları, her dahili düğümün bir öznitelik üzerinde bir "testi" temsil ettiği, akış şemalarına benzer bir yapılardır. Bu ağaçlardaki her bir dal, testin sonucunu temsil eder ve her yaprak düğümü bir sınıf etiketini temsil eder (karar tüm öznitelikler hesaplandıktan sonra alınır). Kökten yaprağa giden yollar, sınıflandırma kurallarını temsil eder [32].

Rastgele Orman algoritması, eğitimden önce ayarlanması gereken üç ana hiper parametreye sahiptir. Bunlar, düğüm boyutunu, ağaç sayısını ve örneklenen özelliklerin sayısını içerir. Rastgele Orman sınıflandırıcısı, regresyon veya sınıflandırma problemlerini çözmek için kullanılabilir.

Rastgele Orman algoritması, bir karar ağaçları koleksiyonundan oluşmaktadır ve problemin türüne bağlı olarak, tahminin kararı değişiklik gösterecektir. Regresyon amacıyla kullanıldığında bireysel karar ağaçlarının ortalaması alınmaktadır veya sınıflandırma amacıyla kullanıldığında ise en sık görülen kategorik değişken olan çoğunluk oyu tahmin edilen sınıfı belirlemektedir [33].



Şekil 3.2. Rastgele Orman

"Kalabalıkların Bilgeliği (Wisdom of the Crowds)" yaklaşımı sayesinde rastgele orman algoritması, son derece yüksek doğruluk elde edebilmektedir. Genellikle diğer doğrusal modellerden daha iyi sonuçlar ürettiği görülmektedir.

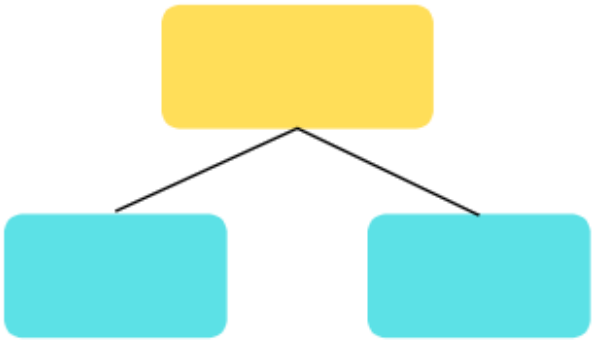
3.1.2. Artırma (Boosting), AdaBoost ve Gradyan Artırma

Artırma verilere basit modeller uygulama ve ardından verileri hatalar için analiz etme sürecidir. Ardışık ağaçlar (rastgele örnek) oluşturulur ve her adımda amaç, bir önceki ağaca kıyasla doğruluğu artırmaktır. Bir girdi bir hipotez tarafından yanlış sınıflandırıldığında, bir sonraki hipotezin onu doğru bir şekilde sınıflandırması daha olası olacak şekilde ağırlığı artar. Bu süreç, zayıf öğrenenleri daha iyi performans gösteren bir modele dönüştürür. Sonuç olarak, torbalama ve artırma gerçek dünya uygulamalarında da farklılıklar göstermektedir. Torbalama, kredi onay süreçleri ve istatistiksel genom bilimi için kullanılırken, artırma görüntü tanıma uygulamaları

ve arama motorlarında daha fazla kullanılmaktadır [34]. Artırma kullanan en yaygın algoritmalara örnek olarak Gradyan Artırma ve AdaBoost gösterilebilir.

3.1.2.1. Adaptive Boosting Algoritması veya AdaBoost Algoritması

AdaBoost algoritması Yoav Freund ve Robert Schapire tarafından 1995 yılında önerilmiştir [35]. Bu yöntem, yanlış sınıflandırılmış veri noktalarını belirleyerek ve eğitim hatasını en aza indirmek için ağırlıklarını ayarlayarak yinelemeli olarak çalışır. Model, en güçlü tahmin ediciyi bulana kadar sıralı bir şekilde optimize etmeye devam eder. AdaBoost ile kullanılan en yaygın algoritma, Şekil 3.3'te gösterildiği gibi yalnızca bir bölme anlamına gelen bir seviyeli karar ağaçlarıdır. Bu ağaçlara Karar Kütükleri (Decision Stump) de denebilmektedir [36]. Her kütük bir özellik ve bir eşik değeri (threshold - T) seçer ve ardından örnek verileri eşiğin her iki tarafında iki gruba böler.



Şekil 3.3. Karar Kütükleri Yapısı

Her bir özellik için bir karar kütüğü oluşturulduktan sonra, her ağacın Gini İndeksi hesaplanır. Gini İndeksi en düşük olan ağaç ilk kütük olarak seçilir. Bundan sonra algoritma bir model oluşturur ve tüm veri noktalarına eşit ağırlıklar verir. Örnek veri ağırlıklarını hesaplamak için Denklem **(3.1)** kullanılır:

$$Örnek\ veri\ Ağırlığı = \frac{1}{Örnek\ sayısı} \tag{3.1}$$

Bundan sonra, bu kütüğün aldığı '$Söyleme\ Miktarı$' (Amount of Say) veya '$Önem$' (Importance) hesaplamak için toplam hata hesaplanır. Toplam Hata, yanlış sınıflandırılmış örnek verilerin ağırlıklarının toplamına eşittir.

'$Söyleme\ Miktarı$'nı hesaplamak için Denklem (**3.2**) kullanılır:

$$Söyleme\ Miktarı\ (\alpha)\ =\ \frac{1}{2}log\ (\frac{1-Toplam\ Hata}{Toplam\ Hata}) \tag{3.2}$$

Yanlış tahminlere daha fazla ağırlık verilirken doğru tahminlerin ağırlıkları azaltılacaktır. Ağırlıklar güncellendikten sonra bir sonraki model oluşturulduğunda daha yüksek ağırlıklı noktalara daha fazla tercih verilecektir. Ağırlıkları güncellemek için Denklem (**3.3**) kullanılır:

$$Yeni\ Örnek\ Veri\ Ağırlığı\ =\ Eski\ Ağırlık\ *\ e^{\pm\ Söyleme\ Miktarı\ (\alpha)} \tag{3.3}$$

Daha sonra, her örnek verinin yeni örnek veri ağırlıklarına göre seçilme olasılığına sahip önyüklenmiş bir veri seti oluşturulur. Bu, yanlış sınıflandırılan örnek verilerin birden çok kez seçilme olasılığının daha yüksek olduğu ve bir sonraki kütük, yanlış sınıflandırılan örnek verinin doğru şekilde sınıflandırılmasına daha fazla odaklanacağı anlamına gelir. Yeni önyüklemeli veri seti oluşturulduktan sonra, örnek verilere tekrar eşit ağırlıklar verilir ve daha düşük bir hata alınana kadar veya n kütük sayısı oluşturulana kadar işlem tekrarlanır [36].

3.1.2.2. Gradyan Artırma Algoritması (Gradient Boosting Machine – GBM)

Leo Breiman'ın çalışmasına dayanan Jerome H. Friedman, sırayla çalışan ve bu sonraki modellerin her biri önceki modelin hatasını azaltmaya çalışan gradyan artırmayı geliştirmiştir [37]. Ancak, AdaBoost gibi örnek verilerinin ağırlıklarını değiştirmek yerine, gradyan artırma, önceki tahmin edicinin kalan hataları üzerinde eğitim gerçekleştirir. Gradyan artırma adını, gradyan iniş algoritmasını ve artırma yöntemini birleştirdiği için almıştır.

İlk olarak model kurulur ve verilen veriler üzerinde tahminler yapılır. İkinci adımda bu tahminler, daha sonra bir hedef olarak kullanılacak olan hatayı belirlemek için kullanılacaktır. Ardından, gerçek hedef ve tahminler kullanılarak hata hesaplanır. Üçüncü adımda bu hatalar üzerine ikinci bir model kurulur ve hatalar yeni hedef olarak yapılarak ve orijinal özellikleri kullanarak tahminler yapılır. Bu durumda elde edilen tahminler aslında hata değerleri olacaktır. Dördüncü adımda, birinci modelden gelen tahminlere ikinci modelden gelen tahminler eklenerek birinci modelin tahminleri güncellenir. Son olarak, yeni hataların hesaplandığı ve bu yeni hataların hedef olarak belirlendiği 2'den 4'e kadar olan adımlar tekrarlanır. Bu işlemler, hata sıfır olana kadar veya oluşturulacak model sayısını belirten durdurma kriterine ulaşılıncaya kadar tekrarlanır [38].

3.2. KULLANILAN REGRESYON, SINIFLANDIRMA VE KÜMELEME ALGORİTMALARI

3.2.1. K- En Yakın Komşu (K-Nearest Neighbors – kNN) Algoritması

K-En Yakın Komşu algoritması (K-Nearest Neighbors – kNN), tüm makine öğrenmesi algoritmalarının en temel algoritmalarından biridir. Denetimli öğrenme türünde bir algoritmadır ve ilk olarak 1951'de Evelyn Fix ve Joseph Hodges tarafından geliştirilmiş ve daha sonra Thomas Cover tarafından genişletilmiştir. Hem sınıflandırma hem de regresyon problemlerini çözmek için kullanılabilir. Her iki durumda da girdi, bir veri setindeki eğitim örneklerinden oluşur, ancak çıktı, kNN'nin sınıflandırma veya regresyon için kullanılmasına bağlıdır [39]. kNN, veri noktalarını yakınlıklarına ve diğer mevcut verilerle olan ilişkilerine göre tahmin eder. Bu algoritma, benzer veri noktalarının birbirine yakın bulunabileceğini varsayar. Sonuç olarak, genellikle Öklid uzaklığı aracılığıyla, bir grafikteki veri noktaları arasındaki mesafeyi hesaplamaya çalışır ve daha sonra, kNN sınıflandırmasında, komşularının etiketlerinin çoğunluğuna göre tahminini belirler. $k = 1$ ise, tahmin, tek en yakın komşunun sınıfına atanır. kNN regresyonunda tahmin, en yakın k komşunun değerlerinin ortalamasıdır [40].

Algoritma iki aşamada çalışır: çevrimdışı aşama ve çevrimiçi aşama. İlk olarak veri seti iki alt kümeye ayrılır: bir eğitim kümesi ve bir deneme kümesi. Alt kümeler arasında tarafsız bir mekansal yayılma sağlamak için ana veri seti rastgele bölünmelidir. Deneme kümesinden daha büyük olması gereken eğitim kümesi referans noktası görevi görür. Daha sonra bu iki alt küme algoritma için aşağıdaki gibi kullanılabilir.

Çevrimdışı aşamada eğitim aşaması tamamlanır ve çevrimiçi aşamada tahmin elde edilir; bu tezde konum tahmini elde edilecektir. Çevrimiçi aşamada, Denklem (**3.4**)'de gösterildiği gibi, deneme kümesindeki her RSSI değeri ile eğitim kümesindeki her bir RSSI değeri arasındaki d mesafesi hesaplanarak eğitim ve deneme kümeleri arasındaki mesafe matrisi hesaplanmalıdır. Bu denklemde, deneme kümesi ve eğitim kümesi için RSSI ölçümleri sırasıyla $RSSI_{deneme}$ ve $RSSI_{eğitim}$ ile temsil edilir.

$$d = \left(\sum |RSSI_{\ deneme} - RSSI_{eğitim}|^{q}\right)^{\frac{1}{q}} \tag{3.4}$$

Denklem (**3.4**)'te Manhattan mesafesi kullanılıyorsa, $q = 1$; Öklid uzaklığı kullanılıyorsa, $q = 2$ değerleri kullanılır.

Uygun parametre k'nın seçilmesinden sonra algoritma, Denklem (**3.5**)'de gösterildiği gibi k örneğinin ortalama koordinatlarını bularak bir cihazın konumunu tahmin eder [41].

$$\bar{x} = \frac{\sum_{i=1}^{k} x_i}{k}, \bar{y} = \frac{\sum_{i=1}^{k} y_i}{k} \tag{3.5}$$

Konum tahmini hatası, tahminleri deneme örneği gerçek GPS koordinatlarıyla karşılaştırılarak ölçülebilir. Hem sınıflandırma hem de regresyon için, komşuların katkılarına ağırlık atamak için uygun bir teknik kullanarak daha yakın komşuların, uzak olanlara göre ortalamaya daha fazla katkıda bulunması sağlanabilir. Örneğin, yaygın olarak kullanılan ve d değerinin komşuya olan mesafeyi gösterdiği bir ağırlıklandırma şemasında her komşuya $1/d$'lik bir ağırlık verilmektedir.

