

T.C.
İSTANBUL SABAHATTİN ZAİM ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR BİLİMİ VE MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR BİLİMİ VE MÜHENDİSLİĞİ BİLİM DALI

VÜCUT KİNEMATİĞİ İZLEME VERİLERİNİN
ANALİTİĞİ

YÜKSEK LİSANS TEZİ

Ayşenur GENÇDOĞMUŞ

İstanbul
Haziran, 2019

T.C.
İSTANBUL SABAHATTİN ZAİM ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR BİLİMİ VE MÜHENDİSLİĞİ
YÜKSEK LİSANS PROGRAMI

VÜCUT KİNEMATİĞİ İZLEME VERİLERİNİN ANALİTİĞİ

YÜKSEK LİSANS TEZİ

Ayşenur GENÇDOĞMUŞ

Tez Danışmanı
Doç. Dr. Gülüstan DOĞAN

Tez 2. Danışmanı
Prof. Dr. Yusuf ÖZTÜRK

İstanbul
Haziran, 2019

Fen Bilimleri Enstitüsü Müdürlüğüne,

Bu çalışma, jürimiz tarafından Bilgisayar Mühendisliği Anabilim Dalı, Bilgisayar Bilimleri ve Mühendisliği Bilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman

Doç. Dr.  Gülistan DOĞAN

Üye

Dr. Öğr. Üyesi  Farzad KIANI

Üye

Dr. Öğr. Üyesi  Yahya ŞİRİN

Onay

Yukarıdaki imzaların, adı geçen öğretim üyelerine ait olduğunu onaylıyorum.


Prof. Dr. Ahmet Korhan BİNARK
Enstitü Müdürü

BİLİMSEL ETİK BİLDİRİMİ

Yüksek lisans tezi olarak hazırladığım "Vücut Kinematiki İzleme Verilerinin Analitiki" adlı çalışmanın öneri aşamasından sonuçlandırıldığı aşamaya kadar geçen süreçte bilimsel etiğe ve akademik kurallara özenle uyduğumu, tez içindeki tüm bilgileri bilimsel ahlak ve gelenek çerçevesinde elde ettiğimi, tez yazım kurallarına uygun olarak hazırladığımı, bu çalışmada doğrudan veya dolaylı olarak yaptığım her alıntıya kaynak gösterdiğimi ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu beyan ederim.

Ayşenur Gençdoğmuş



ÖNSÖZ

Araştırmamdaki her aşamada bana yardımcı olan değerli tez danışmanım Doç. Dr. Gülüstan DOĞAN'a, yaptığı çalışmalarla bize vizyon katan çok değerli hocam tez eş danışmanım Prof. Dr Yusuf ÖZTÜRK'e, destek ve yardımları için sevgili Gözde Merve Demirci'ye, Şeref Recep Keskin'e ve Merve Kayhan hocama, lisans ve yüksek lisans eğitimim boyunca benden desteklerini esirgemeyen aileme teşekkürlerimi sunarım.

Ayşenur GENÇDOĞMUŞ

İstanbul - 2019

ÖZET

VÜCUT KİNEMATİĞİ İZLEME VERİLERİNİN ANALİTİĞİ

Ayşenur Gençdoğmuş

Yüksek Lisans, Bilgisayar Bilimi ve Mühendisliği

Tez danışmanı: Doç Dr. Gülüstan Doğan

Tez 2. Danışmanı: Prof. Dr. Yusuf Öztürk

Haziran-2019, 101 Sayfa

İnsan etkinlikleri tanıma konusu; bilgisayar-insan etkileşimi, robotik, kablosuz sensör ağları gibi birçok konuları içermekte ve çok aktif bir araştırma konusu olmaktadır. Aktivitelerin tanınması üzerine araştırmalar ağırlıklı olarak RGB videolar ve giyilebilir sensörleri kapsamaktadır.

Aktivitelerin tanınması, tehlikeli olayları tespit etmek veya yalnız yaşayan insanları izlemek gibi birçok alana uygulanabilir. Özellikle günlük faaliyetlerin izlenmesi kişiler için daha önemli olmakla birlikte; yapılandırılmamış günlük aktivitelerin tanınması çok daha zor bir görev haline gelir.

Aktiviteleri doğru tanımlama; sensörlerin konumları, pil durumları, ortamda sensörleri etkileyecek çevresel faktörler, verideki gürültü gibi birçok parametreye bağlı olduğu için hala çözülmesi gereken birçok probleme sahiptir.

Çalışmanın temel amacı, kablosuz iki vücut sensöründen elde edilen vücut kinematığı verilerinin (ham verilerin) işlenerek derin öğrenme ve makine öğrenmesi metotlarının uygulanmasıdır. Son yıllarda derin öğrenme temelli yaklaşımlar, görüntü sınıflandırma, nesne tanıma vb. birçok alanda büyük başarılar elde etmiştir. Bu nedenle bu çalışmada makine öğrenmesi ve derin öğrenmesi yaklaşımlarını başarıları karşılaştırılmaktadır. Uygulanan metotların doğrulukları kıyaslanarak verimize en uygun metot veya metotlar belirlenmiş olacaktır. Metotların başarı kıyaslamalarına ek olarak çalışmamızda derin öğrenme metoduyla eğitilecek LSTM sinir ağında kullanılacak hiperparametreler için en uygun değerler test edilerek belirlenmiştir.

Metotlar uygulanmadan önce veride birçok ön işleme yapılmıştır. Kablosuz iki vücut sensörünün birbirine göre hareketleri inceleneceği için sensörlerden kaydedilen verilerin ardışık sırayla gelmesi beklenmektedir. Sırasız veya farklı sırayla gelen verilerin eşleştirilmesi için bir metot geliştirilmiştir. Eşleştirilen veri (ham veri)

üzerinde bir dizi dönüşüm işlemi yapılmıştır. Kuarterniyon cinsinden kaydedilen veri öncelikle rotasyon matrisine, rotasyon matrisinden de Euler açılara dönüştürülmüştür.

Kaydedilen veri, gerek ortamdaki kaynaklanan gerek sensörlerden kaynaklanan sorunlar nedeniyle gürültü bir veri olduğu için, verilerin eşleştirilmesi ve dönüştürülmesi işlemlerinden sonra filrlenmesi gerekmektedir. Bu amaçla, medyan filtre kullanılarak gürültülü sinyaller ve uç değerler düzeltilmiştir.

Filtrelenmiş veri üzerinde öncelikle LSTM sinir ağı eğittiğimiz derin öğrenme metodu için test edilerek en uygun hiperparametreler belirlenmiştir. Denetimli makine öğrenme metodlarından DecisionTree, RandomForest, GradientBoosting, AdaBoost, KNN, GaussianNaiveBayes metodları denenmiştir. Test edilen makine öğrenme metodlarının doğrulukları kendi aralarında ve derin öğrenme metodu ile karşılaştırılmış ve sonuç olarak LSTM sinir ağı geliştirilerek oluşturulmuş derin öğrenme metodunun daha başarılı olduğunu gösterilmiştir.

Anahtar Kelimeler: Büyük Veri Analizi, Veri Madenciliği, Makine Öğrenmesi, Derin Öğrenme, LSTM Sinir Ağı

ABSTRACT

BODY KINEMATICS MONITORING DATA ANALYTICS

Ayşenur Gençdoğmuş

Master of Science, Computer Science and Engineering

Supervisor: Assoc. Prof. Dr. Gülüstan Doğan

Co-Advisor: Prof. Dr. Yusuf ÖZTÜRK

June-2019, 101 Pages

Recognition of human activities; includes many threads such as computer-human interaction, robotics, wireless sensor networks and it is a very active research topic. Research on recognition of activities mainly includes RGB videos and wearable sensors. The recognition of activities can be applied to many areas, such as detecting dangerous events or watching people living alone. In particular, monitoring of daily activities is more important for people; recognition of unstructured daily activities becomes a much more difficult task. Accurate description of activities; Since it depends on many parameters such as the location of the sensors, battery conditions, environmental factors affecting the sensors in the environment, noise in the data, it still has many problems to be solved.

The main purpose of the study is to apply deep learning and machine learning methods by processing body kinematics data (raw data) obtained from two wireless body sensors. In recent years, deep learning based approaches, image classification, object recognition and so on. has achieved great success in many areas. Therefore, in this study, machine learning and deep learning approaches are compared. By comparing the accuracy of the applied methods, the most efficient methods or methods will be determined. In addition to the success comparisons of the methods, the most effective values for the hyperparameters to be used in the LSTM neural network to be trained with the deep learning method were determined by testing.

Before the methods were applied, several preprocesses were made on the raw data. Since the movements of the two wireless body sensors relative to each other will be examined, it is expected that the data recorded from the sensors will come in sequential order. A method has been developed for mapping data from unordered or different order. A series of conversions were performed on the matched data (raw data). The data recorded in quaternion were first converted to rotation matrix and from rotation matrix to Euler angles.

Since the recorded data was a noise data due to both environmental and sensor

problems, the data had to be filtered after mapping and conversion. For this purpose, noisy signals and extreme values were corrected using the median filter. The most appropriate hyperparameters were determined on the filtered data by testing the LSTM neural network for deep learning method. Decision Tree, Random Forest, Gradient Boosting, AdaBoost, KNN, GaussianNaiveBayes supervised machine learning methods were tried. The accuracy of the machine learning methods tested was compared with each other and with the deep learning method and as a result it was shown that the deep learning method which was formed by developing LSTM neural network was more successful.