3.2.2. K-Ortalamalar Kümeleme (K- Means Clustering) Algoritması

K-Ortalamalar kümeleme algoritması, en basit ve en popüler denetimsiz makine öğrenmesi algoritmalarından biridir. Veri setlerini alt kümelere bölmek için açık bir mesafe ölçüsü kullanan bir kümeleme algoritmasıdır. Genellikle, denetimsiz algoritmalar, sınıflandırılmamış veya etiketlenmemiş veri noktalarını içeren veri setlerindeki desenleri (patterns) tanımlamaktadır. K-Ortalamalar algoritmasının amacı, benzer veri noktalarını birlikte gruplandırmak ve altta yatan desenleri keşfetmektir. Bu amaca ulaşmak için K-ortalamalar, bir veri setinde sabit sayıda (k) küme arar. Bir küme, belirli benzerlikler nedeniyle bir araya getirilen bir veri noktaları koleksiyonunu ifade eder.

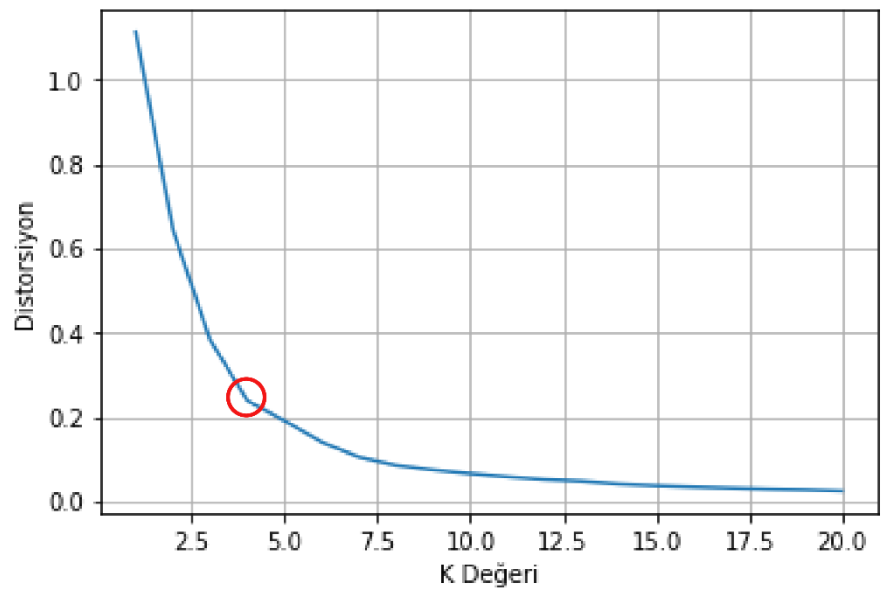
İlk olarak, veri setinde ihtiyaç duyulan kümelerin merkezlerinin sayısını ifade eden bir hedef sayı k tanımlanır. Bir kümenin merkezi (centroid), kümenin merkezini temsil eden hayali veya gerçek bir konumdur [42]. Her bir veri noktası, küme içinde kareler toplamının azaltılması yoluyla kümelerden birine atanır. Küme İçinde Kareler Toplamı (Within Cluster Sum of Squares- WCSS), bir küme içindeki tüm noktaların kümenin merkezine olan ortalama uzaklığının karesinin toplamıdır. İstenilen durum bu değerin mümkün olduğunca düşük olmasıdır.

K- Ortalamalar kümelemesi için en yaygın olarak kullanılan mesafe hesaplaması, iki veri noktası arasındaki mesafeyi hesaplamak için kullanılan Öklid Mesafesi formülüdür. K-Ortalamalarındaki '*means'*, verilerin ortalamasını ifade eder. Burada temel amaç kümenin

merkezini bulmaktır. K-Ortalamalar algoritması ilk olarak, her küme için başlangıç noktası olarak kullanılan, rastgele seçilmiş küme merkezlerinden oluşan bir ilk grup ile başlar, daha sonra girdilerin her biri ile *k* küme merkezleri arasındaki mesafeyi hesaplar ve en kısa mesafeli kümeye yeniden atar. Yeniden atamadan sonra, kümedeki veri noktalarının ortalaması hesaplanarak her kümenin merkezi güncellenir. Algoritma, küme merkezlerinin konumlarını optimize etmek için yinelemeli (tekrarlayan) hesaplamalar gerçekleştirir [43].

Algoritma, kümelerin oluşturulmasını ve optimize edilmesini, ya kümeleme başarılı olduğu için değerlerinde herhangi bir değişiklik olmayacak şekilde küme merkezleri sabitlendiğinde ya da tanımlanan yineleme sayısına ulaşıldığında durdurur. K-Ortalamalar algoritmasında optimal *k* küme sayısını belirlemek için en çok kullanılan yöntem dirsek yöntemidir.

Şekil 3.4'de gösterilen dirsek yöntemi, farklı *k* değerleri tarafından üretilen maliyet fonksiyonunun değerini göstermektedir. *k* arttıkça, ortalama bozulma azalacaktır, her küme daha az veri noktasına sahip olacak ve veri noktaları ilgili merkezlerine daha yakın olacaktır. Ancak, *k* sayısı çok arttıkça ortalama bozulmadaki iyileştirmeler azalacaktır. Bozulmadaki iyileşmenin yavaş yavaş azalmaya başlayacağı *k* değerine dirsek denir ve bu noktada verilerin daha fazla kümelere bölünmesinin durdurulması gerekmektedir [44].

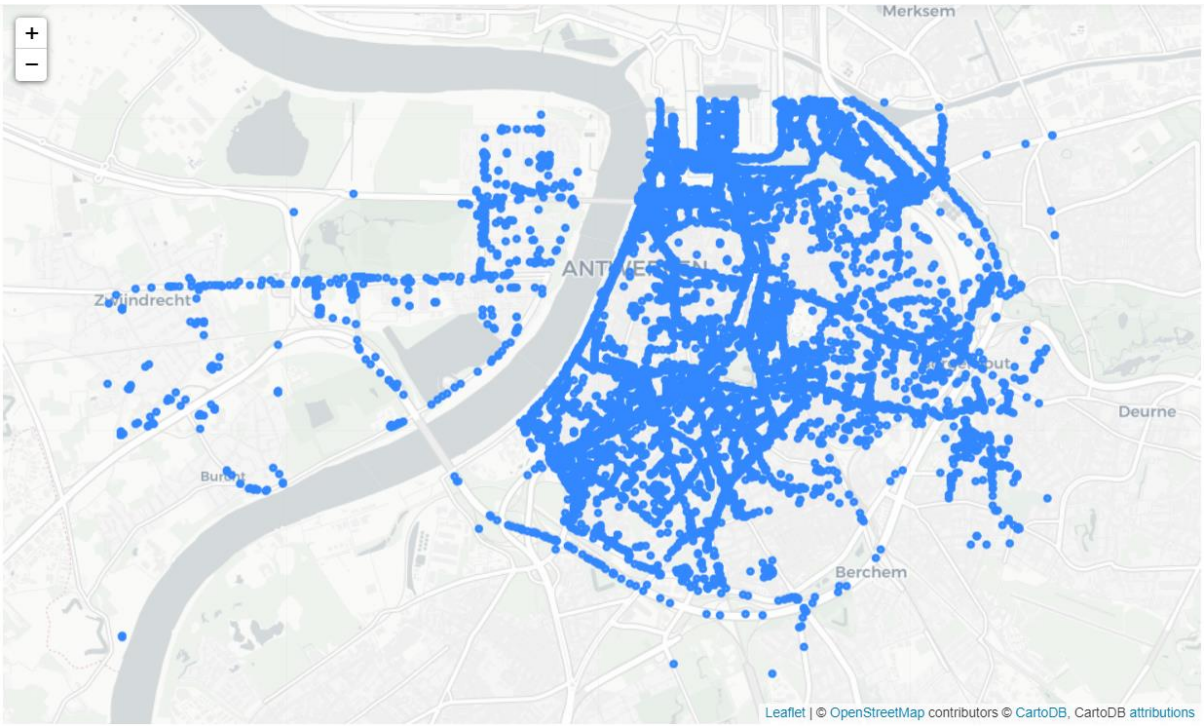


Şekil 3.4. Dirsek Yöntemi

3.3. KULLANILAN VERİ SETİ VE GELİŞTİRME ORTAMI

3.3.1. Kullanılan Veri Seti

Bu tezde kullanılan veri seti, halka açık bir LoRaWAN veri setidir. Aernouts ve diğerleri tarafından yayınlanmıştır [14]. Belçika'nın Antwerp şehrinde bir kentsel alanda parmak izleri toplanarak elde edilen bir veri setidir. 72 ağ geçidi tarafından alınan iletilen sinyalin RSSI değerleri, bir GPS cihazı tarafından tahmin edilen sinyalin aktarım konumunun GPS koordinatları, ayrıca GPS tahminlerinin Yatay Hassasiyet Seyreltmesi (Horizontal Dilution of Precision - HDOP), SF değerleri ve RX Zaman değerleri. Veri seti 130429 kayıt içermektedir. Veri setinin mekansal dağılımı Şekil 3.5'de gösterilmektedir.



Şekil 3.5. Veri setinin veri noktalarının mekansal dağılımı

3.3.2. Veri Setinin Ön İşlemesi

Veri setinin ön işlemesinde verilerin toplandığı alanda 72 ağ geçidi olmasına rağmen sadece 44 ağ geçidinin mesaj aldığı görülmüştür. Bu nedenle herhangi bir mesaj almayan 28 ağ geçidi veri setinden çıkarılmıştır. Ek olarak, 615 adet kaydın 2'den fazla HDOP değerine sahip olduğu belirlenmiştir. 2'den yüksek bir HDOP değeri, zayıf bir GPS sinyal kalitesini göstermektedir. Bu kayıtlar tahmin hata oranını arttırdığı için veri setinden çıkarılmıştır [45].

SF, HDOP ve RX Time değerlerinin de veri setinden çıkarılmasından sonra geriye 129814 kayıt kalmıştır. Son olarak daha fazla ön işleme için, üçten daha az ağ geçidi tarafından alınan tüm kayıtlar kaldırılmıştır ve kalan kayıt sayısı nihai olarak 54614 olmuştur.

Bundan sonra, mesajların %0,1'inden daha azını alan tüm ağ geçitleri veri setinden çıkarıldığında veri setinde 28 ağ geçidi kalmaktadır. RSSI değerleri arasındaki tüm sıfır ve boş değerler, sütunun ortalama değerine empoze edilmiştir.

Veri setinin 30 sütun ve 54614 kayıttan oluşan son haline ait bir örneği Tablo 3.1'de gösterilmiştir.

Tablo 3.1. İşlenmiş Veri Seti Örneği

BI1	**BI2**	**....**	**BI27**	**BI28**	**Enlem**	**Boylam**
-200	-107	...	-108	-200	51.19404	4.418484
-200	-111	...	-200	-200	51.2138	4.445716
...	...	...	...	...	...	...

-200 dB, mesajı almayan ağ geçitleri için veri seti oluşturucuları tarafından verilen değer olan menzil dışı bir RSSI değeridir. Çalışmamızda veri seti rastgele bir eğitim kümesine ve bir test kümesine (sırasıyla %70 ve %30) bölünmüştür.

3.3.3. Geliştirme Ortamı

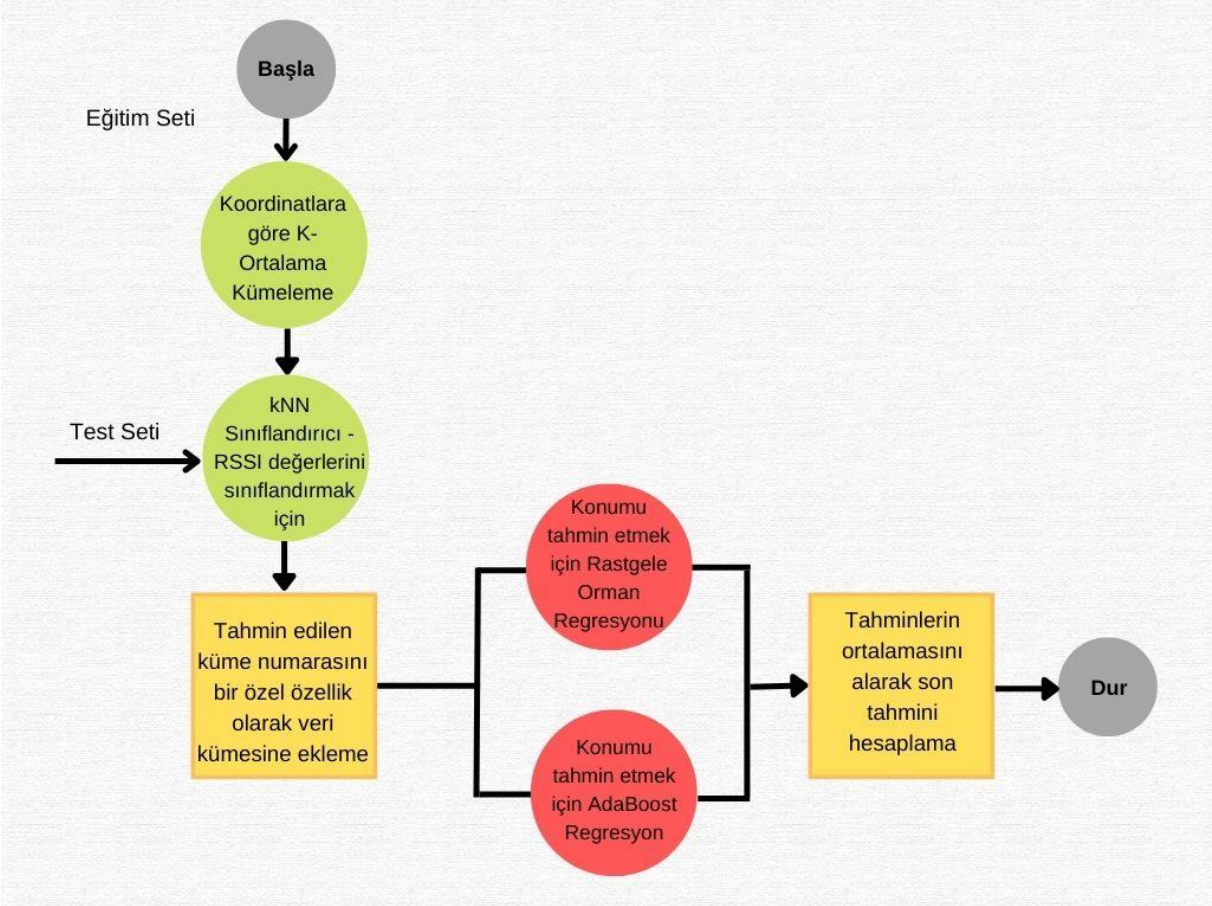
Bu çalışmada yapılan testler Spyder IDE 5.1.5 geliştirme ortamında Python 3.10.2 dili ile kodlanmıştır.

Kullanılan kütüphaneler şu şekildedir:

- *scikit-learn* sürüm 1.0.2.sınıflandırma, regresyon ve kümeleme algoritmalarının uygulanması, veri ön işlemesi ve R^2 belirleme katsayısının hesaplanması için kullanılmıştır.
- Rapor edilen deneylerde mesafeleri ölçmek için Geopy 2.2.0,
- Mesafelerin ortalama ve medyan hatasını hesaplamak için *Statistics* kütüphanesi,
- Veri görselleştirme için *Matplotlib*, *Seaborn* ve *Folium* kütüphaneleri kullanılmıştır.

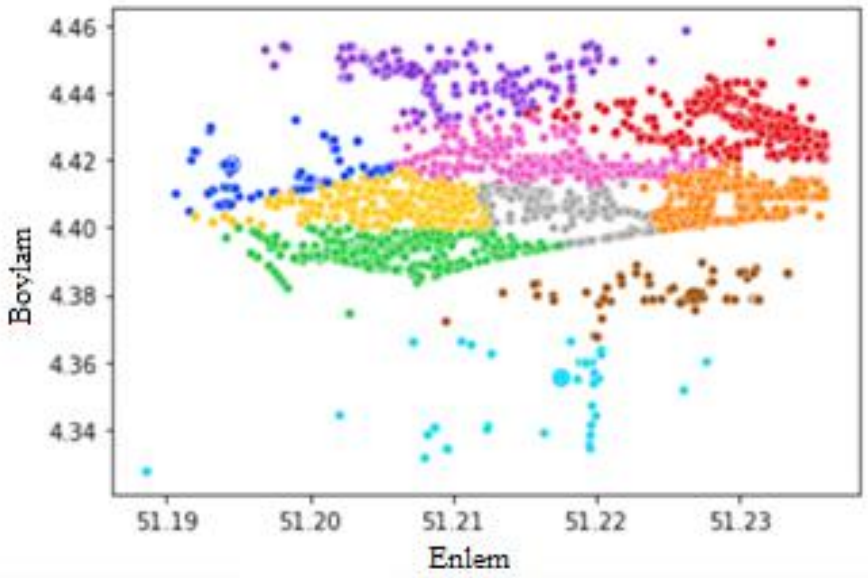
3.4. ÖNERİLEN METODOLOJİ

Bu tez çalışmasının birinci kısmında, RSSI değerlerine dayalı konum tahmini için beş algoritma arasında bir karşılaştırma yapılmıştır. Bu algoritmalar arasında Ortalama Hata, Medyan hata ve Belirleme Katsayısı metrikleri arasında karşılaştırma yapılarak sonuçlar elde edilmiştir. Bu kısımda Rastgele Orman, AdaBoost, Gradyan Artırma, kNN regresyonu ve Ağırlıklı kNN regresyonu algoritmaları yer almıştır. Daha sonra çalışmanın ikinci kısmında, farklı hibrit konum tahmin yöntemleri değerlendirilmiş ve karşılaştırılmıştır. Bu kısımda karşılaştırılan metrikler, Ortalama Hata ve Medyan Hata olarak belirlenmiştir. Hibrit yöntemin sonuçlarına bağlı olarak Rastgele Orman ve AdaBoost algoritmalarını hibrit olarak kullanan bir metodoloji önerilmiştir. Üçüncü kısımda önerilen bu metodolojinin akış şeması Şekil 3.6'da gösterilmektedir.



Şekil 3.6. Önerilen Metodolojinin Akış Şeması

Veri setine ilk olarak K-Ortalama Kümeleme algoritması uygulanmıştır. Bunun sonucunda kümeleme, alanı Şekil 3.7'de gösterildiği gibi daha küçük alanlara bölen koordinatlar kullanılarak gerçekleştirilmiştir.



Şekil 3.7. K- Ortalama Kümeleme Algoritmasının sonucunda oluşan kümeler

Daha sonra bu aşamada elde edilen küme numarası değeri veri setine özel bir nitelik olarak eklenmiştir. Bu değer daha sonra kNN sınıflandırıcısı tarafından test seti kullanılırken her bir veri örneğinin RSSI değerlerini ait oldukları kümeye sınıflandırmak için kullanılmıştır. Son olarak, tahmin edilen küme numarası niteliği, RSSI değerleri ile, konum tahminlerinin ortalama tahminini bulmak için önceden seçilen Rastgele Orman ve AdaBoost hibrit yönteminde kullanılmıştır. Önerilen yöntem, biri hibrit yöntem, biri yalnızca AdaBoost ve diğeri yalnızca Rastgele Orman kullanılarak olmak üzere üç yaklaşımda gerçekleştirilmiş ve sonuçları karşılaştırılmıştır.

3.5. PERFORMANS METRİKLERİ

Tüm algoritmalar için ölçülen performans metrikleri, ortalama hata (mean error), medyan hata (median error) ve belirleme katsayısıdır (coefficient of determination). Ortalama hata, metre cinsinden tanımlanır ve tahmin edilen konumlar ile gerçek konumlar arasındaki tüm mesafeler hesaplanarak hataların ortalamasını ifade etmektedir. Medyan hata, metre cinsinden tanımlanır ve tahmin edilen konumlar ile gerçek konumlar arasındaki sıralanmış hata mesafeleri listesinin "ortası"nı göstermektedir.

R-kare (veya R^2) olarak da bilinen belirleme katsayısı, regresyon puanı işlevidir ve belirli bir olayın sonucunu tahmin ederken bir değişkendeki farklılıkların ikinci bir değişkendeki farklılıkla nasıl açıklanabileceğini incelemektedir. Buna "uyum derecesi" denir. 0.0 ile 1.0 arasında bir değer ile temsil edilir. 1.0 değeri mükemmel bir uyumu gösterir ve bu nedenle gelecekteki tahminler için oldukça güvenilir bir model olduğunu, 0.0 değeri ise hesaplamanın verileri hiç doğru bir şekilde modellemediğini gösterir. Örneğin 0.20 değeri, bağımlı değişkenin %20'sinin bağımsız değişken tarafından tahmin edildiğini gösterirken, 0.50 değeri, bağımlı değişkenin %50'sinin bağımsız değişken tarafından tahmin edildiğini gösterir ve değerler bu şekilde devam eder [46].

4. BULGULAR

Uygulanan tüm senaryolardaki konum tahminlerinin sonuçları üç bölümde sunulmuştur. Bölüm 4.1'de beş algoritmanın sonuçları ve karşılaştırma değerlendirilmesi, Bölüm 4.2'de hibrit yöntemlerin sonuçları ve Bölüm 4.3'te önerilen metodolojinin sonuçları verilmiştir.

4.1. KULLANILAN ALGORİTMALAR İLE ELDEN EDİLEN SONUÇLAR

Rastgele Orman, AdaBoost regresyonu, Gradyan Artırma regresyonu, kNN regresyonu, ağırlıklı kNN regresyonu yönemlerinin karşılaştırılmalı olarak incelendiği bu bölümde, her algoritmada farklı parametreler için ortalama hata, medyan hata ve belirleme katsayısı hesaplanmıştır. Veri seti, tüm algoritmalarda sırasıyla %70 ve %30 örnekleri içeren bir eğitim kümesi ve bir test kümesi olarak rastgele bölünmüştür.

4.1.1. kNN Algoritması ile Elde Edilen Sonuçlar

Algoritma, farklı $1-16$ arası k komşuluk değerleri ile gerçeklenmiştir. kNN algoritmasında kullanılan değişkenler ve değerleri Tablo 4.1'de gösterilmektedir. . Elde edilen ortalama hata, medyan hata ve R^2 değerleri Tablo 4.2'de verilmiştir.

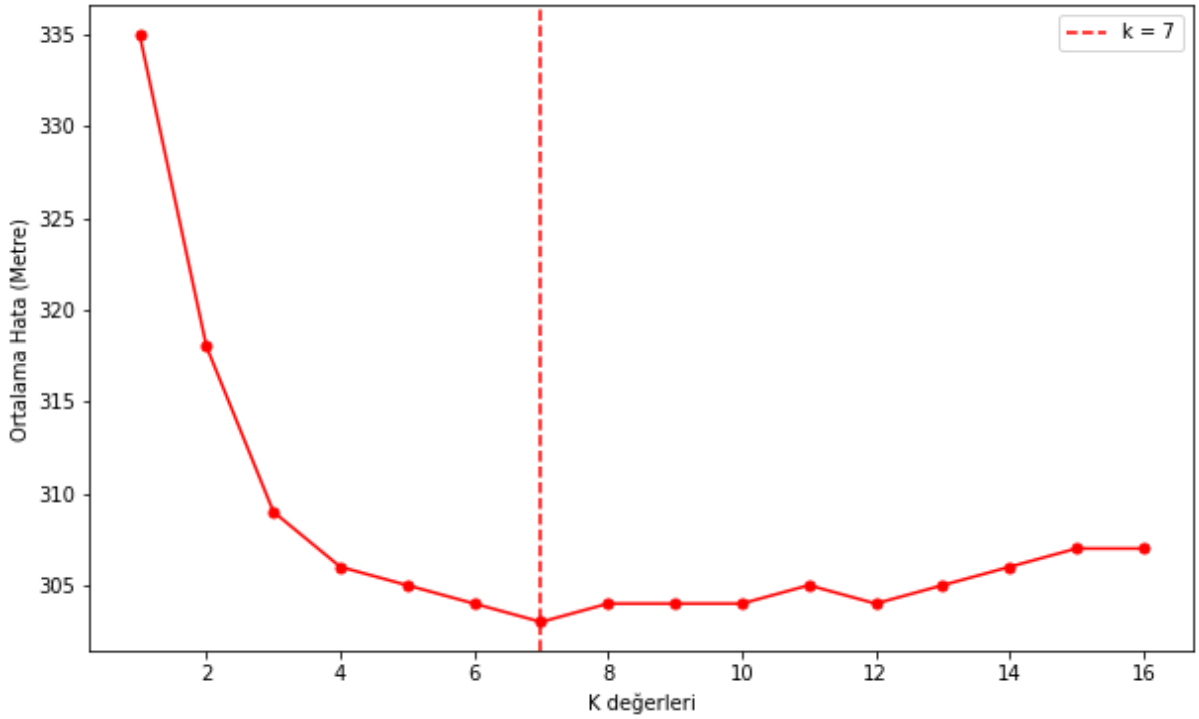
Tablo 4.1. kNN algoritmasında kullanılan değişkenler ve değerleri

Parametre Adı	Parametre Açıklaması	Parametre Değeri
weights	Ağırlık	Tek tip (uniform)
algorithm	Algoritma	brute
p	Minkowski metriği için güç	Öklit uzaklığı = 2
n_neighbors	Komşu- N	[1 – 16]

Tablo 4.2. kNN algoritması ile elde edilen ortalama hata, medyan hata ve $\boldsymbol{R^2}$değerleri

K Değeri	Ortalama Hata (metre)	Medyan Hata (metre)	R^2
1	335	**134**	0.87
2	318	178	0.90
3	309	184	0.91
4	306	190	0.91
5	305	191	0.91
6	304	192	0.91
7	**303**	195	0.91
8	304	194	0.91
9	304	195	0.91
10	304	195	0.91
11	305	195	0.91
12	304	196	0.92
13	305	198	0.92
14	306	200	0.91
15	307	198	0.91
16	307	200	0.91

$k = 7$ için en düşük ortalama hata değeri 303 metre, $k = 1$ için en düşük medyan hata değeri 134 metre $ve\ k = 12$ ya da $k = 13$ olduğunda en yüksek R^2 elde edilmiştir. k değeri veya komşu sayısı düşük olduğunda, model yalnızca veri örneğine en yakın değerleri seçmekte, ne kadar çok komşu dahil edilirse sonuç o kadar doğruluğa yaklaşmakta, ancak komşu sayısı çok fazla artarsa performans düşmeye başlamaktadır. Ortalama hata ile k değerleri arasındaki ilişkiyi gösteren grafik Şekil 4.1'de gösterilmektedir.