Keywords: Big Data Analytics, Data Minig, Machine Learning, Deep Learning, LSTM Neural Network



DIŞ KAPAK	
İÇ KAPAK	
TEZ ONAYI	
BİLİMSEL ETİK BİLDİRİMİ.....	i
ÖNSÖZ.....	ii
ÖZET.....	iii
ABSTRACT.....	v
İÇİNDEKİLER.....	vii
TABLolar LİSTESİ.....	ix
ŞEKİLLER LİSTESİ.....	x
KISALTMALAR LİSTESİ.....	xii
GİRİŞ.....	1
1 BİRİNCİ BÖLÜM.....	3
1.1 LİTERATÜR TARAMASI.....	3
1.1.1 Sensörlerde Hareket Verisi Toplama.....	3
1.1.2 Sensörlerde Hareket Verisi Analizi.....	5
2 İKİNCİ BÖLÜM.....	11
2.1 VÜCUT KİNEMATİĞİ VERİLERİNİN TOPLANMASI.....	11
2.1.1 Kuaterniyonlar (Quaternions).....	13
2.2 VERİLERLE YAPILAN İŞLEMLER.....	15
2.2.1 Verilerin Temizlenmesi-Gruplara Ayrılması.....	17
2.2.2 Sensör Verilerinin Eşleştirilmesi.....	20
2.2.3 Rotasyon Matrisinin Hesaplanması.....	21
2.2.4 Rotasyon Matrislerinin Euler Açıklarına Dönüştürülmesi.....	21
2.3 VERİLERİN GÖRSELLEŞTİRİLMESİ.....	22

2.3.1	Gürültülü Verinin Görselleştirilmesi.....	22
2.3.2	Gürültülü Verinin Filtrelenmesi.....	29
2.3.2.1	Medyan Filtre.....	29
2.3.3	Filtrelenmiş Verinin Görselleştirilmesi.....	30
2.3.4	Verinin 3 Boyutlu Dağılım Grafiklerinin Çizilmesi.....	40
2.4	VERİNİN KÜMELENMESİ.....	43
2.4.1	Verideki Küme Sayısının Belirlenmesi.....	44
3	ÜÇÜNCÜ BÖLÜM.....	47
3.1	DERİN ÖĞRENME TABANLI AKTİVİTE TANIMA.....	47
3.1.1	Derin Öğrenme Metodolojisi.....	47
3.1.2	Vücut Kinematığı Verileri Üzerinde Derin Öğrenme Uygulaması.....	48
3.1.3	LSTM Sinir Ağının Oluşturulması.....	50
3.1.4	Veri Setinin Görselleştirilmesi.....	51
3.1.5	Derin Öğrenme Modelinin Oluşturulması.....	52
3.1.6	Derin Öğrenme Metodu Testleri.....	53
3.1.6.1	LSTM Katman Sayısının Karşılaştırılması.....	53
3.1.6.2	Eğitim Turu (Epoch) Sayısının Karşılaştırılması.....	59
3.1.6.3	Yığın Boyutunun (Batch Size) Karşılaştırılması.....	65
3.1.7	Değerlendirme Kriterleri.....	71
3.1.8	LSTM Sinir Ağı Eğitim Test Sonuçları.....	72
4	DÖRDÜNCÜ BÖLÜM.....	76
4.1	MAKİNE ÖĞRENMESİ TABANLI AKTİVİTE TANIMA.....	76
4.1.1	Makine Öğrenmesi Metodolojisi.....	76
4.1.2	Vücut Kinematığı Verileri Üzerinde Makine Öğrenmesi Uygulaması.....	76
4.1.3	Değerlendirme Kriterleri.....	79
4.1.4	Makine Öğrenmesi Test Sonuçları.....	80

5 BEŞİNCİ BÖLÜM.....	82
5.1 SONUÇLAR VE ÖNERİLER.....	82
REFERANSLAR.....	84
ÖZGEÇMİŞ.....	88

TABLolar LİSTESİ

Tablo 2.1: Genel Pozisyon Bilgileri.....	17
Tablo 2.2: Ayaktayken kaydedilen pozisyon bilgileri.....	18
Tablo 2.3: Otururken kaydedilen pozisyon bilgileri.....	18
Tablo 2.4: Fonksiyonel aktiviteler sırasında kaydedilen pozisyon bilgileri.....	18
Tablo 2.5: Ayaktayken Kaydedilen Pozisyonlara Ait Yaklaşık Süre Bilgileri.	24
Tablo 2.6: Otururken Kaydedilen Pozisyonlara Ait Yaklaşık Süre ve Kayıt Sayısı Bilgileri.....	26
Tablo 2.7: Fonksiyonel Aktiviteler Sırasında Kaydedilen Pozisyonlara Ait Yaklaşık Süre Bilgileri.....	27
Tablo 2.8: Diğer Aktiviteler Sırasında Kaydedilen Pozisyonların Yaklaşık Süre Bilgileri.....	28
Tablo 3.1: Eğitim Turu Sayısına Göre Sınıflandırma Değerleri.....	74
Tablo 3.2: Yığın Boyutuna Göre Sınıflandırma Değerleri.....	74
Tablo 3.3: LSTM Katman Sayısına Göre Sınıflandırma Değerleri.....	75
Tablo 4.1: Farklı Makine Öğrenmesi Metotlarına Göre Sınıflandırma Değerleri.....	81

ŞEKİLLER LİSTESİ

Şekil 2.1: Csv formatında kaydedilen verinin işlenmemiş hali.....	12
Şekil 2.2: x, y, z, w kuarterniyonlarına karşılık gelen dönme matrisi.....	15
Şekil 2.3: Veriden bir kesit (Seq_num, MAC_ID_of_sensor, Time_Stamp, Ignore ve Position).....	16
Şekil 2.4: Verinin kuarterniyon değerlerden oluşan diğer kısmı (Quat_X, Quat_Y, Quat_Z, Quat_W).....	17

Şekil 2.5: Görselleştirilen ham (gürültülü) veri örneği.....	23
Şekil 2.6: Ayaktayken kaydedilmiş (gürültülü) pozisyonların grafiği.....	25
Şekil 2.7: Otururken kaydedilmiş (gürültülü) pozisyonların grafiği.....	26
Şekil 2.8: Fonksiyonel aktiviteler sırasında kaydedilmiş (gürültülü) pozisyonların grafiği.....	25
Şekil 2.9: Ecological monitoring ve Walking outdoor aktiviteleri sırasında kaydedilmiş (gürültülü) pozisyonların grafiği.....	29
Şekil 2.10: Görselleştirilen filtrelenmiş veri örneği.....	30
Şekil 2.11 Ayaktayken kaydedilmiş pozisyonların filtrelenmiş grafiği.....	34
Şekil 2.12 Otururken kaydedilmiş pozisyonların filtrelenmiş grafiği,.....	36
Şekil 2.13: Fonksiyonel aktiviteler sırasında kaydedilmiş pozisyonların filtrelenmiş grafiği.....	38
Şekil 2.14: Ecological monitoring ve walking outdoor aktiviteleri sırasında ölçülmüş pozisyonların filtrelenmiş grafiği.....	39
Şekil 2.15: Ayaktayken kaydedilen pozisyonların 3 boyutlu dağılım grafikleri.....	40
Şekil 2.16: Otururken kaydedilen pozisyonların 3 boyutlu dağılım grafikleri.....	41
Şekil 2.17: Fonksiyonel aktiviteler sırasında ölçülmüş pozisyonların 3 boyutlu dağılım grafikleri	42
Şekil 2.18: Ecological monitoring ve Walking_outdoor aktiviteleri sırasında ölçülmüş pozisyonların 3 boyutlu dağılım grafikleri.....	42
Şekil 2.19: İki farklı pozisyona uygulanmış K-means algoritması sonucu.....	43
Şekil 2.20: Elbow metoduna göre verideki küme sayısını gösteren grafik.....	44
Şekil 2.21: Silhouette metoduna göre verideki küme sayısını gösteren grafik.....	45
Şekil 2.22: Gap Statistic metoduna göre verideki küme sayısını gösteren grafik.....	46
Şekil 2.23: Elbow-Gap Statistic-Sihouette metotları sonuçlarının karşılaştırılması.....	46
Şekil 3.1: LSTM sinir ağı şeması.....	48
Şekil 3.2: Derin Öğrenme modeli için yeniden düzenlenmiş (filtrelenmiş) vücut kinematiki verisi	49
Şekil 3.3: Derin Öğrenme yöntemine ait parametre ve hiper parametrelerin tanımlanması.....	50
Şekil 3.4: Veri setindeki her bir aktiviteye ait kayıt sayısı.....	51

Şekil 3.5: "Ecological Monitoring" isimli pozisyona ait x, y, z eksenlerindeki Euler açı değerleri	52
Şekil 3.6: LSTM katman sayısı 2 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi.....	54
Şekil 3.7: LSTM katman sayısı 4 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi.....	55
Şekil 3.8: LSTM katman sayısı 6 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi.....	56
Şekil 3.9: LSTM katman sayısı 8 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi.....	57
Şekil 3.10: LSTM katman sayısı 10 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi.....	58
Şekil 3.11: Eğitim turu 50 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi.....	60
Şekil 3.12: Eğitim turu 100 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi.....	61
Şekil 3.13: Eğitim turu 150 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi.....	62
Şekil 3.14: Eğitim turu 200 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi.....	63
Şekil 3.15: Eğitim turu 250 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi.....	64
Şekil 3.16: Yığın boyutu 32 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi.....	66
Şekil 3.17: Yığın boyutu 64 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi.....	67
Şekil 3.18: Yığın boyutu 128 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi.....	68
Şekil 3.19: Yığın boyutu 256 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi.....	69
Şekil 3.20: Yığın boyutu 512 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi.....	70
Şekil 3.21: LSTM katman sayısı - Doğruluk değişimi grafiği.....	72
Şekil 3.22: Eğitim Turu (Epoch) Sayısı - Doğruluk değişimi grafiği.....	72
Şekil 3.23: Yığın Boyutu - Doğruluk değişimi grafiği.....	73

KISALTMALAR LİSTESİ

ADL:	Activity of Daily Life (Günlük Yaşam Aktiviteleri)
ANN:	Artificial Neural Networks (Yapay Sinir Ağları)
BKM:	Body Kinematics Monitoring (Vücut Kinematiği İzleme)
BLE:	Bluetooth Low Energy (Bluetooth Düşük Enerji)
CNN:	Convolutional Neural Networks (Konvolüsyonel sinir ağı)
DCNN:	Deep Convolutional Neural Networks (Derin Konvolüsyonel sinir ağı)
EMG:	Elektromiyografi
HAR:	Human Activity Recognition (İnsan Aktivite Tanıma)
HMM:	Hidden Markov Models (Gizli Markov Modelleri)
K-NN:	K-Nearest Neighbors (K-En Yakın Komşular)
LSTM:	Long-Short Term Memory (Uzun-Kısa Süreli Bellek)
RMSProp:	Root Mean Square Propagation (Kök Ortalaması Karesi Yayılımı)
RNN:	Recurrent Neural Networks (Tekrarlayan Sinir Ağları)
SMM:	Spine Motion Monitor (Omurga Hareket Monitörü)
SVM:	Support Vector Machine (Destek Vektör Makinesi)

GİRİŞ

Günümüzde birçok insan, özellikle de orta yaşlı ve yaşlı kişiler, eklem ağrılarından şikayet etmektedirler. Bu durumun en önemli örneklerinden biri de özellikle bel ağrılarıdır. Bel ağrısının birçok nedeni olabilir. Örneğin; fiziksel zorlamalar, kaza sırasında yaralanmalar, elektronik cihazların önünde çok fazla zaman harcamak, günlük aktiviteler sırasında uzun ve yanıltıcı duruşlar gibi. Omurga üzerine uygulanan kuvvet, sırt ağrılarının ana nedenlerinden biridir. Doğru omurga pozisyonunu korumak, insanların sırt ağrısını azaltmasına yardımcı olabilir.

Bel ağrısının tedavisinde çeşitli faktörler etkilidir. Bel ağrısı tedavisi yöntemlerini inceleyen birçok araştırmacı ve klinisyen, hastaların fiziksel aktiviteleri sırasında gözlemlenmesinin en önemli tekniklerden biri olduğu üzerinde durmuştur. Ancak, hastalar laboratuvar ortamında rahatça hareket edemeyeceğinden, laboratuvar ortamında doğru şekilde gözlem yapmak genellikle zordur. Bir yandan tek bir seansta omurga duruşunun süresi ve fonksiyonel aktivitelerin hareket sıklığı hakkında kesin bilgi vermek de mümkün değildir. Tüm bu kısıtlamalar göz önünde bulundurularak hastanın hareket verileri, pelvise ve omurgaya yerleştirilen; omurganın konumunu ve hareketini izlemek için San Diego Üniversitesi'nde yapılan bir çalışmada (Paladugu, vd., 2016) oluşturulan iki kablosuz vücut sensörü kullanılarak toplanmıştır.

Bu çalışma, San Diego Üniversitesi'ndeki çalışmanın devamı ve tamamlayıcısı niteliğindedir ve kronik bel ağrısına sahip veya potansiyel bel ağrısı olabilecek hastaların pelvisine ve omurgasına bağlı kablosuz vücut sensörlerinden elde edilen verileri koordine etmeyi amaçlamaktadır. Sensörler sayesinde elde edilen bu ölçümler, deneklerin günlük aktiviteleri sırasındaki duruşlarını anlamada yardımcı olacaktır. Bu sayede hastanın hangi pozisyonlara dikkat etmesi gerektiği ve fizyoterapistlere ise bel ağrısı çeken hastalar için hangi tedaviyi temel alacağı konusunda daha fazla bilgi sağlamaya yardımcı olması amaçlanmaktadır.

Vücut kinematiği izleme verilerinin analitiği için kullanılacak veriler, sensörlerin 9 eksenli hareket işlemcilerinden gelir. Bu veri, Bluetooth Düşük Enerji (BLE) üzerinden kablosuz olarak 100 Hz örnekleme frekansında bir Android cihaza iletilir. BKM (Vücut Kinematiği İzleme) sisteminde; kişilerin ekolojik ortamdaki hareket kinematiği verileri izlenmeden önce, bel omurgası duruşunun bilinen bir açı

aralığında engellenmeden 3D hareketi ölçülerek, omurga hareketlerinin büyüklüğü ve sıklığı laboratuvarında doğrulanmıştır. Bu yöntemle laboratuvardaki ölçümleri doğrulanan sensörler, kronik bel ağrısı olan hastaların günlük pozisyonlarını ve duruş verilerini toplamak için kullanılmıştır.

Çalışmamızdaki amacımız, vücut sensörlerinden gelen veriler sayesinde pozisyonları ve omurga duruşlarını belirlemek için makine öğrenme yöntemleri ve derin öğrenme metodları ile analiz etmektir. Kullanılan derin öğrenme metodlarından LSTM yöntemi ve denetimli (supervised) makine öğrenmesi metodlarından Naive Bayes, Decision Tree, Random Forest, AdaBoost, GradientBoosting ve k-NN metodlarıyla eğitmek ve sonuçları analiz ederek karşılaştırmaktır.

Veri setlerinde kullanılan parametreler; Sıra Numarası (Sequence Number), Sensöre ait MAC kimliği (MAC ID of sensor), Zaman Damgası (Time Stamp), Konum, X eksenindeki kuaterniyon değeri (Quaternion X), Y eksenindeki kuaterniyon değeri (Quaternion Y) Z eksenindeki kuaterniyon değeri (Quaternion Z), vektörün dönme açısının skalar kısmı (Quaternion W) olarak belirlenmiştir.

Analiz sonuçlarına göre, kişinin hareketleri sınıflandırılacak, hareket modelleri oluşturulacaktır. Tüm bu sonuçlar göz önüne alınarak gelecek çalışma olarak da duruş bozukluğuna bağlı olan olası bel ağrısı tahminini ve bu sayede önlem alınmasını sağlamaktır. Bu çalışmanın amacı vücuda yapıştırılmış sensörler aracılığıyla toplanan vücut kinematiği izleme verilerinin analiz edilmesi, makine öğrenmesi ve derin öğrenme ile tahmin modelleri geliştirmektir. Derin öğrenme modelinde Long Short-Term Memory metodu; makine öğrenme modellerinden ise denetimli modellerden olan k-NN, Naive Bayes, Decision Tree, Random Forest, AdaBoost ve GradientBoosting metodları test edilmiştir. Modellerin test sonuçları ve kayıp değerleri analiz edilerek karşılaştırılacaktır. Bu sayede en iyi model veya modellerin hangisi olabileceği belirlenecektir. Ayrıca, birden fazla veri seti kullanarak analiz etmek, ağrı koşullarını belirlemek ve ağrıya neden olabilecek olası pozisyonları tahmin etmek amaçlanmaktadır.

BİRİNCİ BÖLÜM

1.1. LİTERATÜR TARAMASI

1.1.1 Sensörlerde Hareket Verisi Toplama

Vücut kinematiği verisi toplamak için kablosuz vücut sensörleri kullanılmıştır. Bu örneklerden biri de eş danışmanım Prof. Dr. Yusuf Öztürk ve çalışma arkadaşlarının geliştirdiği vücut kinematiğini ve kas aktivitesini incelemek için 8 diferansiyel EMG kanalı ve 9 eksenli hareket işlemcisine sahip sensörlerdir. Bu sensörler hareketliliklerini sınırlamadığı için sporcular ve askerler için sağlık konularında kullanılabilir (Su, vd., 2015). Solunum aktivitelerinin de vücut gövdesi hareketinden etkilenebildiği düşünülmektedir. Bu nedenle önceki çalışmaya benzer olarak solunum hareketi verilerini kaydeden bir çift kablosuz, giyilebilir IMU sensörü kullanılmışlardır (Gaidhani, vd., 2017).

Mobil algılama ve hesaplama uygulamaları genellikle ivmeölçerler, jiroskoplar ve manyetometreler gibi sensörlerden gelen zaman serisi girişlerini gerektirir. Bu çalışmada, önemli sorunlardan biri olan cihazdaki sensör ölçümlerindeki gürültüyü doğrudan ele alan ve özelleştirme zorluklarını birleşik bir şekilde gösteren derin bir öğrenme çerçevesi olan DeepSense önerilmiştir (Yao, vd., 2017).

Bir çalışmada, gömülü atalet sensörleriyle bele monte edilmiş bir akıllı telefon taşınırken Günlük Yaşam Aktiviteleri (ADL) yapan 30 kişinin kayıtlarından oluşturulan bir Aktivite Tanıma veritabanı tanımlanmıştır. Veri setinde çok sınıflı bir Destek Vektör Makinesinden (SVM) yararlanılarak elde edilen sonuçlar kabul edilmiştir (Anguita, vd., 2013).

Kamiar Aminian Bijan Najafi'nin çalışmasında tasarladığı sensörler, verileri taşınabilir bir veri kaydedici ile birlikte ve basit biyomekanik modeller kullanarak kaydetmiştir. Sensörler, dış mekan ve uzun vadeli hareketlerin yakalanmasına izin verdiği için standart hareket yakalama sistemlerindeki bazı sınırlamaların da üstesinden gelebilmektedir. Bu çalışma, günlük fiziksel aktiviteleri ve yürüyüş analizlerini göstermekle beraber yaşlılarda düşme riski, yaşam kalitesi, ortopedik sonuç ve spor performansı ile ilgili çeşitli klinik araştırmaları da ortaya koymaktadır. Bu çalışmada kullanılan sensörler, standart teknolojiye uygun olmayan ortamlar için özellikle avantajlıdır (Aminian, 2004).

Diğer arařtırmacıların üzerinde çalıştığı hareket ölçüm sistemi ise hareket verilerini, piyasada mevcut olan sistemlere benzer bir video tabanlı veri toplama stratejisini kullanarak elde etmişlerdir (Roy B. Davis, 1991). Bazı arařtırmacılar ise sagittal düzlemde yürüme kinematiğini elde etmek için genel amaçlı bir sistem geliřtirmiştir. Bu çalışmada elde edilen tüm sinyallerin işleme işlemi için Matlab'te yapılmış ve aynı zamanda ham verilerdeki gürültüyü gidermek için altıncı dereceli bir Butterworth low-pass filtresi (3Hz kesme frekansı ile) kullanmıştır. Çalışmada kullanılan sensörlerden elde edilen sonuçlar ile hareket yakalama sistemi (MOCAP/Vicon) ölçümleri arasındaki karşılaştırma, değerlerin kare farklarının ortalamasının kökü hesaplanarak yapılmıştır (Ruth E ve Mayagoitia, 2002).

Diğer bir çalışmada, ayak ve ayak bileđi hareketi verilerini benzer şekilde elde etmek için beş kameralı bir hareket analiz sistemi kullanılmış, Aynı şekilde, yüksek frekanslı gürültü ve benzer klinik durumlar için (6 Hz Butterworth low pass) filtreye ihtiyaç duyulmuştur. Ayrıca bu çalışmada, bizim çalışmamıza benzer olarak klinik uygulamaların doğruluđunu ve kolaylığını korumak için göreceli ayak ve ayak bileđi segmentinin oryantasyonunu tanımlayan Euler yöntemi kullanılmıştır. (Kidder, vd., 1996)

Bazı spor bilimciler, her bir performansın çeşitli aşamalarını görüntülemiş ve atletizmdeki performansları kaydetmek için temel biyomekanik yöntemleri kullanmışlardır. Bu çalışmada kullanılan veriler 1984 Yaz Olimpiyat Oyunları sırasında toplanmış ve daha sonra veriler üzerinde, veri azaltma ve analiz işlemleri yapılmıştır (Mayagoitia ve RE, 2002). Yine aynı arařtırmacı, insan yürüyüşünün sađlığın en önemli bir göstergesi olduğunu düşünen ve yürüyüş analizinin çok çeşitli diyabet, nörolojik hastalıklar ve düşme tespit ve tahmininde uygulanabileceğini iddia etmişlerdir. Bu amaçla tasarlanan Kinect sensörleri vücudun her yerinden gelen verileri toplayarak, arařtırmacıların yürüyüş analizi yapmalarını sağlamıştır. Böylece, sanal iskelet verileri öğrenilmiş bir modele girdi olarak kullanarak, yürüme analizi için yeni bir yöntem sunup yürüyüş özellikleri kümesinin dođru ve sađlam ölçümlerini göstermişlerdir.

Diđer tarafta yine yürüyüş analizi için uzun süre boyunca yürüyüşe müdahale etmeden ve hiçbir kesinti olmadan veri toplayan kablosuz sensörler tasarlanmıştır. Yürüyüş analizinde kullanılacak bu veriler, bir ayakkabının üzerine takılacak kablosuz sensörler sayesinde gerçekleştirilmektedir (Bamberg, vd., 2008).