Şekil 4.1. Ortalama hata ile ***k*** değerleri arasındaki ilişki

4.1.2. Ağırlıklı KNN Algoritması ile Elde Edilen Sonuçlar

Algoritma $1-20$ arası k komşuluk değerleri ile gerçeklenmiştir. WKNN algoritmasında kullanılan değişkenler ve değerleri Tablo 4.3'te gösterilmektedir. Elde edilen ortalama hata, medyan hata ve R^2 değerleri Tablo 4.4'te verilmiştir.

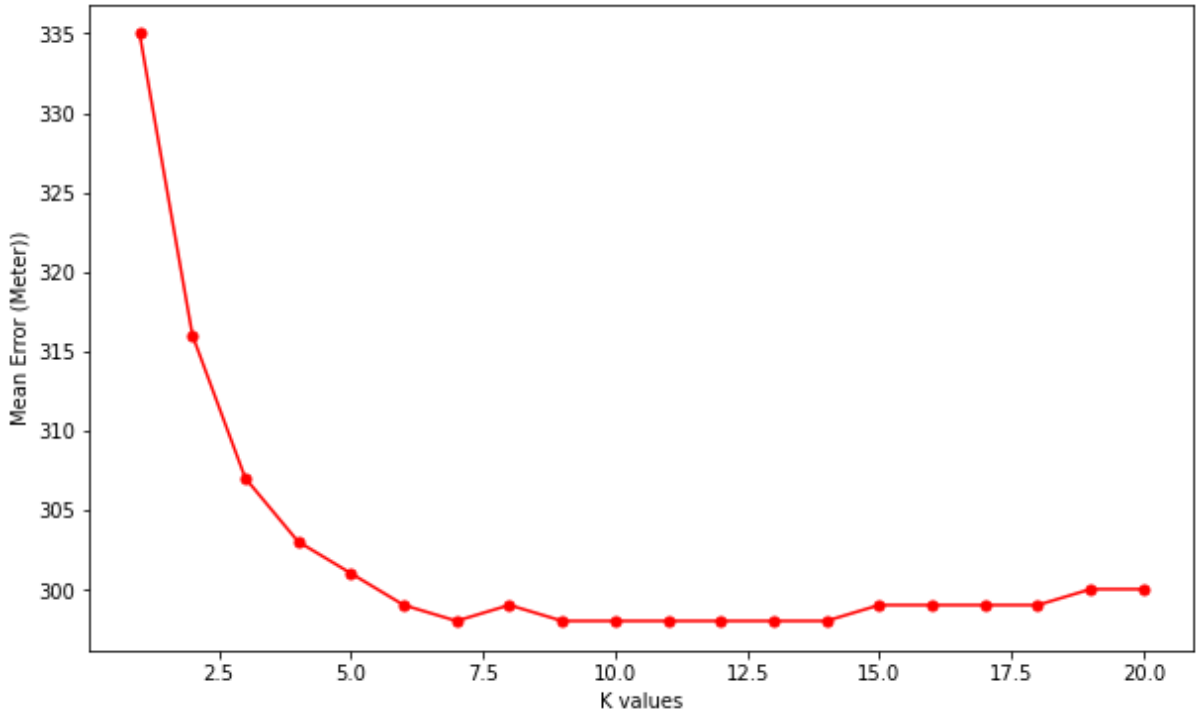
Tablo 4.3. Ağırlıklı kNN algoritmasında kullanılan değişkenler ve değerleri

Parametre Adı	Parametre Açıklaması	Parametre Değeri
weights	Ağırlık	distance
algorithm	Algoritma	brute
p	Minkowski metriği için güç	Öklit uzaklığı=2
n_neighbors	Komşu- N	[1 – 20]

Tablo 4.4. WKNN algoritması ile elde edilen ortalama hata, medyan hata ve R^2 değerleri

K Değeri	Ortalama Hata (metre)	Medyan Hata (metre)	R^2
1	335	134	0.87
2	316	169	0.90
3	307	172	0.91
4	303	179	0.91
5	301	179	0.91
6	299	178	0.91
7	**298**	180	0.91
8	299	180	0.91
9	**298**	180	0.91
10	298	182	0.92
11	298	181	0.92
12	298	182	0.92
13	298	183	0.92
14	298	184	0.92
15	299	185	0.92
16	299	185	0.92
17	299	186	0.92
18	299	187	0.92
19	300	187	0.92
20	300	188	0.92

$k = 7$ veya $k = 9$ değerleri için en düşük ortalama hata değeri 298 metre ve $k = 1$ değeri için olduğunda en düşük medyan değeri 134 metre ve k değeri 10 ile 20 arasında olduğunda en yüksek R^2 elde edilmiştir. Daha önce belirtildiği gibi, k değeri veya komşu sayısı düşük olduğunda, model sadece veri örneğine en yakın değerleri seçmektedir, ne kadar çok komşu dahil ederse, sonuç o kadar doğruluğa doğru yaklaşmaktadır. Ortalama hata ile k değerleri arasındaki ilişkiyi gösteren grafik Şekil 4.2'de gösterilmektedir.



Şekil 4.2. Ortalama hata ile k değerleri arasındaki ilişki

4.1.3. Rastgele Orman Algoritması ile Elde Edilen Sonuçlar

Algoritma farklı ağaç sayıları ile gerçeklenmiştir. Rastgele Orman algoritmasında kullanılan değişkenler ve değerleri Tablo 4.5'te gösterilmektedir. Elde edilen ortalama hata, medyan hata ve R^2 değerleri Tablo 4.6'da verilmiştir.

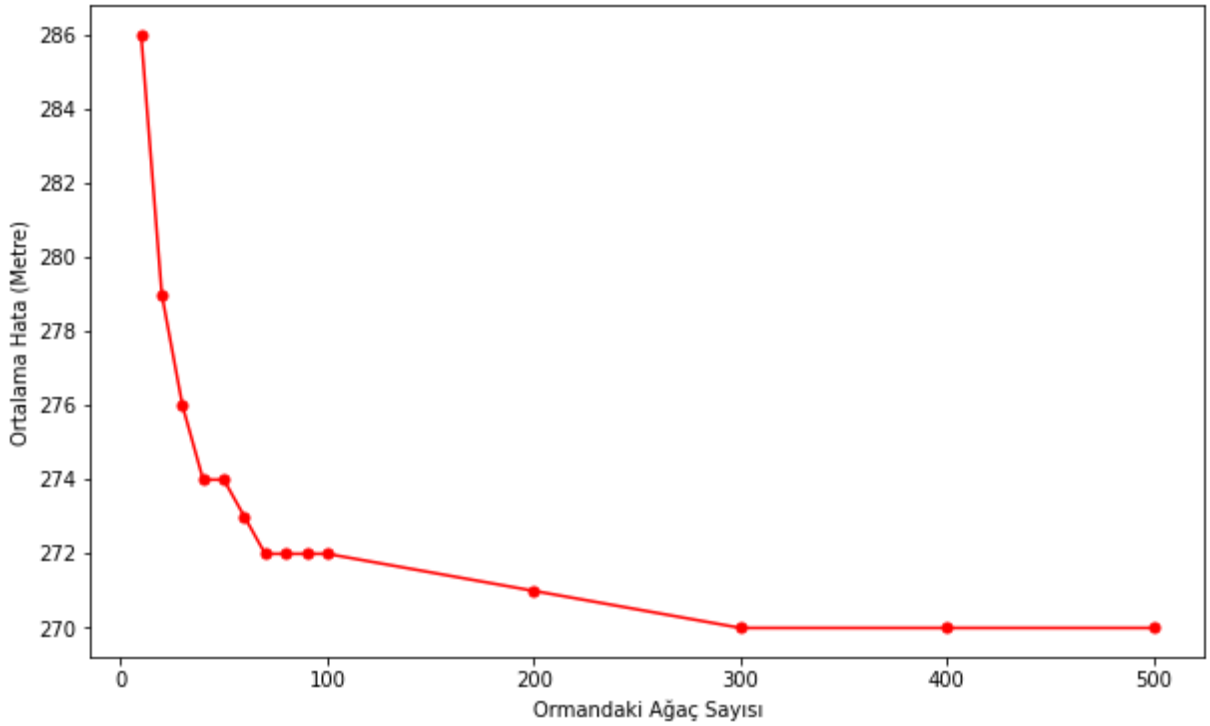
Tablo 4.5. Rastgele Orman algoritmasında kullanılan değişken ve değeri

Parametre Adı	Parametre Açıklaması	Parametre Değeri
n_estimators	Ormandaki Ağaç Sayısı	10,20,30,40,50,60,70,80,90,100, 200,300,400,500

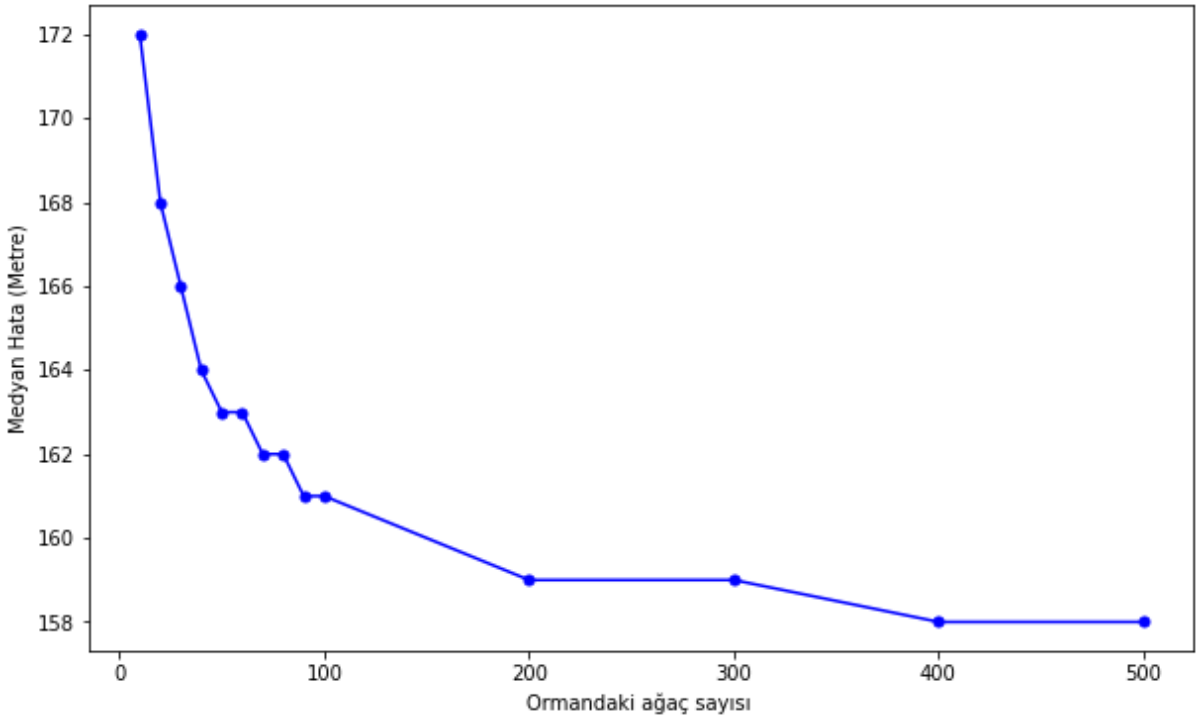
Tablo 4.6. Rastgele Orman algoritması ile elde edilen ortalama hata, medyan hata ve R^2 değerleri

Ormandaki Ağaç Sayısı	Ortalama Hata (metre)	Medyan Hata (metre)	R^2
10	286	172	0.92
20	279	168	0.92
30	276	166	0.93
40	274	164	0.93
50	274	163	0.93
60	273	163	0.93
70	272	162	0.93
80	272	162	0.93
90	272	161	0.93
100	272	161	0.93
200	271	159	0.93
300	**270**	159	0.93
400	270	**158**	0.93
500	270	158	0.93

Ormandaki ağaç sayısı 300 ile 500 arasında olduğunda en düşük ortalama hata değeri 270 metre, ağaç sayısı 400 ile 500 arasında olduğunda en düşük medyan hata değeri 158 metre, ağaç sayısı 30'dan fazla olduğunda en yüksek R^2 elde edildi. Şekil 4.3, ortalama hata ile ağaç sayısı arasındaki ilişkiyi, Şekil 4.4 ise medyan hata ile ağaç sayısı arasındaki ilişkiyi göstermektedir. Daha fazla ağaç daha iyi doğrulukla sonuçlanmaktadır, ancak bu aynı zamanda daha fazla hesaplama maliyeti anlamına gelmektedir ve belirli ağaç sayısından sonra bu iyileştirme ihmal edilebilmektedir. Şekil 4.3'de gördüğümüz gibi, bir süre sonra ağaç sayısını artırsak bile hata oranında önemli bir iyileşme ortaya çıkmamaktadır.



Şekil 4.3. Ortalama hata ile ormandaki ağaç sayısı arasındaki ilişki



Şekil 4.4. Medyan hata ile ormandaki ağaç sayısı arasındaki ilişki

4.1.4. Gradyan Artırma Algoritması ile Elde Edilen Sonuçlar

Algoritma, farklı artırma aşamalarının sayısı, bireysel regresyon tahmincilerinin farklı maksimum derinlik değerleri ve iki farklı öğrenme oranı kullanılarak gerçeklenmiştir. Gradyan Artırma algoritmasında kullanılan değişkenler ve değerleri Tablo 4.7'de gösterilmektedir.
Elde edilen ortalama hata, medyan hata ve R^2 değerleri Tablo 4.8'de verilmiştir.

Tablo 4.7. Gradyan artırma algoritmasında kullanılan değişkenler ve değerleri

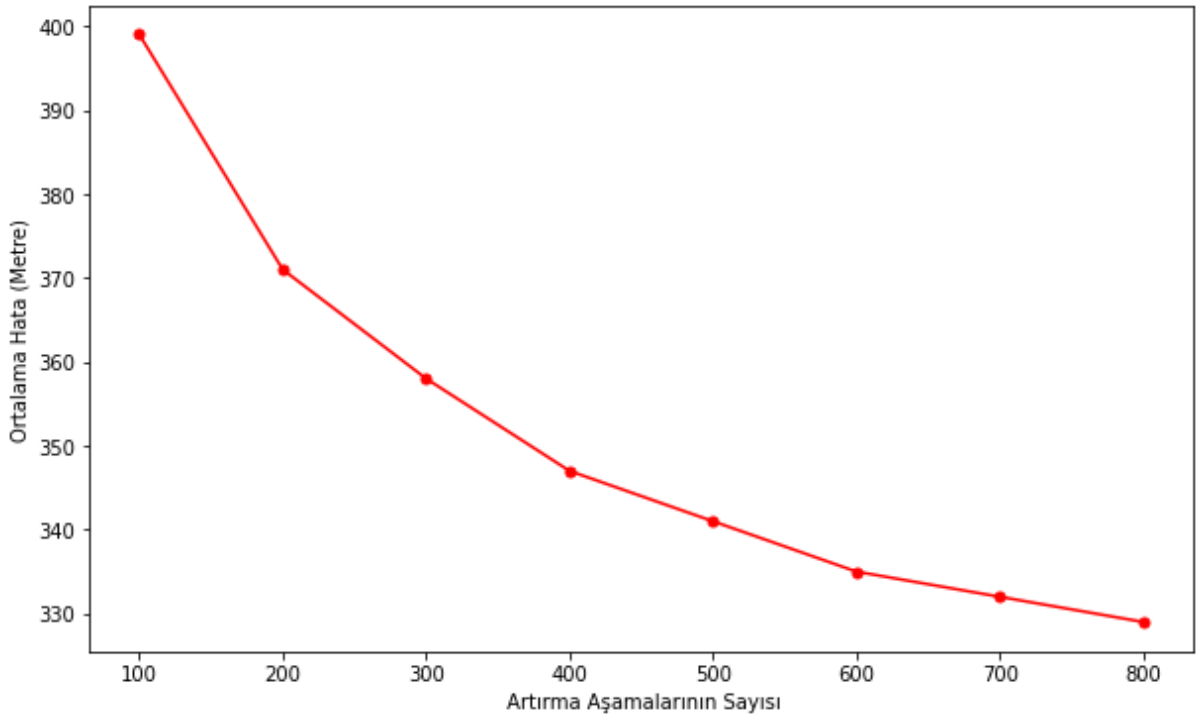
Parametre Adı	Parametre Açıklaması	Parametre Değeri
loss	Optimize edilecek kayıp fonksiyonu.	huber
n_estimators	Gerçekleştirilecek artırma aşamalarının sayısı.	100,200,300,400,500,600, 700,800,900,1000
learning_rate	Modelin ne kadar hızlı öğrendiği anlamına gelir. Eklenen her ağaç, genel modeli değiştirir. Değişikliğin büyüklüğü öğrenme oranı ile kontrol edilir.	0.1, 0.15
max_depth	Bireysel regresyon tahmincilerinin maksimum derinliği.	3,6,7,8,9

Tablo 4.8. Gradyan Artırma algoritması ile elde edilen ortalama hata, medyan hata ve R^2 değerleri

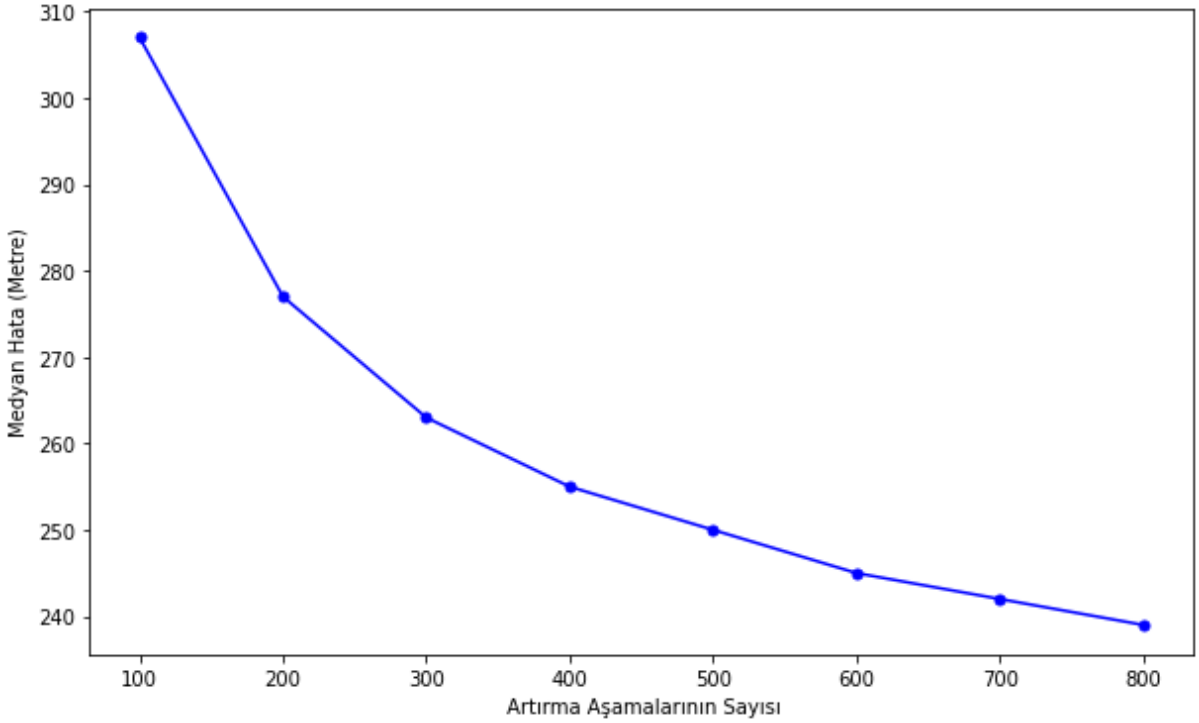
Öğrenme Oranı 0.1					Öğrenme Oranı 0.15				
Artırma Aşamalarının Sayısı	**Max. derinlik**	**Ortalama Hata (metre)**	**Medyan Hata (metre)**	R^2	**Artırma Aşamalarının Sayısı**	**Max. derinlik**	**Ortalama Hata (metre)**	**Medyan Hata (metre)**	R^2
100	3	399	307	0.88	700	3	323	234	0.91
200	3	371	277	0.89	700	6	286	192	0.93
300	3	358	263	0.90	800	6	285	192	0.93
400	3	347	255	0.90	800	7	285	188	0.93
500	3	341	250	0.91	800	8	**283**	186	0.93

600	3	335	245	0.91	800	9	285	**185**	0.93
700	3	332	242	0.91	900	6	286	191	0.93
800	3	329	239	0.91	1000	6	286	193	0.93

Artırma aşamalarının sayısı 800 ve maksimum derinlik 8 olduğunda en düşük ortalama hata değeri 283 metre, artırma aşamalarının sayısı 800 ve maksimum derinlik 9 olduğunda en düşük medyan hata değeri 185 metre, artırma aşamalarının sayısının 700 ile 1000 arasında ve maksimum derinliğin 6 ya da 9 olduğunda en yüksek R^2 elde edilmiştir. Öğrenme oranı 0.1 iken, ortalama hata ile artırma aşamalarının sayısı arasındaki ilişkiyi gösteren grafik Şekil 4.5'de gösterilmektedir. Öğrenme oranı 0.1 iken, medyan hata ile artırma aşamalarının sayısı arasındaki ilişkiyi gösteren grafik ise Şekil 4.6'da gösterilmektedir.



Şekil 4.5. Öğrenme oranı 0.1 iken, Ortalama hata ile artırma aşamalarının sayısı arasındaki ilişki



Şekil 4.6. Öğrenme oranı 0.1 iken, Medyan hata ile artırma aşamalarının sayısı arasındaki ilişki

4.1.5. Adaptive Boost Algoritması ile Elde Edilen Sonuçlar

Algoritma, bireysel regresyon tahmin edicilerinin farklı maksimum derinlik değerleri ile ve iki farklı öğrenme oranı kullanılarak gerçeklenmiştir. Kayıp fonksiyonu ve *n_estimators* gibi parametrelerin nihai sonuç üzerinde büyük bir etkisi olmadığı için her parametrede yalnızca bir değer kullanılmıştır.

AdaBoost algoritmasında kullanılan değişkenler ve değerleri Tablo 4.9'da gösterilmektedir. Elde edilen ortalama hata, medyan hata ve R^2 değerleri Tablo 4.10'da verilmiştir.

Tablo 4.9. AdaBoost algoritmasında kullanılan değişkenler ve değerleri

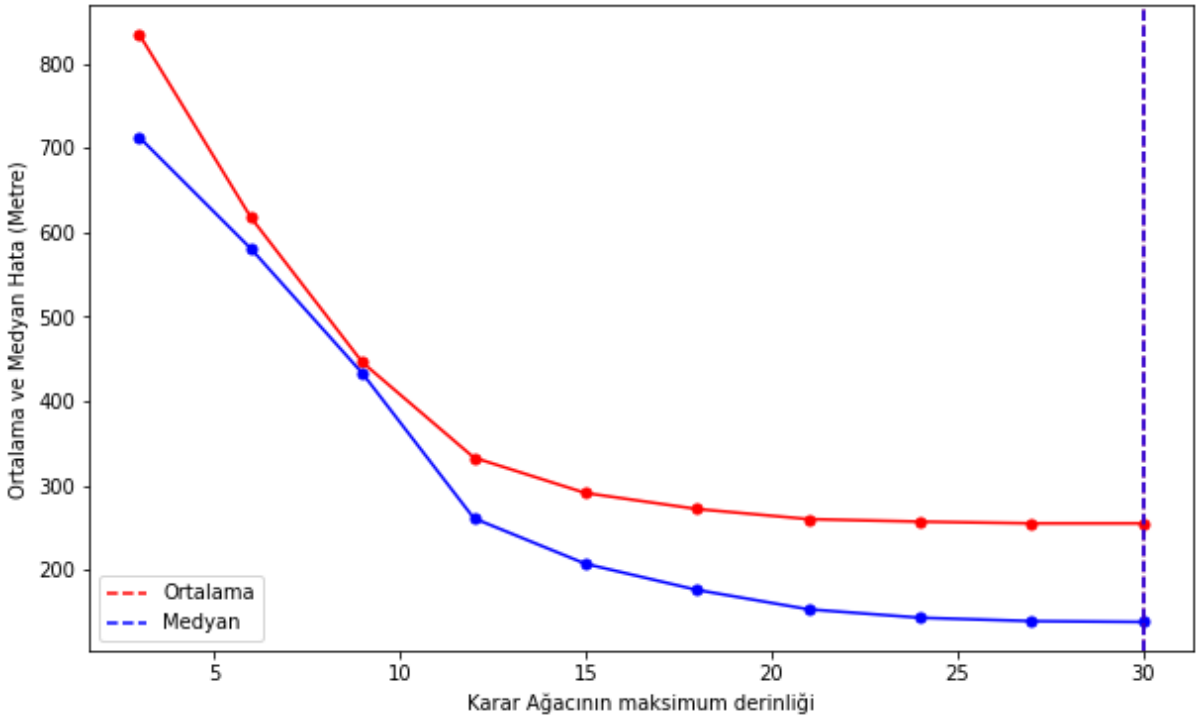
Parametre Adı	Parametre Açıklaması	Parametre Değeri
base_estimator ve içinde max_depth parametresi var	Temel tahmin edici, max_depth parametresine sahip DecisionTreeRegressor'dur	3,6,9,12,15,18, 21,24,27,30
n_estimators	Artırmanın sonlandırıldığı maksimum tahmin edici sayısı	50
learning_rate	Her modelin ağırlıklara katkısı	1.0, 0.1

loss	Her artırma yinelemesinden sonra ağırlıkları güncellerken kullanılacak kayıp işlevi	linear