Duruşun belirlenmesi ve faaliyetlerinin izlenmesi, enerji harcamasının doğru bir şekilde tahmin edilmesini sağladığı için şişmanlığın önlenmesine ve tedavisine yardımcı olabileceği düşünülmektedir. Vücuda yerleştirilen birden fazla sensör günlük kullanım için çok rahatsız edici olabileceği için bu makalede, yeni bir giyilebilir sensör sunulmuştur. Bu sensör, minimum hızlanma gerektirir. Özellik çıkarımı gerektirmez ve aynı zamanda genel aktivitelerin kesin olarak tanınmasını sağlayan giyilebilir bir sensördür. Ayakkabı sensörü, 9 erişkinde oturma ve ayakta durma performanslarını test ederken, yürüme, koşma, durma / inme ve bisiklet sürme aktiviteleri de test etmektedir. Sınıflandırma için destek vektör makineleri (SVM'ler) kullanılmıştır (Sazonov, vd., 2010).

Ayrıca vücut aktivitelerinin izlenmesi, yaşlıların normal davranış biçimindeki değişiklikleri değerlendirmek için çevre yaşam ortamlarında önemli bir rol oynamaktadır. Bu amaçla, etkinlik izleme için kompozit bir eylem tanımı ve tespit modeli sunan model geliştirilmiştir. Bu model, bir basit eylemden diğerine geçişleri saptayarak ve bu eylemlerin türlerini belirleyerek bileşik eylemlerin gerçek zamanlı sürekli izlenmesini sağlamaktadır. Veri toplama için yerleşik bir üç eksenli ivmeölçer ile giyilebilir bir TI Chronos saat kullanır ve basit eylemlerin sınıflandırılması için Naive Bayes sınıflandırıcı kullanılmıştır; yürü, otur ve yat (Uslu, Altun, ve Baydere, 2011). Buradaki çalışmada kullanılan Naive Bayes sınıflandırıcısı bizim çalışmamızda da test edilmiş ve diğer makine öğrenmesi ve derin öğrenme metotlarına göre doğruluk sonuçları karşılaştırılmıştır.

1.1.2. Sensörlerde Hareket Verisi Analizi

Mobil ve giyilebilir sensör tabanlı insan etkinliği tanıma için derin öğrenme yöntemlerinin derinlemesine özetlerini sağlayan bir derleme çalışması olarak hazırlanmıştır. Çalışmaları yalnızca üretken, ayırt edici ve hibrit yöntemlerle kategorize etmekle kalmayan bu çalışmada, aynı zamanda önemli avantajlar da vurgulanmıştır (Nweke, vd., 2018).

Derin öğrenmenin gelişmesiyle birlikte ortaya çıkan Human Activity Recognition (HAR) problemlerini ele almak için yeni fikirler ortaya çıkmıştır. Bunlardan biri de çift yönlü artık (artan) Long Short-Term Memory (LSTM) hücrelerini kullanan derin bir ağ mimarisi önerilmesidir. Yönerilen bu yöntemin avantajları, iki yönlü bir bağlantının pozitif zaman yönünü (ileri durum) ve negatif zaman yönünü (geri

durum) birleştirilebilmesidir (Zhao, vd., 2018).

Uzun Kısa Süreli Belleğe (LSTM) sahip tekrarlayan sinir ağlarının (RNN'ler) otomatik olarak özellik gösterimlerini öğrenebileceğini ve uzun vadeli geçici bağımlılıkları modelleyebileceğini göz önüne alarak, iskelet tabanlı eylem tanıma için uçtan uca tamamen bağlı derin LSTM ağı önerilen çalışma, üç insan eylemi tanıma veri setindeki deneysel sonuçlar, önerilen modelin etkinliğini tutarlı bir şekilde göstermiştir (W. Zhu, vd., 2016).

Farklı bir çalışmada, insan etkinliği tanıma için bir akıllı telefon, atalet sensörleri tabanlı bir yaklaşım sunulmuş ve önerilen yaklaşım, tipik çoklu sınıf Destek Vektör Makinesi (SVM) ve Yapay Sinir Ağı (ANN) gibi geleneksel ifade tanıma yaklaşımlarıyla karşılaştırılmıştır (Hassan, vd., 2018). Derin öğrenme yaklaşımına (denetlenen veya yarı denetlenen) ve yanıt süresine (çevrimdışı veya çevrimiçi) uygun olarak iki seviyeli bir taksonomi önerilen çalışmalar da vardır. Bu çalışmada, ana konular ve zorlukların her biri için ana çözümler tartışılmaktadır (Lara ve Labrador, 2012).

Verimli ve etkili HAR gerçekleştirmek için derin bir evrimsel sinir ağı (convnet) önerilen bir diğer çalışmada, özellikle daha önce sınıflandırmanın çok zor olduğu algılananlara çok benzeyen aktiviteler konusunda da neredeyse mükemmel bir sınıflandırma elde edilmiştir (Ronaldo ve Cho, 2016). Farklı bir yazıda ise HAR problemi için sistematik bir özellik öğrenme yöntemi önerilmiştir. Bu yöntem, ham girdilerden özellik öğrenmesini sistematik bir şekilde otomatikleştirmek için derin bir evrişik sinir ağını (CNN) benimseyerek çalışır (Yang, vd., 2015). CNN'nin iyi çalıştığını gösteren diğer çalışma, CNN metodunun herhangi bir özellik çıkarım yöntemi olmadan ortalama % 93.8'lik bir kesinliğe ulaşabileceğini göstermektedir (Chen ve Xue, 2015). Buna benzer olarak başka bir çalışmada da, evrimsel sinir ağı (convnet)'in insan aktivite tanımadaki en iyi sonucu elde eden SVM'deki tüm diğer son teknoloji teknikleri geride bıraktığını göstermiştir (Ronaldo ve Cho 2015).

İnsan fiziksel aktivitesinin vücuttaki ivmeölçerler kullanılarak nasıl sınıflandırılacağı, bu amaçla kullanılan hesaplama algoritmalarına büyük önem verilerek tartışıldığı özellikle, Gizli Markov Modellerine (HMM) dayalı sınıflandırıcılara daha çok önem verildiği çalışmada bir ivmeölçer zaman serisinin veri setini analiz ederek gösterilmekte ve tartışılmaktadır (Mannini ve Sabatini, 2010). Diğer çalışmada, etkinlik tanıma için ayırt edici özellikleri otomatik olarak çıkarmak için konvolüsyonel sinir ağlarına (CNN) dayanan bir yöntem geliştirilmiş

ve CNN tabanlı yeni yaklaşımlarının pratik olduğu ve mevcut durumdan daha yüksek doğruluk elde ettiği gösterilmiştir (Zen, vd., 2014).

Üç eksenli ivmeölçer ve derin öğrenme paradigmasını kullanarak insan faaliyetlerini tanıma problemiyle uğraşan diğer araştırmacılar makalelerinde, derin aktivite tanıma modellerinin (a) insan faaliyetlerinin daha iyi tanıma doğruluğunu sağladığını, (b) mevcut sistemlerde el yapımı özelliklerin pahalı tasarımından kaçınıldığını ve (c) denetimsiz özellik çıkarımı için büyük ve etiketlenmemiş hızlanma örneklerinden yararlandığını göstermektedir. Bu üç eksenli ivmeölçerler sayesinde, en gelişmiş insan aktivitesi tanıma yöntemleri üzerinden gerçek dünya veri setlerinde önemli bir tanıma gelişimi olduğunu göstermişlerdir (Alsheikh, vd., 2016). Düşük güçlü giyilebilir cihazlar için doğru ve gerçek zamanlı sınıflandırma sağlamak için derin bir öğrenme metodolojisine dayanan bir insan etkinliği tanıma tekniği tasarlanan diğer çalışmada, özellik üretme parametrelerinin sistematik bir analizi ve mobil cihazlarda ve sensor düğümlerinde etkinlik tanıma hesaplama sürelerinin karşılaştırılması da sunulmuştur (Ravi, vd., 2016b).

Giyilebilir sensörlerden alınan verilerin analizi ve sınıflandırılması için hibrit sınıflandırıcıları kullanarak gerçek zamanlı aktivite izlemesi için bir eylem tanıma ve tespit mekanizması sunan bir çalışma yapılmıştır. Bu çalışmada öncelikle basit ve bileşik işlemlerin tespiti için tek sensör için tekli sınıflandırıcı model (SSSC), çoklu sensörler için gelişmiş Multi-Sensor Multi Classifier (MSMC) modeli sunulmuştur. Daha sonra model, gerçek zamanlı izleme amacıyla çoklu sensörler ve sınıflandırıcılar ile geliştirilmiştir. Sınıflandırıcı olarak Naive Bayes yöntemi; makine öğrenmesi metotlarından ise Hidden Markov Model (HMM) ve Susan corner detector (SCD) yöntemleri kullanılmıştır (Uslu, vd., 2013).

Diğer bir çalışmada ise giyilebilir sensörler ile yakalanan hareket verilerini içeren üç temsili veri setinde derin, evrimsel ve tekrarlayan yaklaşımları araştırılmıştır. Tekrarlayan yaklaşımların nasıl eğitileceği, yeni bir düzenleme yaklaşımının nasıl tanıtılacağı ve en son teknolojiye sahip bir veri setinde en gelişmiş teknolojinin nasıl daha iyi bir performans gösterdiklerini açıklanmıştır. Bu amaçla birden çok tekrarlayan birim katman içeren LSTM'ler kullanılmıştır (Hammerla, Halloran, ve Plötz, 2016). Bizim çalışmamızda bu çalışmaya benzer olarak çok katmanlı LSTM modeli uygulanacak ve bu katman sayısının modelin öğrenmesindeki katkısı karşılaştırılacaktır.

Aktivite tanıma için derin öğrenme yaklaşımlarının kullanılmayı amaçlayan araştırmacılar, çalışmalarında, konvolüsyonlu katman sayısı ve çekirdek boyutu gibi çeşitli önemli hiper parametrelerinin CNN'nin performansı üzerindeki etkisi izlenmiştir (Zebin, Scully, ve Ozanyan, 2016). Bizim çalışmamızda LSTM yöntemi için hiper parametrelerin değişiminin performans üzerindeki etkisi izlenmiştir. Derin Konvolüsyonel Sinir Ağlarının (DCNN) etkinlik tanıma görevi için etkinlik görüntüsünden en uygun özellikleri otomatik olarak öğrenmesini sağlayan çalışma, etkinlik tanıma doğruluğu ve hesaplama maliyeti açısından en gelişmiş teknolojilerden daha iyi performans göstermektedir (Jiang ve Yin, 2015). Diğer bir çalışmada gerçek zamanlı bir sürekli etkinlik izleyebilen Real Time Activity Monitoring (RAM) adında bir sistem geliştirilmiştir. RAM, Support Vector Machines (SVM) yöntemiyle, çoklu sınıf sınıflandırmasını gerçekleştirmekte ve SVM ve En Yakın Komşu (KNN) sınıflandırıcılarının önceden tanımlanmış özelliklerle beslediği 8 farklı konfigürasyonla karşılaştırmaktadır (Uslu ve Baydere 2015). Son zamanlarda, derin öğrenmenin son ilerlemesi, otomatik yüksek seviyeli özellik çıkarımını gerçekleştirmeyi mümkün kılarak, birçok alanda ümit verici bir performans sergilemektedir. Mevcut literatürü üç açıdan özetleyen, sensör yöntemi, derin model ve uygulamaları, bu çalışma aynı zamanda derin öğrenmeye dayalı sensör temelli aktivite tanıma alanındaki son gelişmeleri de araştırmaktadır (Wang, vd., 2019).