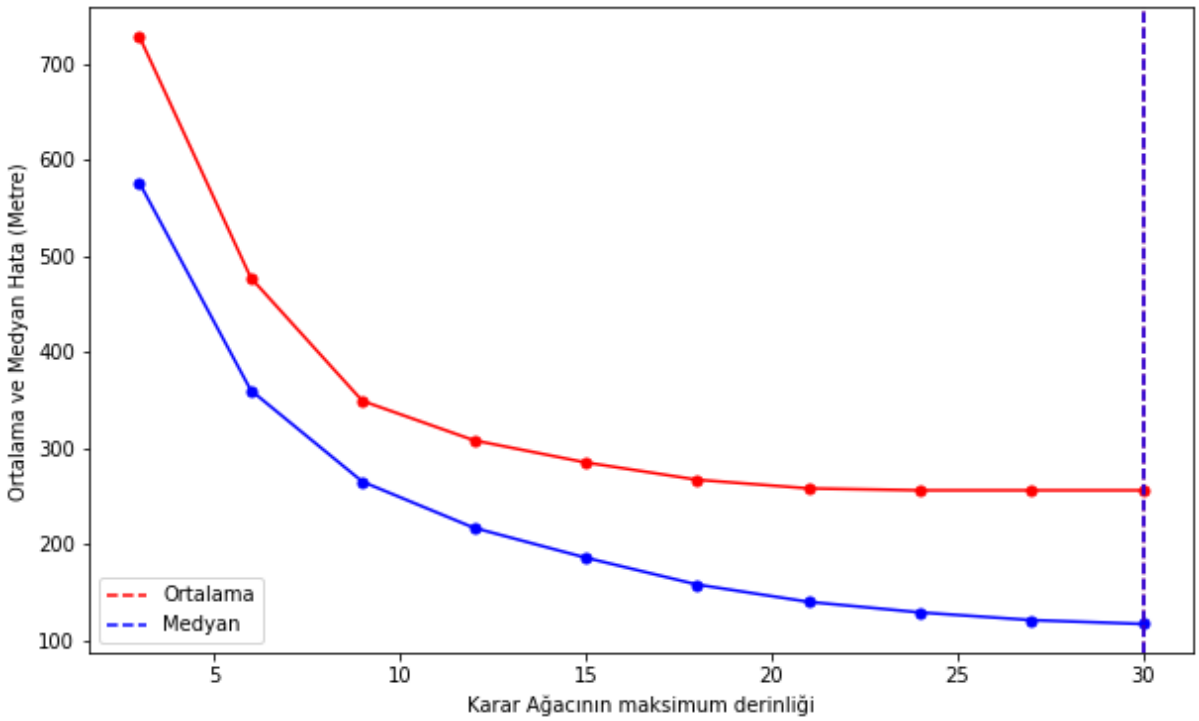
Tablo 4.10. AdaBoost algoritması ile elde edilen ortalama hata, medyan hata ve R^2 değerleri

Max. Derinlik	Öğrenme Oranı 1.0			Öğrenme Oranı 0.1		
	Ortalama Hata (metre)	Medyan Hata (metre)	R^2	Ortalama Hata (metre)	Medyan Hata (metre)	R^2
3	835	713	0.67	728	576	0.71
6	618	581	0.81	477	360	0.86
9	446	433	0.89	349	265	0.91
12	333	261	0.92	308	217	0.92
15	291	207	0.93	285	186	0.93
18	272	176	0.93	267	158	0.93
21	260	153	0.93	258	140	0.93
24	257	143	0.93	256	129	0.93
27	255	139	0.93	256	121	0.93
30	**255**	138	0.93	256	**117**	0.93

Maksimum derinlik 30 ve öğrenme oranı 1.0 olduğunda en düşük ortalama hata değeri 255 metre, maksimum derinlik 30 ve öğrenme oranı 0.1 olduğunda en düşük medyan hata değeri 117 metre ve maksimum derinlik 15 ile 30 arasında ve öğrenme oranı 1.0 ya da 0.1 olduğunda en yüksek R^2 elde edilmiştir. Öğrenme oranı 1.0 iken, ortalama hata ve medyan hata ile karar ağacının maksimum derinliği arasındaki ilişkiyi gösteren grafik Şekil 4.7'de gösterilmektedir. Öğrenme oranı 0.1 iken, ortalama hata ve medyan hata ile karar ağacının maksimum derinliği arasındaki ilişkiyi gösteren grafik Şekil 4.8'de gösterilmektedir.



Şekil 4.7. Öğrenme oranı 1.0 iken, ortalama hata ve medyan hata ile karar ağacının maksimum derinliği arasındaki ilişki



Şekil 4.8. Öğrenme oranı 0.1 iken, ortalama hata ve medyan hata ile karar ağacının maksimum derinliği arasındaki ilişki

4.1.6. Tüm Algoritmalar Arasındaki Karşılaştırma ile Elde Edilen Sonuçlar

Bu bölümde gerçeklenen ve test edilen beş algoritma için ortalama hata ve medyan hatanın karşılaştırması Tablo 4.11'de yer almaktadır. En düşük ortalama hata değeri ve en düşük medyan hata değeri sırasıyla 255 metre ve 117 metre ile AdaBoost algoritması kullanıldığında elde edilmiştir.

Tablo 4.11. Tüm algoritmalar ile elde edilen ortalama hata ve medyan hata karşılaştırılma tablosu

Algoritma Adı	Ortalama Hata (metre)	Medyan Hata (metre)	Kullanılan Değişkenler ve Değerleri
KNN	335	134	K=1
	303	195	K=7
Ağırlıklı kNN	298	180	K=7 ya da 9
Rastgele Orman	270	158	n_estimators =400 ile 500 arasında
Gradyan Artırma	283	186	max_depth= 8 & learning_rate= 0.15
	285	185	max_depth= 9 & learning_rate= 0.15
AdaBoost	256	**117**	max_depth= 30 & learning_rate= 0.1
	255	138	max_depth= 30 & learning_rate= 1.0

4.2. HİBRİT YÖNTEMLER İLE ELDE EDİLEN SONUÇLAR

Her hibrit yöntem için ortalama hata, medyan hata ve belirleme katsayısı hesaplanmıştır. Hibrit sonuçlar, Tablo 4.12'de verilen parametreleri kullanan üç veya iki algoritmanın tahmin ortalamasıdır. Elde edilen ortalama hata, medyan hata ve R^2 değerleri Tablo 4.13'te yer almaktadır. Bu parametrelerin, Bölüm 4.1'de yapılan testlerin sonuçlarının değerlendirilmesi sonucu belirlenmiştir.

Tablo 4.12. Tüm algoritmalarda kullanılan değişkenler ve değerleri

Algoritma Adı	Kullanılan Değişkenler ve Değerleri
KNN- Ağırlıklı kNN	n_neighbors=7, algorithm = brute , p=2
Rastgele Orman	n_estimators = 200
Gradyan Artırma	n_estimators =800, max_depth = 8, learning_rate=0.15, loss=huber
AdaBoost	max_depth = 30, learning_rate =0.1

Tablo 4.13. Hibrit yöntemler ile elde edilen ortalama hata, medyan hata ve R^2 değerleri

KNN	Ağırlıklı kNN	Rastgele Orman	Gradyan Artırma	AdaBoost	Ortalama Hata (metre)	Medyan Hata (metre)	R^2
✓	-	✓	✓	-	266	170	0.93
✓	-	✓	-	✓	261	156	0.93
✓	-	-	✓	✓	260	159	0.93
-	✓	✓	✓	-	264	166	0.93
-	✓	✓	-	✓	260	151	0.93
-	✓	-	✓	✓	259	154	0.93
-	-	✓	✓	✓	257	150	0.93
-	-	✓	✓	-	266	168	0.93
-	-	-	✓	✓	260	149	0.93
-	-	✓	-	✓	**254**	**137**	0.93

Rastgele Orman ve AdaBoost algoritmalarına sahip hibrit yöntem kullanıldığında en düşük ortalama hata ve medyan hata sırasıyla 254 metre ve 137 metre elde edilmiştir. Bu sonuçların ışığında sonraki bölümde Rastgele Orman ve AdaBoost algoritmaları seçilmiştir.

4.3. ÖNERİLEN METODOLOJİ İLE ELDE EDİLEN SONUÇLAR

Önerilen metodolojide farklı küme sayıları kullanılmıştır. Rastgele Orman ve AdaBoost algoritmalarında kullanılan değişkenler ve değerleri Tablo 4.14'te gösterilmektedir.

Tablo 4.14. Rastgele Orman ve AdaBoost algoritmalarda kullanılan değişkenler ve değerleri

Algoritma Adı	Kullanılan değişkenler ve değerleri
Rastgele Orman	n_estimators = 400
AdaBoost	max_depth=30 , learning_rate=0.1

4.3.1. AdaBoost Kullanarak Önerilen Metodoloji ile Elde Edilen Sonuçlar

Önerilen metodolojide AdaBoost algoritması kullanılarak elde edilen ortalama hata, medyan hata ve R^2değerleri Tablo 4.15'de verilmiştir.

Tablo 4.15. AdaBoost kullanarak önerilen metodoloji ile elde edilen ortalama hata, medyan hata ve R^2 değerleri

Küme Sayısı	Ortalama Hata (metre)	Medyan Hata (metre)	R^2
1	**256**	116	0 .93
5	267	111	0.92
10	276	107	0.90
15	282	104	0.90
20	289	104	0.90
25	289	102	0.90
30	289	101	0.90
50	290	**97**	0.90
60	290	99	0.90
70	290	100	0.90
100	291	101	0.90
200	283	108	0.91
250	281	112	0.91
300	277	112	0.91
350	274	110	0.91
400	277	111	0.91
1000	267	110	0.92
3000	262	113	0.92

4.3.2. Rastgele Orman Kullanarak Önerilen Yöntem ile Elde Edilen Sonuçlar

Önerilen metodolojide Rasgele Orman algoritması kullanılarak elde edilen ortalama hata, medyan hata ve R^2değerleri Tablo 4.16'da verilmiştir.

Tablo 4.16. Rasgele Orman kullanarak önerilen metodoloji ile elde edilen ortalama hata, medyan hata ve R^2 değerleri

Küme Sayısı	Ortalama Hata (metre)	Medyan Hata (metre)	R^2
1	**271**	160	0.93
5	275	144	0.92
10	281	134	0.91

15	286	120	0.90
20	291	116	0.90
25	291	115	0.90
30	291	114	0.90
50	293	107	0.90
60	294	110	0.90
70	289	115	0.91
100	290	129	0.91
200	281	137	0.92
250	281	140	0.92
300	280	142	0.92
350	277	140	0.92
400	278	141	0.92
1000	273	148	0.92
3000	272	156	0.93

4.3.3. Hibrit Yöntem Kullanılarak Önerilen Metodoloji ile Elde Edilen Sonuçlar

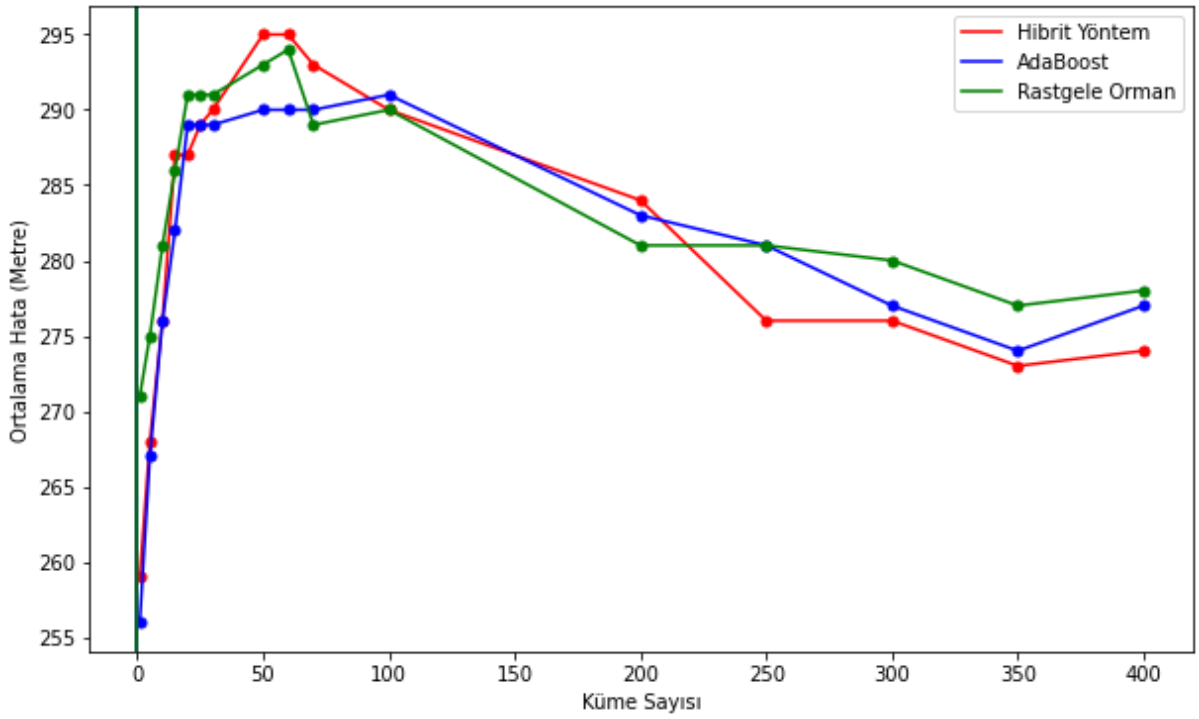
Elde edilen ortalama hata, medyan hata ve R^2 değerleri Tablo 4.17'de verilmiştir. Üç yaklaşımın ortalama hatası ile küme sayısı arasındaki ilişkiyi gösteren grafik Şekil 4.9'da gösterilmektedir. Şekil 4.10 ise üç yaklaşımın medyan hatası ile küme sayısı arasındaki ilişkiyi göstermektedir.

Tablo 4.17. Hibrit yöntem kullanarak önerilen yöntem ile elde edilen ortalama hata, medyan hata ve $\boldsymbol{R^2}$ değerleri

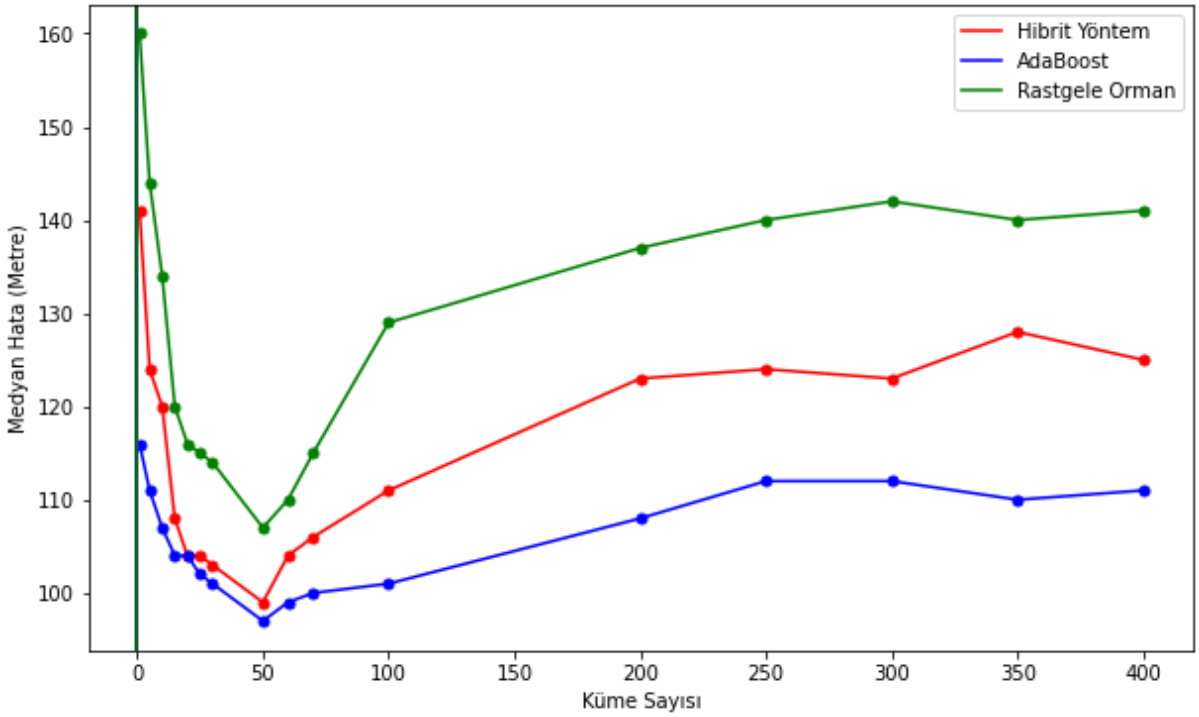
Küme Sayısı	**Ortalama Hata (metre)**	**Medyan Hata (metre)**	$\boldsymbol{R^2}$
1	**259**	141	0.93
5	268	124	0.92
10	276	120	0.91
15	287	108	0.90
20	287	104	0.90
25	289	104	0.90
30	290	103	0.90
50	295	**99**	0.90

60	295	104	0.90
70	293	106	0.90
100	290	111	0.91
200	284	123	0.91
250	276	124	0.92
300	276	123	0.92
350	273	128	0.92
400	274	125	0.92
1000	268	132	0.92
3000	262	135	0.93

Hibrit yöntem kullanılarak önerilen yöntemin elde edilen tüm sonuçları EK.1'de verilmiştir.



Şekil 4.9. Üç yaklaşımın ortalama hatası ile küme sayısı arasındaki ilişki



Şekil 4.10. Üç yaklaşımın Medyan hatası ile küme sayısı arasındaki ilişki

5. TARTIŞMA

Bu tez çalışmasının birinci kısmında gerçeklenen ve test edilen beş algoritma arasında yapılan karşılaştırma sonuçlarında AdaBoost yöntemi ile sırasıyla 255 metre ve 117 metre en düşük ortalama hata ve medyan hata değerleri elde edilmiştir. AdaBoost'un ardından Rastgele Orman algoritması ile 270 metre ortalama hata ve 158 metre medyan hata sonuçları elde edilmiştir. Janssen ve arkadaşları [29] 'da bu tezde kullanılan veri setini kullanarak Rastgele Orman ile 351 metre, ağırlıklı kNN ile 348 metre ve kNN ile 354 metre ortalama hata elde etmişlerdir. Bu tez çalışmasında veri setine uygulanan veri ön işlemesi adımları sayesinde elde edilen sonuçlarda literatürde yer alan sonuçlara kıyasla büyük oranda bir iyileşme elde edilmiştir. Ayrıca Rastgele Orman yöntemi için elde edilen ortalama hata, yine literatürde daha önce elde edilen ortalama hata değerinden 96 metre daha düşüktür. İkinci kısımda, Rastgele Orman ve AdaBoost algoritmalarının kullanıldığı hibrit yöntem ile diğer hibrit yöntemlere göre en düşük ortalama hata 254 metre ve en düşük medyan hata 137 metre olarak elde edilmiştir.

Son olarak tez çalışmasının üçüncü kısmında, Şekil 4.10'da gösterildiği gibi, önerilen metodolojinin yalnızca AdaBoost ile kullanılması durumunda, Rastgele Orman veya hibrit yöntemle kullanılmasına kıyasla daha düşük medyan hata meydana gelmektedir. Ayrıca küme sayısı ne olursa olsun tek başına AdaBoost algoritmasına kıyasla önerilen yöntem daha düşük medyan hata vermektedir. AdaBoost kullanılan önerilen yöntemde medyan hata aralığı 97-116 metre arasında tespit edilmiştir. Önerilen metodolojinin Rastgele Orman ile kullanılması, tek başına Rastgele Orman kullanımına kıyasla medyan hata değerlerinde de iyileşmeler göstermiştir.

Önerilen metodoloji hem Rastgele Orman hem de AdaBoost veya bunlardan biri ile kullanırken, küme sayısı ne kadar düşük olursa ortalama hatanın o kadar düşük olmasına sebep olmaktadır. Ayrıca küme sayısı ne kadar yüksek olursa medyan hatanın da o kadar düşük olmasına neden olmaktadır. Diğer bir yanda, çok sayıda küme kullanıldığında, sınıflandırma aşamasındaki hassasiyet arttığı için ortalama hata değeri artmakta, bu da verilerin yanlış kümede sınıflandırılması durumunda bir sonraki aşamada tahmin üzerinde kötü bir etki yaratabilmektedir. Ayrıca, sınıflandırıcı doğru kümeyi tahmin ettiğinde, küme özelliği, doğru tahmine daha fazla ağırlık verdiği için medyan hata değeri azalır. Küme sayısındaki artış çok

yüksek olduğunda küme özelliğinin etkisi azalmaya başlamakta ve ortalama hata oranı tekrar azalmaktadır.

Önerilen metodolojiyi AdaBoost ile kullanmak, her durumda 97-116 metre aralığında daha düşük medyan hata vermektedir. Metodoloji, hibrit yöntemle gerçeklendiğinde ve küme sayısı 10-100 aralığında olduğunda 99 – 112 metre aralığında medyan hata, küme sayısı 10'dan düşük veya 100'den büyük olduğunda yine 112-139 aralığında ve düşük olarak nitelendirilebilen bir medyan hata elde edilir.

Önerilen yöntemin sonuçları aynı veri setini kullanan diğer çalışmalarla da karşılaştırılmıştır. Li ve arkadaşları [30] 'da 346.03 metrelik bir ortalama hata ve 158.41 metrelik bir medyan hata elde etmişlerdir. Başka bir çalışmada Anagnostopoulos ve arkadaşları sinir ağlarını kullanarak 357 metre ortalama hata ve 206 metre medyan hata elde etmişlerdir [13].

Literatürde yer alan çalışmalar ile karşılaştırıldığında ortalama hata ve medyan hata değerleri sırasıyla 191-357 metre ve 158-206 metre aralıklarında olup bu tez çalışmasının birinci kısmında ortalama hata ve medyan hata sırasıyla 255 metre ve 117 metre, ikinci kısmında 254 metre ve 137 metre, üçüncü kısmında ise 256 metre ve 97 metre olarak elde edilmiştir. Bu tez çalışmasında elde edilen ortalama hata değerleri ile karşılaştırılan diğer çalışmaların sonuçlarının birbirine çok yakın ve tutarlı olduğu görülmüştür. Elde edilen medyan hata değerlerinde ise diğer çalışmalara göre bir iyileşme sağlanmıştır.

6. SONUÇ VE ÖNERİLER

Nesnelerin interneti; veri toplama, paylaşma ve analiz etme sürecini daha kolay, daha verimli ve ekonomik hale getirme yeteneği nedeniyle günümüzde bu amaçla kullanılan ve öne çıkan bir yaklaşımdır. Nesnelerin internetinde, nesnelerin arasındaki bağlantıyı sağlamak için çok çeşitli iletişim teknolojileri kullanılabilmektedir; ancak geleneksel kablosuz ağların eksiklikleri, düşük güçlü geniş alan ağı (LPWAN) adlı yeni bir kablosuz iletişim teknolojisinin ortaya çıkmasına neden olmuştur. LoRaWAN ise düşük maliyeti, uzun pil ömrü ve güvenilir mobil iletişim özellikleriyle konum tabanlı uygulamalar için en iyi seçenek olarak kabul edilmiştir.

Bu tez çalışmasında, kullanılan algoritmaları ve önerilen metodolojiyi test etmek ve değerlendirmek için halka açık bir LoRaWan veri seti kullanılmıştır. RSSI ölçümleri tabanlı konum tahmini için beş makine öğrenmesi algoritması karşılaştırılmış ve performansları incelenmiştir. Her algoritma farklı parametreler kullanılarak farklı senaryolarda test edilmiştir.

Ayrıca daha sonra, farklı hibrit yöntemler değerlendirilmiş ve karşılaştırılmıştır. Algoritmaların, hibrit yöntemlerin ve önerilen metodolojinin sonuçları, aynı veri setini kullanan diğer çalışmalarla karşılaştırılmıştır. Diğer çalışmalar ile kıyaslandığında, elde edilen ortalama hata değerlerinin çok yakın ve tutarlı olduğu görülmüştür. Elde edilen medyan hata değerlerinde ise diğer çalışmalara göre bir iyileşme sağlanmıştır.