Derin bir öğrenme metodolojisi sunan bir diğer makalede sunulan birleştirilmiş yöntem ise, düğüm içi hesaplamanın gerekli olduğu tipik bir derin öğrenme çerçevesinde mevcut olan bazı sınırlamaların üstesinden gelmeyi amaçlamaktadır (Ravi, vd., 2016a).

Bir araştırmada, aktivite tanıma yapmak için evrişimli ve LSTM tekrarlayan birimlerine dayalı genel bir derin çerçeve önerilmiş olup, bunlar: (i) çok modelli giyilebilir sensörler için uygundur; (ii) doğal olarak sensör füzyonu yapabilirler; (iii) özelliklerin tasarlanmasında uzman bilgisi gerektirmez; ve (iv) özellik aktivasyonlarının zamansal dinamiklerini açıkça modelleyen bir çerçevedir (Ordóñez ve Roggen 2016). Geleneksel LSTM'den farklı olarak, bilgileri yeni bir havuz katmanı ile birden fazla LSTM arasında paylaşılan bir yöntem geliştirilmiştir. Bu havuz katman, komşu yörüngelere karşılık gelen LSTM'lerin gizli temsilini birleştirerek çalışır. Metodun performansı birkaç kamuya açık veri setinde gösterilmiştir. Bu model önceki tahmin yöntemlerinden % 42 daha iyi

performans gösterildiği bulunmuştur (Alahi, vd., 2016).

Daha güçlü bir ağaç yapısı tabanlı geçiş yöntemi öneren araştırmacılar, 3 boyutlu iskelet verilerindeki gürültüyü ve tıkanıklığı ele almak için, sıralı giriş verilerinin güvenilirliğini öğrenmek ve buna bağlı olarak bellek hücresinde depolanan uzun vadeli bağlam bilgisinin güncellenmesi üzerindeki etkisini ayarlamak için LSTM içinde yeni geçiş mekanizması sunmuşlardır. Sunulan metot, insanlar için 3D hareket analizinde zorlu veri setlerinde bile son teknoloji performansa ulaşmaktadır (Liu, vd., 2016).

Bir diğer çalışmada ise bizim çalışmamızda pelvis ve omurgaya yerleştirilen kablosuz iki vücut sensörüne benzer olarak, sırasıyla bir ayağa ve bir bele bağlanan iki giyilebilir atalet sensöründen elde edilen verileri birleştirerek bir insanın günlük aktivitelerini değerlendirmesini önerilmiştir. İlk olarak, bu iki sensörden gelen veriler, aktivitenin türünü belirlemek amacıyla genel olarak üç sınıfa ayrılmıştır: sıfır yer değiştirme aktivitesi, geçiş aktivitesi ve güçlü yer değiştirme aktivitesi. İkincisi, etkinlikleri daha da ayırt etmek için sezgisel ayrımcılığa veya gizli Markov modellerine (HMM) dayanan sınıflandırma modülü uygulanmıştır. Giyilebilir sensör sistemi kullanarak deneyler yapılmış ve elde edilen sonuçlar algoritmanın etkinliğini ve doğruluğunu kanıtlamıştır. (C. Zhu ve Sheng, 2009) Değişken uzunluktaki giriş dizilerinde uzun vadeli bağımlılıkları yakalayabilen tanıma modelleri için derinlemesine tekrarlayan sinir ağlarının (DRNN'ler) kullanılmasını öneren bir çalışma da vardır. Bu çalışmada Uzun kısa süreli bellek (LSTM) DRNN'lerine dayanan tek yönlü, çift yönlü ve basamaklı mimarileri sunmuş ve çeşitli veri kümeleri üzerindeki etkilerini değerlendirilmiştir. Deneysel sonuçlar, önerilen modellerimizin destek vektör makinesi (SVM) ve k-en yakın komşular (KNN) gibi geleneksel makine öğrenimini kullanan yöntemleri geride bıraktığını göstermiştir (Murad ve Pyun, 2017). Çalışmamız, LSTM sinir ağı eğitilerek oluşturulmuş derin öğrenmesi metoduyla klasik makine öğrenmesi metotlarını kıyaslamamız yönüyle bu çalışmaya benzemektedir.

Bu çalışmanın temel amacı ve literatür katkısı BKM (Body Kinematic Monitoring) sistemi tarafından ölçülen 3 boyutlu omurga duruşu ve hareket verilerini kullanılabilir hale getirmek, veri üzerinde gerekli dönüşümleri yapmak, gürültülü veriyi filtreleyip, filtrelenen veri ile yeni bir makine öğrenmesi yöntemi olan derin öğrenme metodunu uygulamaktır. Tüm işlemler sonucunda kişinin omurga duruşu ve pozisyonları analiz edilerek uzmanların olası bel ağrılarına önlem alabilmeleri

amaçlanmaktadır. Gövdeye bitişik olan bu sensör ağından yapılan ölçümler, bir mekanik omurga test platformu kullanılarak doğrulanmıştır. Bu doğrulanan ölçümlere göre, sensörden alınan yanlış ve gürültülü veriler silinecek, gerekli dönüşümler yapılacak, farklı pozisyonlara ayrılacak ve pozisyonların 3 eksenli grafikleri yorumlanarak duruş ve pozisyon hakkında bilgi vermek amaçlanmaktadır.

Araştırmamızda, R programlama dili, sensörden gelen konum verilerinin analizi, dönüştürülmesi, çizilmesi, kümelenmesi vb. işlemler için kullanılacaktır. Analizin sonuçlarına göre, hareket ve duruşu tahmin etmek için makine öğrenmesi ve derin öğrenme yöntemlerini R programlama dili ile birleştirmek planlanmaktadır. Derin öğrenme ve makine öğrenmesi metotları Python programlama diline ait tensor-flow, numpy, seaborn ve pandas açık kaynaklı kütüphaneleri kullanılarak denenmiş ve sonuçlar raporlanmıştır.

İKİNCİ BÖLÜM

2.1 VÜCUT KİNEMATİĞİ VERİLERİNİN TOPLANMASI

Bu çalışmadaki amacımız bel ağrısına sahip veya bel ağrısına sahip olabilecek potansiyel bel kişilerden toplanmış vücut kinematığı verilerinin analitiğini yapmaktır. Kişilerden elde edilen verilerin laboratuvar ortamında değil; kişinin kendi doğal yaşamında doğal hareketleriyle elde edilen veriler olmasına özen gösterilmiştir. Tasarlanan sensörlerin doğrulanması amacıyla öncelikle laboratuvar ortamında veriler kaydedilmiştir. Sensörler, San Diego Üniversitesinde Prof. Dr. Yusuf Öztürk hocanın danışmanlığı yaptığı bir yüksek lisans tez çalışmasında geliştirilmiştir. Veriler, insan vücuduna yerleştirilmiş 3 cm çaplı iki farklı sensörden gelmektedir. Sensörlerde bulunan 9 eksenli hareket işlemcisi tarafından toplanan veriler, bir Bluetooth Low Energy (BLE) protokolünün kullanıldığı kablosuz bir arayüz vasıtasıyla, 100 Hz'lik bir örnekleme frekansında bir android cihaza kablosuz olarak iletilmiştir. Android cihaz tarafından kaydedilen 9 boyutlu hareket verileri, araştırmacı veya klinisyenin uygun şekilde kullanması için yerel olarak depolanır. Ek olarak, sensör düğümleri, yüksek bant genişliği gerektiren uygulamalar için Enhanced Shock Burst protokolünü uygulamaktadırlar (Paladugu, vd., 2016).

Bir sensör pelvis bölgesinde diğer sensör ise omurganın ortasında yer almaktadır. Sensörler insan derisine direkt olarak yapıştırılmıştır. İnsan vücuduna yerleştirilen kablosuz bu iki sensörün aynı hizada konumlandırılması çok önemlidir. Her sensör ideal olarak aynı sıra üzerinde hizalandığında, sensörlerin açılma hareketi aynı düzlemde ölçülebilecektir. Bununla birlikte, sensörler tam olarak hizalanmadığında, hatalar ortaya çıkabilmektedir. Laboratuvar verilerinde pozisyonların tanımlanması ve belirlenmesi işlemleri tamamlanmıştır. Daha sonra farklı kişilerin vücutlarına sensörler yerleştirilmiş ve günlük yaşamlarına devam etmesi istenmiştir. Günlük yaşamları sırasında kaydedilen veriler daha sonra analizler için kullanılmıştır.

Sensörlerin doğrulanması amacıyla öncelikle laboratuvar ortamında yapılan testler; bir objenin alınması, ayağa kalkıp oturmak, merdiven çıkmak, yürümek ve uzun süre oturmak gibi aktiviteleri içermektedir. Dinamik denemeler, aynı hareketin 10 kez tekrarlanmasıyla; statik denemeler ise bir hareketin 5 saniye boyunca devam etmesi şeklinde gerçekleştirilmiştir. Verileri toplamak için Spine Motion Monitor (SMM)

isimli bir Andorid uygulama geliştirilmiştir. Test işleminin sonunda, android cihaz tarafından toplanan 9 boyutlu hareket verileri işlenmiş ve sensörlerin geçerliliğini test etmek için mocap sisteminden alınan ölçümlerle karşılaştırılmıştır.

Ekolojik izleme için, uygulama tarafından toplanan veriler, hasta gün sonunda sensör düğümlerini ve android cihazı geri getirdiğinde klinisyen tarafından analiz edilebilen bir csv dosyasında saklanır.

Kaydedilen csv dosyasında tutulan bilgilerin başlıkları aşağıda verilmiştir. Bu bilgiler sırasıyla: Sıra numarası (Sequence Number), Sensörün Medya Erişim Kontrolü Kimliği (MAC ID of sensor), Zaman Damgası (Time Stamp), Göz ardı edilebilecek değer (Ignore), Pozisyon (Position), X eksenindeki kuaterniyon değeri (Quaternion X), Y eksenindeki kuaterniyon değeri (Quaternion Y), Z eksenindeki

Seq_Num	MAC_ID_of_sensor	Time_Stamp	Ignore	Position	Quat_X	Quat_Y	Quat_Z	Quat_W	Ignore
11	1	2016/05/09 08:22:29.714	0	1	0.421881	-0.025316	-0.097683	0.901018	4
59	1	2016/05/09 08:22:29.762	0	1	0.421972	-0.025345	-0.097696	0.900973	4
108	1	2016/05/09 08:22:29.811	0	1	0.422079	-0.025367	-0.097707	0.900921	4
109	1	2016/05/09 08:22:29.812	0	1	0.422200	-0.025391	-0.097720	0.900862	4
110	1	2016/05/09 08:22:29.813	0	1	0.422284	-0.025403	-0.097729	0.900822	4
111	1	2016/05/09 08:22:29.814	0	1	0.422372	-0.025426	-0.097741	0.900779	4
206	1	2016/05/09 08:22:29.909	0	1	0.422492	-0.025452	-0.097751	0.900720	4
207	1	2016/05/09 08:22:29.910	0	1	0.422635	-0.025471	-0.097758	0.900652	4
208	1	2016/05/09 08:22:29.911	0	1	0.422869	-0.025516	-0.097778	0.900539	4
209	1	2016/05/09 08:22:29.912	0	1	0.422869	-0.025516	-0.097778	0.900539	4
255	1	2016/05/09 08:22:29.957	0	1	0.422869	-0.025516	-0.097778	0.900539	4
256	1	2016/05/09 08:22:29.958	0	1	0.422869	-0.025516	-0.097778	0.900539	4

kuaterniyon değeri (Quaternion Z), Vektörün dönme açısının belirten skalar kısım (Quaternion W) Göz ardı edilebilecek değer (Ignore) şeklindedir.

Şekil 2.1: Csv formatında kaydedilen verinin işlenmemiş hali.

Şekil 2.1'de yer alan örnek veri, laboratuvar ortamında kaydedilmiş veri olup, verideki tüm bilgiler analitik yaparken işimize yaramayacağı için analiz yaparken bazı sütunlar kullanılmamıştır. 4. ve 10. Sütunlarda yer alan göz ardı edilebilecek değerler (Ignore) veriden çıkartılarak veri analizine başlanmıştır.

Veride yer alan sıra numarası (sequence) bilgisi verilerin kayıt sırasını göstermektedir. Sensörün Medya Erişim Kontrolü Kimliği (MAC ID of sensor) bilgisi ise sensörlere ait kimlik numarası olarak tanımlanmıştır. Veride, iki sensörden gelen veriler kaydolmakta olup ID numaraları 0 ve 1 olarak tanımlanmıştır. Zaman Damgası (Time Stamp) ise verinin kaydolduğu an bilgisini göstermektedir. Göz ardı edilebilecek değer (Ignore) ise sensörü tasarlanan çalışmada kullanılmıştır, bizim çalışmamızda önemli bir bilgi olmadığı için analiz işlemleri yapılmadan önce

veriden çıkartılacaktır. Pozisyon (Position) bilgisi pozisyon numaralarını tutan bilgidir. Pozisyon numaraları laboratuvar ortamında etiketlenmiştir. Quaternion X, x eksenindeki kuaterniyon değeri, Quaternion Y, y eksenindeki kuaterniyon değerini, Quaternion Z, z eksenindeki kuaterniyon değerini göstermektedir. Quaternion W ise vektörün dönme açısının belirten skalar kısmı ifade etmektedir. 10. sütunda yer alan göz ardı edilebilecek değer (Ignore), tıpkı 4. sütundaki göz ardı edilebilecek değer gibidir. Dolayısıyla bizim çalışmamızda da bu değerler göz ardı edilerek işlem yapıldı.

2.1.1 Kuaterniyonlar (Quaternions)

Kuaterniyonlar, dönüşü ve yönlendirmeyi tanımlayan basit ama en güçlü matematiksel araçlardan biridir (Kuipers, vd., 1999). Kuaterniyonlar, Hamilton tarafından geliştirilmiş olup, dört boyut olarak ele alınan üç boyuta genişletilmiş karmaşık sayılardır. Ayrıca, üç boyutlu uzayda iki vektörün veya yönlendirilmiş iki çizginin bölümü olarak tanımlanabilirler. Kuaterniyonlar aşağıdaki gibi gösterilebilirler:

$$q = w + xi + yj + zk \quad (2.1)$$

2.1 numaralı denklemde yer alan w, dönme açısını gösteren skaler kısımdır. x, y, z ise 3 boyutlu uzayın vektörel kısımlarıdır. Kuaterniyonlar tanımlanırken kullanılan i, j, k karmaşık sayıları arasında ise 2.2 numaralı formüldeki gibi bir ilişki bulunmaktadır.

$$\begin{aligned} i^2 = j^2 = k^2 = ijk = -1 \\ \text{ve} \\ ij = k = ji \end{aligned} \quad (2.2)$$

San Diego üniversitesinde yapılmış olan çalışmada hareket verileri IMU tarafından kuaterniyonlar olarak rapor edilmiştir. Daha sonra normalize edilen kuaterniyonlar, iletişim modülü vasıtasıyla ana bilgisayar uygulamasına iletilir.

Bir kuaterniyon (x, y, z, w) , geometrik olarak 2.3, 2.4, 2.5, 2.6 numaralı denklemlerle temsil edilebilir:

$$x = X \times \sin \frac{\alpha}{2} \quad (2.3)$$

$$y = Y \times \sin \frac{\alpha}{2} \quad (2.4)$$

$$z = Z \times \sin \frac{\alpha}{2} \quad (2.5)$$

$$w = Y \times \cos \frac{\alpha}{2} \quad (2.6)$$

Denklemlerde X, Y, Z, 3 boyutlu alanda dönme eksenlerinin birim uzunluklarıdır ve α , radyan cinsinden eksen etrafında dönme açısıdır. Vücut kinematiği veri analizi yaparken kullandığımız verilerin kaydedildiği bu çalışmada (Paladugu, vd., 2016), Spinal Motion Monitor (SMM) sensörlerini doğrulamak için sensörlerin açısız hareketi, üç boyutlu uzaysal (mekansal) dönüş tahmin edilerek belirlenmiştir. X eksenini, Y eksenini ve Z eksenini boyunca dönme açıları sırasıyla Roll, Pitch ve Yaw (Euler Angles) ile verilmiştir.

$$Roll \theta = \arctan \frac{2w \times x + y \times z}{1 - 2x^2 + y^2} \quad (2.7)$$

$$Pitch \theta = \arcsin 2w \times y - x \times z \quad (2.8)$$

$$Yaw \psi = \arctan \frac{2w \times z + x \times y}{1 - 2y^2 + z^2} \quad (2.9)$$

Kuaterniyonlar, dönme matrisleri, dönme eksenleri ve açıları, küresel dönme açıları ve Euler dönme açıları gibi çeşitli diğer geometrik ifadelerle dönüştürülebilir. Buradaki bir dönüş, eksenlerin dönüşü veya nesnenin dönüşü olarak düşünülebilir. Her iki dönüş için de; bir dönüş, üç eksenin etrafındaki dönmelerin bileşimi olarak tanımlanmaktadır. Bizim çalışmamızda elde edilen kuaterniyon değerleri rotation matrislere dönüştürerek kullanacağız. x, y, z, w kuaterniyonlarına karşılık gelen dönme matrisi M şekil 2.2'deki gibi tanımlanır:

$$R_{xyz} = \begin{bmatrix} 1 - 2y^2 - 2z^2 & 2xy + 2wz & 2xz - 2wy \\ 2xy - 2wz & 1 - 2x^2 - 2z^2 & 2yz + 2wx \\ 2xz + 2wy & 2yz - 2wz & 1 - 2x^2 - 2y^2 \end{bmatrix}$$

Şekil 2.2: x, y, z, w kuaterniyonlarına karşılık gelen dönme matrisi

2.2 VERİLERLE YAPILAN İŞLEMLER

Verinin düzenlenmesi, temizlenmesi, analizi işlemleri R programlama dili aracılığıyla yapılmıştır. R, istatistiksel hesaplama ve grafikler için yazılım ortamı olup aynı zamanda programlama dilidir. R Foundation tarafından desteklenen ve GNU Tasarısının parçası olan bir özgür yazılımdır. Yeni Zelanda Auckland Üniversitesinden Ross Ihaka ve Robert Gentleman tarafından ortaya çıkarılan R, hâlihazırda R Geliştirme Çekirdek Ekibi tarafından geliştirilmektedir. S programlama diline benzeyen R, S'nin uyarlaması olarak değerlendirilebilir. İstatistikî yazılım geliştirme için istatistikçiler arasında standart haline gelen R, istatistikî yazılım geliştirme ve veri analizi alanında kullanılmaktadır.

Bu çalışmamızda, R dilini oldukça kolay ve gelişmiş bir arayüz olan RStudio aracılığıyla kullandık. İnsan vücuduna yapıştırılan kablosuz iki vücut sensöründen gelen verilerin kaydedilmesi, mobil uygulama aracılığıyla gerçekleşmektedir. Kaydedilen verilen sensörlerden sırayla gelmesi beklenmektedir. Fakat gerek bulunan ortamdan kaynaklanan nedenler gerekse sensörden kaynaklanan nedenlerden dolayı veri akışı tamamiyle bu şekilde gerçekleşmemiştir. Bu nedenle veri analizi yapılırken bu duruma dikkat edilerek hesaplamalar yapılmıştır. Sensörlerden gelen veriler .csv formatında kaydedilmektedir. Ekolojik olarak kaydedilen verinin işlenmesi işleminden önce laboratuvarda toplanan doğrulama/test verisini kullanarak işleme başladık. Laboratuvar verisi daha sonra

ekolojik olarak toplanan verilerin anlamlandırılmasında kullanılacaktır.

Öncelikle laboratuvar verisi RStudio'dan okunur ve gerekli sütun başlıkları atamaları yapılır. Veride kullanmayacağımız (göz ardı edilebilecek) bilgileri bulunmaktaydı, bu sütunlar veriden çıkarılarak devam edilir. Veri iki parça halinde incelenecektir.

Seq_Num	MAC_ID_of_sensor	Time_Stamp	Ignore	Position
11	1	2016/05/09 08:22:29.714	0	1
59	1	2016/05/09 08:22:29.762	0	1
108	1	2016/05/09 08:22:29.811	0	1
109	1	2016/05/09 08:22:29.812	0	1
110	1	2016/05/09 08:22:29.813	0	1
111	1	2016/05/09 08:22:29.814	0	1
206	1	2016/05/09 08:22:29.909	0	1
207	1	2016/05/09 08:22:29.910	0	1
208	1	2016/05/09 08:22:29.911	0	1
209	1	2016/05/09 08:22:29.912	0	1
255	1	2016/05/09 08:22:29.957	0	1
256	1	2016/05/09 08:22:29.958	0	1

Şekil 2.3: Veriden bir kesit (Seq_num, MAC_ID_of_sensor, Time_Stamp, Ignore ve Position)

Şekil 2.3'de bir kesiti gösterilen veri, Seq_num, MAC_ID_of_sensor, Time_Stamp, Ignore ve Position değerlerinden oluşurken; Şekil 2.4'te kuaterniyon değerleri (Quat_X, Quat_Y, Quat_Z, Quat_W) tutulur.