Sonuç olarak, bu tez çalışmasının birinci kısmında elde edilen ortalama hata ve medyan hata sırasıyla 255 metre ve 117 metre, ikinci kısmında 254 metre ve 137 metre ve üçüncü kısmında ise 256 metre ve 97 metredir.

Gelecekte yapılacak olan çalışmalarda önerilen metodoloji farklı veri seti ve farklı regresyon algoritmaları kullanılarak değerlendirilebilir. Ayrıca hem TDoA hem de RSSI parmak izi tekniklerini kullanan hibrit bir çözüm ile doğruluğu daha fazla iyileştirilebilir ve LoRaWAN konum belirleme işlevselliği geliştirilebilir.

KAYNAKLAR

[1] R. S. Sinha, Y. Wei and S.-H. Hwang, "A survey on LPWA technology: LoRa and NB-IoT," *Science Direct,* 2017.

[2] M. Centenaro, L. Vangelista, A. Zanella and M. Zorzi, "Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios," 2016.

[3] D. Patel and M. Won, "Experimental Study on Low Power Wide Area Networks (LPWAN) for Mobile Internet of Things," *IEEE,* 2017.

[4] U. Raza, P. Kulkarni and M. Sooriyabandara, "Low Power Wide Area Networks: An Overview," *IEEE,* 2017.

[5] K. Mekki, E. Bajic, F. Chaxel and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," *Science Direct,* 2017.

[6] W. Guibene, J. Nowack, N. Chalikias, K. Fitzgibbon, M. Kelly and D. Prendergast, "Evaluation of LPWAN Technologies for Smart Cities: River Monitoring Use-Case," *IEEE,* 2017.

[7] «Sigfox,» Sigfox, 2010. [Çevrimiçi]. Available: https://www.sigfox.com/en. [Erişildi: 20 June 2022].

[8] "Lora Alliance - About LoRaWAN," [Online]. Available: lora-alliance.org/about-lorawan/. [Accessed 23 February 2022].

[9] K. Mekki, E. Bajic, F. Chaxel and F. Meyer, "Overview of Cellular LPWAN Technologies for IoT Deployment: Sigfox, LoRaWAN, and NB-IoT," *IEEE,* 2018.

[10] "Lora Alliance - What is LoRaWAN," [Online]. Available: lora-alliance.org/resource_hub/what-is-lorawan/. [Accessed 23 February 2022].

[11] P. S. Cheong, J. Bergs, C. Hawinkel and J. Famaey, "Comparison of LoRaWAN classes and their power consumption," *IEEE,* 2017.

[12] B. C. Fargas ve M. N. Petersen, «GPS-free Geolocation using LoRa in Low-Power WANs,» *IEEE,* 2017.

[13] G. G. Anagnostopoulos and A. Kalousis, "A Reproducible Comparison of RSSI Fingerprinting Localization Methods Using LoRaWAN," *IEEE,* 2019.

[14] M. Aernouts, R. Berkvens, K. Van Vlaenderen and M. Weyn, "Sigfox and LoRaWAN Datasets for Fingerprint Localization in Large Urban and Rural Areas," 19 July 2019. [Online]. Available: https://zenodo.org/record/3904158#.YlL6TchBxPb. [Accessed 1 February 2022].

[15] M. Dogan, «Indoor Localization And Tracking Based on RSSI and Accelerometer Measurements,» MIDDLE EAST TECHNICAL UNIVERSITY, Ankara, 2015.

[16] S. Yiu n, M. Dashti, H. Claussen and F. Perez-Cruz, "WirelessRSSI fingerprintinglocalization," *Elsevier,* 2017.

[17] Y. HAZNEDAR, «Wi-Fi ve BLE hibrit teknolojilere dayalı makine öğrenme destekli iç mekan konumlandırma,» ULUSAL TEZ MERKEZİ, İstanbul, 2021.

[18] G. G. Anagnostopoulos and A. Kalousis, "A Reproducible Analysis of RSSI Fingerprinting for Outdoor Localization Using Sigfox: Preprocessing and Hyperparameter Tuning," *IEEE,* 2019.

[19] "LoRa - Developer Portal," [Online]. Available: lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/. [Accessed 23 February 2022].

[20] P. Chaudhari, A. K. Tiwari, S. Pattewar and S. N. Shelke, "Smart Infrastructure Monitoring using LoRaWAN Technology article," *IEEE,* 2021.

[21] W. Zhao, S. Lin, J. Han, R. Xu and L. Hou, "Design and Implementation of Smart Irrigation System Based on LoRa," *IEEE,* 2017.

[22] R. K. Kodali, S. Yerroju ve S. Sahu, «Smart Farm Monitoring Using LoRa Enabled IoT,» *IEEE,* 2018.

[23] T. Porselvi, S. Ganesh CS, B. Janaki, K. Priyadarshini and S. Shajitha Begam, "IoT Based Coal Mine Safety and Health Monitoring System using LoRaWAN," *IEEE,* 2021.

[24] L. A. S. Committee, "LoRaWAN GEOLOCATION WHITEPAPER Prepared by: LoRa Alliance™ Strategy Committee," [Online]. Available: https://lora-alliance.org/. [Accessed 24 February 2022].

[25] W. R. D. Silva, L. Oliveira, N. Kumar, R. A. Rabêlo, C. N. Marins and J. J. P. C. Rodrigues, "An Internet of Things Tracking System Approach Based on LoRa Protocol," *IEEE,* 2019.

[26] P. Guan, Z. Zhang, L. Wei and Y. Zhao, "A Real-Time Bus Positioning System Based on LoRa Technology," *IEEE,* 2018.

[27] D. Plets, N. Podevijn, J. Trogh, L. Martens ve W. Joseph, «Experimental Performance Evaluation of Outdoor TDoA and RSS Positioning in a Public LoRa Network,» *IEEE,* 2018.

[28] N. Podevijn, J. Trogh, M. Aernouts, R. Berkvens, L. Martens, M. Weyn, W. Joseph ve D. Plets, «Compass Aided TDoA Tracking in LoRaWAN networks,» *IEEE,* 2020.

[29] T. Janssen, R. Berkvens and M. Weyn, "Comparing Machine Learning Algorithms for RSS-based localization in LPWAN," *Research Gate,* 2020.

[30] Y. Li, J. Barthelemy, S. Sun, P. Perez and B. Moran, "Urban vehicle localization in public LoRaWan," *IEEE,* 2021.

[31] P. Banerjee, "Bagging vs Boosting," kaggle, 2020. [Online]. Available: https://www.kaggle.com/code/prashant111/bagging-vs-boosting/notebook. [Accessed 16 April 2022].

[32] H. Sharma and K. , "A Survey on Decision Tree Algorithms of Classification in Data Mining," *International Journal of Science and Research (IJSR),* vol. 5, no. 4, p. 4, 2016.

[33] I. C. Education, "Random Forest," IBM, 7 December 2020. [Online]. Available: https://www.ibm.com/cloud/learn/random-forest. [Accessed 15 April 2022].

[34] R. Garg, "A Primer to Ensemble Learning – Bagging and Boosting," Analytics India Magazine, 19 February 2018. [Online]. Available: https://analyticsindiamag.com/primer-ensemble-learning-bagging-boosting/. [Accessed 15 April 2022].

[35] Y. Freund and R. E. Schapire, "A Short Introduction to Boosting," *Journal of Japanese Society for Artificial Intelligence,* 1999.

[36] T. Shin, "A Mathematical Explanation of AdaBoost," Towards data science, 9 July 2020. [Online]. Available: https://towardsdatascience.com/a-mathematical-explanation-of-adaboost-4b0c20ce4382. [Accessed 24 04 2022].

[37] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," *jstor,* 2001.

[38] s. saxena, "Gradient Boosting Machine for Data Scientists," Analytics Vidhya, 25 march 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/03/gradient-boosting-machine-for-data-scientists/. [Accessed 25 April 2022].

[39] N. Altman, "An Introduction to Kernel and Nearest Neighbor Nonparametric Regression," *Cornell University Library ,* 1991.

[40] I. C. Education, "What is Supervised Learning?," IBM, 19 August 2020. [Online]. Available: https://www.ibm.com/cloud/learn/supervised-learning. [Accessed 15 April 2022].

[41] Z. Guowei, X. Zhan and L. Dan, "Research and improvement on indoor localization based on RSSI fingerprint database and K-nearest neighbor points," *IEEE,* 2013.

[42] D. M. J. Garbade, "Understanding K-means Clustering in Machine Learning," 13 September 2018. [Online]. Available: https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1#:~:text=K%2Dmeans%20clustering%20is%20one,known%2C%20or%20labelled%2C%20outcomes.. [Accessed 09 May 2022].

[43] H. Bonthu, “K-means Clustering,” IBM, 08 August 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/08/kmeans-clustering/. [Accessed 09 May 2022].

[44] P. Dangeti, “The Elbow Method,” oreilly, [Online]. Available: https://www.oreilly.com/library/view/statistics-for-machine/9781788295758/c71ea970-0f3c-4973-8d3a-b09a7a6553c1.xhtml. [Accessed 09 May 2022].

[45] E. G. Agency, “Understanding GNSS performance on android using the GPS testc app,” 2018. [Online]. Available: https://www.euspa.europa.eu/sites/default/files/understanding_gnss_performance_on_android_using_the_gps_testc_app.pdf. [Accessed 10 April 2022].

[46] A. Bloomenthal, “Coefficient of Determination,” 10 October 2021. [Online]. Available: https://www.investopedia.com/terms/c/coefficient-of-determination.asp. [Accessed 29 May 2022].

EKLER

EK1. HİBRİT YÖNTEM KULLANILARAK ÖNERİLEN YÖNTEMİN AYRINTILI SONUÇLARI

Küme Sayısı	Ortalama Hata (metre)	Medyan Hata (metre)	R^2
1	**259**	141	0.93
2	262	139	0.93
3	264	133	0.92
4	269	130	0.92
5	268	124	0.92
6	269	119	0.92
7	272	119	0.91
8	274	118	0.91
9	278	115	0.91
10	276	120	0.91
11	282	112	0.90
12	285	114	0.90
13	287	109	0.90
14	288	111	0.90
15	287	108	0.90
16	286	108	0.90
17	288	106	0.90
18	286	104	0.90
19	288	104	0.90
20	287	104	0.90
21	289	106	0.90
22	289	107	0.90
23	289	106	0.90
24	290	106	0.90
25	289	104	0.90

26	287	104	0.90
27	290	105	0.90
28	289	103	0.90
29	290	102	0.90
30	290	103	0.90
31	292	104	0.90
32	290	101	0.90
33	293	103	0.90
34	293	103	0.90
50	295	**99**	0.90
60	295	104	0.90
70	293	106	0.90
100	290	111	0.91
200	284	123	0.91
250	276	124	0.92
300	276	123	0.92
350	273	128	0.92
400	274	125	0.92
1000	268	132	0.92
3000	262	135	0.93

İNTİHAL RAPORU İLK SAYFASI

Gufran Bayraktar

ORİJİNALLİK RAPORU

%9	%6	%1	%5
BENZERLİK ENDEKSİ	İNTERNET KAYNAKLARI	YAYINLAR	ÖĞRENCİ ÖDEVLERİ

BİRİNCİL KAYNAKLAR

1	Submitted to The Scientific & Technological Research Council of Turkey (TUBITAK) Öğrenci Ödevi	%4
2	acikbilim.yok.gov.tr İnternet Kaynağı	%1
3	Submitted to Tobb University of Economics & Technology Öğrenci Ödevi	<%1
4	www.yumpu.com İnternet Kaynağı	<%1
5	igs2018.mu.edu.tr İnternet Kaynağı	<%1
6	9lib.net İnternet Kaynağı	<%1
7	nek.istanbul.edu.tr:4444 İnternet Kaynağı	<%1
8	Submitted to Beykent Universitesi Öğrenci Ödevi	<%1
9	hdl.handle.net	

ETİK KURUL İZİN YAZISI

Uyarı: Canlı denekler üzerinde yapılan tüm araştırmalar için Etik Kurul Belgesi alınması zorunludur.

☐ **Etik Kurul izni gerekmektedir.**

☒ **Etik Kurul izni gerekmemektedir.**

Gufran BAYRAKTAR
(İmza)

KURUM İZNİ YAZILARI

Uyarı: Canlı ve cansız deneklerle yapılan tüm çalışmalar için kurum izin belgelerinin eklenmesi zorunludur. Gizlilik ve mahremiyet içeren durumlarda kurum adı kapatılmalıdır.

☐ **Kurum izni gerekmektedir.**

☒ **Kurum izni gerekmemektedir.**

Gufran BAYRAKTAR
(İmza)