Quat_X	Quat_Y	Quat_Z	Quat_W
0.421881	-0.025316	-0.097683	0.901018
0.421972	-0.025345	-0.097696	0.900973
0.422079	-0.025367	-0.097707	0.900921
0.422200	-0.025391	-0.097720	0.900862
0.422284	-0.025403	-0.097729	0.900822
0.422372	-0.025426	-0.097741	0.900779
0.422492	-0.025452	-0.097751	0.900720
0.422635	-0.025471	-0.097758	0.900652
0.422869	-0.025516	-0.097778	0.900539
0.422869	-0.025516	-0.097778	0.900539
0.422869	-0.025516	-0.097778	0.900539
0.422869	-0.025516	-0.097778	0.900539

Şekil 2.4: Verinin kuarterniyon değerlerden oluşan diğer kısmı (Quat_X, Quat_Y, Quat_Z, Quat_W)

2.2.1 Verilerin Temizlenmesi-Gruplara Ayrılması

Gerek android mobil uygulamadan kaynaklanan hatalar gerekse çevresel etmenler verinin bozulmasına ve gürültü verilerin kaydolmasına neden olmaktadır. Gürültü, eksik ve yanlış veri gelmesi şeklinde olabileceği gibi aynı anda gelen tamamıyla aynı veri şeklinde de olabilir. Verileri iki gruba ayırdıktan sonra, yapacağımız ilk iş üst üste gelen tamamıyla aynı satırların veriden çıkardık. Bu işlemi de yaptıktan sonra Pozisyon numaralarının tutulduğu “pos_file” listesi veride bulunan Position değişkenindeki sadece tekil (benzersiz, eşsiz) değerler olarak oluşturuldu. Pozisyonlar genel olarak 3 grup altında toplanmıştır: otururken, ayakta ve fonksiyonel aktiviteler sırasında kaydedilmiş veriler. Pozisyonlara ait tablolar 2.1, 2.2, 2.3, 2.4’deki gibidir.

Tablo 2.1: Genel Pozisyon Bilgileri

Pozisyon Numarası	Pozisyon
1	Yok
2	İdeal Pozisyon
3	Ekolojik Pozisyon
4	Yürümek (Dışarıda)
5	Oturmak (Dışarıda)
6	Ayakta durmak (Dışarıda)

Tablo 2.2: Ayaktayken kaydedilen pozisyon bilgileri

Ayaktayken	
Pozisyon Numarası	Pozisyon
7	fbend (tekrar eden)
8	extend (tekrar eden)
9	Rtlb (tekrar eden)
10	Ltlb (tekrar eden)
11	Rrot (tekrar eden)
12	Lrot (tekrar eden)
13	Fbend (durađan)
14	extend (durađan)
15	Rtlb (tekrar eden)
16	Ltlb (tekrar eden)
17	Rrot (tekrar eden)
18	Lrot (tekrar eden)
19	Stand_prol

Tablo 2.3: Otururken kaydedilen pozisyon bilgileri

Otururken	
Pozisyon Numarası	Pozisyon
20	sit_fben_rep (tekrar eden)
21	Sit_Ltlb_rep (tekrar eden)
22	Sit_Rtlb_rep (tekrar eden)
23	Sit_Rrot_rep (tekrar eden)
24	Sit_Lrot_rep (tekrar eden)
25	Sit_prol
26	Sit_Slump_static (durađan)
27	Sit_RShift_rep (durađan)
28	Sit_LShift_rep (durađan)

Tablo 2.4: Fonksiyonel aktiviteler sırasında kaydedilen pozisyon bilgileri

Fonksiyonel Aktiviteler	
Pozisyon Numarası	Pozisyon
29	STS_rep (tekrar eden)
30	Stairs_rep (tekrar eden)
31	Pickup_rep (tekrar eden)
32	Walking1
33	Walking2
34	Walking3
35	Walking4
36	Walking5

Verideki kayıtlar, her iki sensörden sırasıyla geldiği için sensör verilerinin eşleşmesi için öncelikle veriyi iki gruba ayırmamız gerekti. Sensör 0 ve sensor 1'den gelen verileri grupladıktan sonra, pos_file listesinden analiz yapmak istenilen pozisyonu seçtik. Veriyi anlamlı gruplara ayırdıktan sonra matematiksel işlemlere başlamadan önce R'da gerekli paketler yüklendi.

“matlib” paketi : Lineer cebir ve çok değişkenli istatistiği öğretmek ve öğrenmek için matris fonksiyonlarını içeren bir pakettir. Verideki kuaterniyonları rotasyon matrisine çevirdikten sonra matris üzerinde yaptığımız tüm işlemler için bu paketi kullandık.

“oro.nifti” paketi : İstatiksel hesaplamalarda kullanılan bu paket Giriş / çıkış fonksiyonları ve görselleştirilmesi ANALYZE, NIFTI veya AFNI formatlarını takip eden tıbbi görüntüleme verileriyle işlem yapmamızı sağladı.

“raster” paketi : Mekansal verilerin okunması, yazılması, manipüle edilmesi, analiz edilmesi ve modellenmesinde kullanılan pakettir. Raster paketi, temel ve üst düzey işlevleri uygulayabilir ve çok büyük dosyaların işlenmesinde kullanılabilir.

“Thermimage” paketi : Termal görüntü ve video dosyalarının girilmesi, ikili ham verilerin çizilmesi ve sıcaklık tahminlerine dönüştürmesi için kullanılan fonksiyonlar ve rutinleri içeren pakettir.

“vmsbase” paketi : Analizlerin farklı adımlarındaki verileri görsel olarak inceleyerek raporlar ve bilimsel yayınlar için etkili çıktılar üretmek için Viewer Tools ile donatılmış bir pakettir. Bizim çalışmamızda radyan değerlerini dereceye çevirirken kullandık.

“Directional” paketi: Yönlü veri analizi için çeşitli fonksiyonları barındıran bir paket- tir. Hipotez testleri, diskriminant ve regresyon analizi işlemleri de bu paket sayesinde gerçekleştirilebilmektedir. Biz çalışmamızda, bu paketi rotasyon matrislerini Euler açılarına çevirmek için kullandık.

“signal” paketi : Aslen 'Matlab' ve 'Octave' için yazılmış bir dizi sinyal işleme fonksiyonlarını içeren bir pakettir. Filtre oluşturmak için yardımcı programları, filtreleme işlevlerini, yeniden örnekleme rutinlerini ve filtre modellerinin görselleştirmesini içerir. Ayrıca enterpolasyon işlevlerini de bu paket sayesinde gerçek- leştirmek mümkündür. Biz bu paketi, hesapladığımız Euler açılarını medyan filtresi uygulamak için kullandık.

Gerekli paketlerin kurulumu tamamlandıktan sonra ilk olarak iki gruba ayırdığımız sensör 0 ve sensör 1 verilerindeki kuaterniyon değerleri için ayrı bir matris oluşturduk. Kuaterniyonlar farklı türlere dönüştürülebilir matematik araçları olduğu için biz bu çalışmamızda, kuaterniyon değerlerini rotasyon matrisine çevirerek kullanacağız.

2.2.2 Sensör Verilerinin Eşleştirilmesi

Çalışmamızın amacı sensörlerin birbirine göre hareket durumlarını izlemek olduğu için iki farklı gruba ayırdığımız verilerin eşleştirme işlemine başlayacağız. Normal şartlarda verilerin her iki sensörden de sırayla gelmesini bekliyoruz fakat gerek ortamdan kaynaklanan nedenler (ses, frekans bozacak etmenler, sensörler arasına giren ve bağlantıyı koparabilecek her şey) gerekse sensörden kaynaklanan sorunlar nedeniyle verilerin her zaman sırayla gelmediğini gördük. Bu nedenle sırayla ardışık gelmeyen bu verileri eşleştirmek için gelen verilerin sıra numaralarını (Seq_Num) dikkate aldık. Referans sensör olarak sensör 0'ı ele aldık ve sensör 0'dan gelen ilk verinin sıra numarasına baktık. Eşleştirmek istediğimiz sensör 1'den gelen verinin sıra numarası sensör 0'dan gelen verinin sıra numarasından 1 numara fazla ise eşleşme gerçekleşecek. Aynı şekilde sıra numaraları arasındaki fark 1'den 5'e kadar kontrol edilecek. Bu aralıkta birbirine uyan sıra numarası varsa eşleşme gerçekleşecek ve eşleşen veriler yeni bir dosyaya aktarılacak.

Sensörlerin eşleştirildiği algoritma, Algoritma 1'deki gibidir:

Algorithm 1 Sensör Verilerinin Eşleştirilmesi

```
1: for i=1:length (data) do
2:   a ← data[i]
3:   if a ∉ data[i] then
4:     for j = 1:5 do
5:       b ← a - j
6:       if a ∈ data[i] then
7:         break
8:       else
9:         b ← a + j
10:        if b ∈ data[i] then
11:          break
12:        end
13:      end
14:    end
15:  end
16: end
```

Kuaterniyonlar, dönme matrisleri, dönme eksenleri ve açıları, küresel dönme açıları ve Euler dönme açıları gibi çeşitli diğer geometrik ifadelerle dönüştürülebildiği için Algoritma 1’de sözde kodunu verdiğiniz sensör eşleşmesi işlemi sırasında, eşleşen her kayıt için kuaterniyon değerleri rotasyon matrisine çevrildi.

2.2.3 Rotasyon Matrisinin Hesaplanması

Algoritma 1’de sözde kodu verilen sensör verilerini eşleştirme işleminden sonra rotasyon matrisini Euler açılarına çevirmek için matris üzerinde üzerinde birkaç matris işlemleri uygulamamız gerekiyordu. Bunların ilki; rotasyon matrisinin simetrisini almak ayna görüntüsü oluşturmaktı. Diğer işlem ise satırların sırasını değiştirmek. 1-2-3 sıralamasına sahip matrisin satır sırası 3-2-1 şeklinde yeniden düzenlendi. Üçüncü işlem olarak; matrisin bazı elemanlarının negatif olması gerektiği için değiştirilecek elemanlar -1 ile çarpıldı. Rotasyon matrisi ise 3x3 boyutlu matrislerdir. Euler açıları hesaplarken kullanılacak olan fonksiyonu ise 4x4 boyutlu matrislerle işlem yapacağı için, rotasyon matrisine yeni satır ve sütunlar eklendi. Eklenen satır ve sütun değerleri “0” değerine sahipti. Sadece matris köşegeni üzerinde yer alan 4. satır 4. sütun elemanına “1” değeri atandı. Matris işlemleri tamamlanmış oldu.

Sensör eşleştirme algoritmasının içinde yer alan ve döngü devam ettikçe eşleşen her kayıt için rotasyon matris hesaplamasını, gerekli matris işlemlerinin yapan algoritmanın sözde kodu Algoritma 2 gibidir.

Algoritma 2: Rotasyon matris hesaplanması ve matris üzerinde yapılan işlemler

- 1: rotMatrix ← quaternion2rotation(quat values)
 - 2: mirror (rotMatrix)
 - 3: change row order of rotMatrix as (3, 2, 1)
 - 4: add a row to rotMatrix as (0,0,0)
 - 5: add a column to rotMatrix as (0,0,0,1)
-

Algoritma 2, rotasyon matris hesaplanması ve matris üzerinde yapılan işlemleri içermektedir.

2.2.4 Rotasyon Matrislerinin Euler Açılara Dönüştürülmesi

Matris üzerinde yapılacak tüm işlemler tamamlandıktan sonra Euler açılarının hesaplaması gerçekleştirildi. Directional paketinde bulunan “rot2eul” fonksiyonu kullanılarak dönüşüm gerçekleştirildi “rot2eul” fonksiyonu trigonometrik fonksiyonları

kullanarak bu işlemi gerçekleştirmektedir. Fonksiyon, açıları sonuçları radyan cinsinden hesapladığı için radyanı dereceye çevirecek bir fonksiyon kullanıldı.

Algoritma 3: Euler Açılarının Hesaplanması

- 1: eulangles \leftarrow rot2eul (rotMatrix)
 - 2: as matrix eulangles
 - 3: rad2deg (eulangles)
 - 4: determine eulangles dimension as 1x3
-

Algoritma 3’de Euler açılarını hesaplanması ve derece olarak gösterilmesi işlemlerini içeren sözde kod verilmiştir.

Çalışmamızda, iki sensörün birbirine göre hareketleri incelendiği için sensör verileri eşleştirildikten sonra eşleşen verilerin birbirine göre değerlerinin hesaplanması gerekmektedir. Bu hesaplama açıların farkını alınarak yapıldı. Hesaplanan farklar yeni dosyaya aktarıldı. Hesaplanan açı farkı değerlerinde çok fazla gürültülü vardı. Bu yüzden filtre uygulandı. Filtrelenmiş veri ile filtrelenmemiş gürültülü veriler arasındaki farkı görebilmek için iki şekilde de ayrı ayrı grafikleri çizilerek görselleştirildi.

2.3 VERİLERİN GÖRSELLEŞTİRİLMESİ

2.3.1 Gürültülü Verinin Görselleştirilmesi

Sensörlerden gelen ham verilerde herhangi bir filtre ve değiştirme işlemi yapılmadan öncelikli gürültülü verilerin grafikleri çizildi. Grafik çizimleri, R’da gelişmiş görselleştirme metodlarını içeren “ggplot2” paketi kurulumu gerçekleştirildi.

“ggplot2” paketi: İstatistiksel programlama dili olan R için bir veri görselleştirme paketidir. Verileri görselleştirirken grafikleri, ölçekler ve katmanlar gibi anlamsal bileşenlere ayırır.

“ggplot2” paketinde bulunan ggplot fonksiyonu gelişmiş grafik çizimlerinde kullanılmaktadır. ggplot, tek başına çok çeşitli parametreleri içerdiği için veri görselleştirmesinde grafik çizimleri için bu fonksiyon yeterli olmaktadır.

Görselleştirilecek gürültülü veri örneği:

time_sequence	x	y	z
0.000	-0.03406948	-0.004366691	-0.2484668
0.031	-0.01902127	-0.023599294	-0.2606156
0.032	-0.10121055	-0.032048747	-0.2823400
0.033	-0.08868251	-0.046886852	-0.3582892
0.569	-0.07970046	-0.060601904	-0.4397794
0.570	0.13056226	0.025901649	-1.9789516
1.110	1.61585676	2.799107353	2.9223574
1.111	1.59757645	3.012891667	2.7737299
1.112	1.59793880	3.002360115	2.8062186
1.112	1.60003542	2.996169452	2.7949674
1.139	1.60953040	2.982539728	2.5562005
1.260	1.61319787	2.979369812	2.6076028

Şekil 2.5: Görselleştirilen ham (gürültülü) veri örneği

Pozisyon verileri genel olarak 3 başlık altında toplanabilir. Bunlar; Otururken ölçülmüş veriler, ayaktaiken ölçülmüş veriler ve fonksiyonel aktiviteler şeklindedir. Bu başlıklar dışında bağımsız olarak ölçülen pozisyonlardan da hata almadan kaydedebildiğimiz iki farklı pozisyon bulunmaktadır. (İdeal pozisyon (Ecological Monitoring) ve dış ortamda yapılan yürüyüş (Walking_outdoor)).

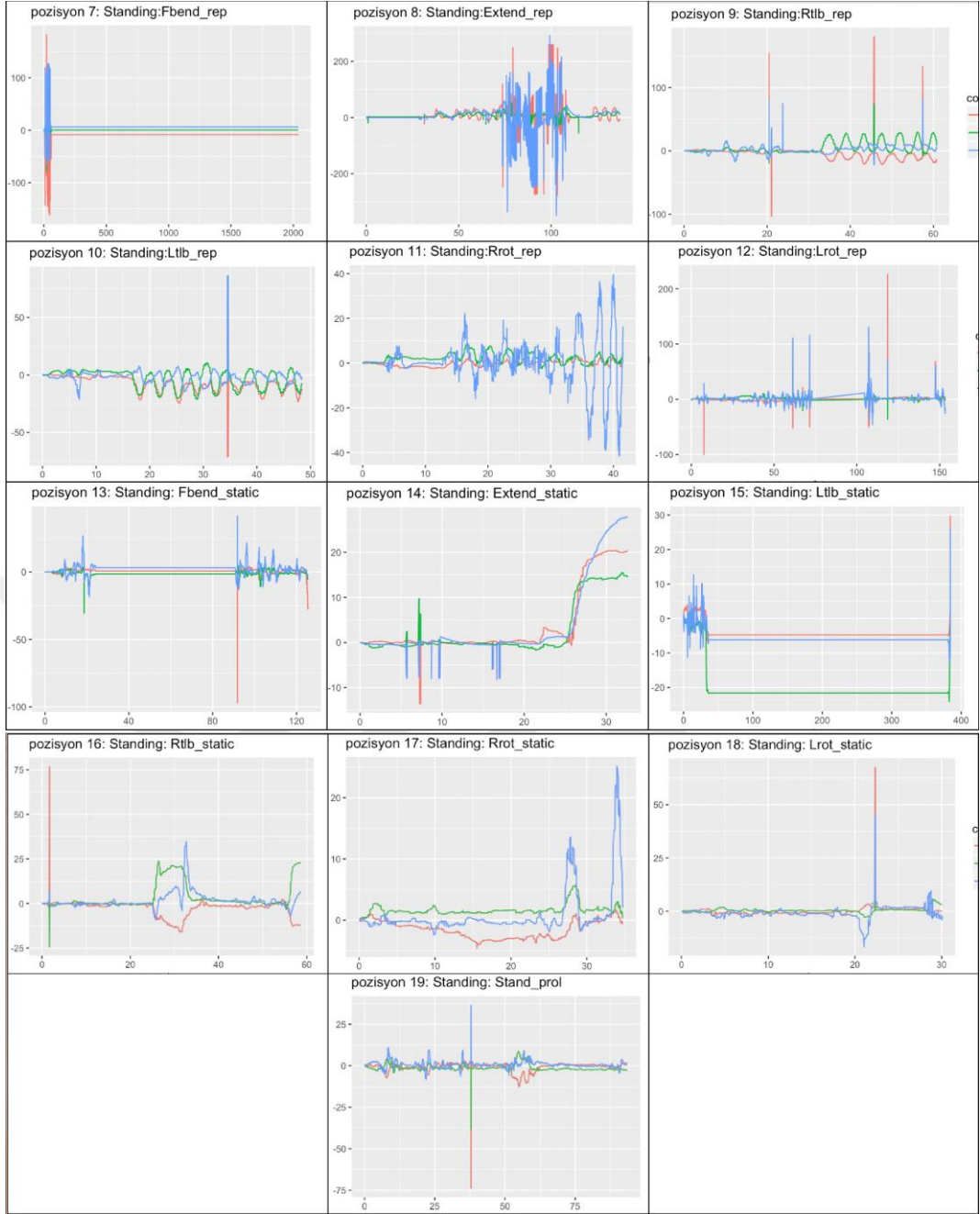
Ayaktayken ölçülen pozisyonlar 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 numaralı pozisyonlardır. Bu pozisyonlar; 7 numaralı “Fbend_rep” pozisyonu, 8 numaralı “Extend_rep” pozisyonu, 9 numaralı “Rtlb_rep” pozisyonu, 10 numaralı “Ltlb_rep” pozisyonu, 11 numaralı “Rrot_rep” pozisyonu, 12 numaralı “Lrot_rep” pozisyonu, 13 numaralı “Fbend_static” pozisyonu, 14 numaralı “Extend_static” pozisyonu, 15 numaralı “Ltlb_static” pozisyonu, 16 numaralı “Rtlb_static” pozisyonu, 17 numaralı “Rrot_static” pozisyonu, 18 numaralı “Lrot_static” pozisyonu, 19 numaralı “Stand_prol” pozisyonudur.

Tablo 2.5: Ayaktayken Kaydedilen Pozisyonlara Ait Yaklaşık Süre Bilgileri

Ayaktayken		
Pozisyon	Süresi	Kayıt Sayısı
Fbend_rep	33 dakika 58 saniye	2911
Extend_rep	2 dakika 27 saniye	5335
Rtlb_rep	1 dakika 1 saniye	2750
Ltlb_rep	49 saniye	2273
Rrot_rep	41 saniye	1881
Lrot_rep	2 dakika 34 saniye	1578
Fbend_static	2 dakika 5 saniye	2901
Extend_static	33 saniye	1543
Ltlb_static	6 dakika 25 saniye	1903
Rtlb_static	59 saniye	2954
Rrot_static	34 saniye	1620
Lrot_static	30 saniye	1452
Stand_prol	1 dakika 34 saniye	4204

Ayaktayken ölçülmüş olan pozisyonlardan; 7 numaralı “Fbend_rep” pozisyonu, 8 numaralı “Extend_rep” pozisyonu, 9 numaralı “Rtlb_rep” pozisyonu, 10 numaralı “Ltlb_rep” pozisyonu, 11 numaralı “Rrot_rep” pozisyonu, 12 numaralı “Lrot_rep” pozisyonu tekrarlı olarak uygulanmıştır.

Ayaktayken ölçülmüş, işlenmemiş, ham (gürültülü) pozisyonların, x y z eksenlerindeki Euler açılarına ait grafikleri Şekil 2.6'deki gibidir.



Şekil 2.6: Ayaktayken kaydedilmiş (gürültülü) pozisyonların grafiği

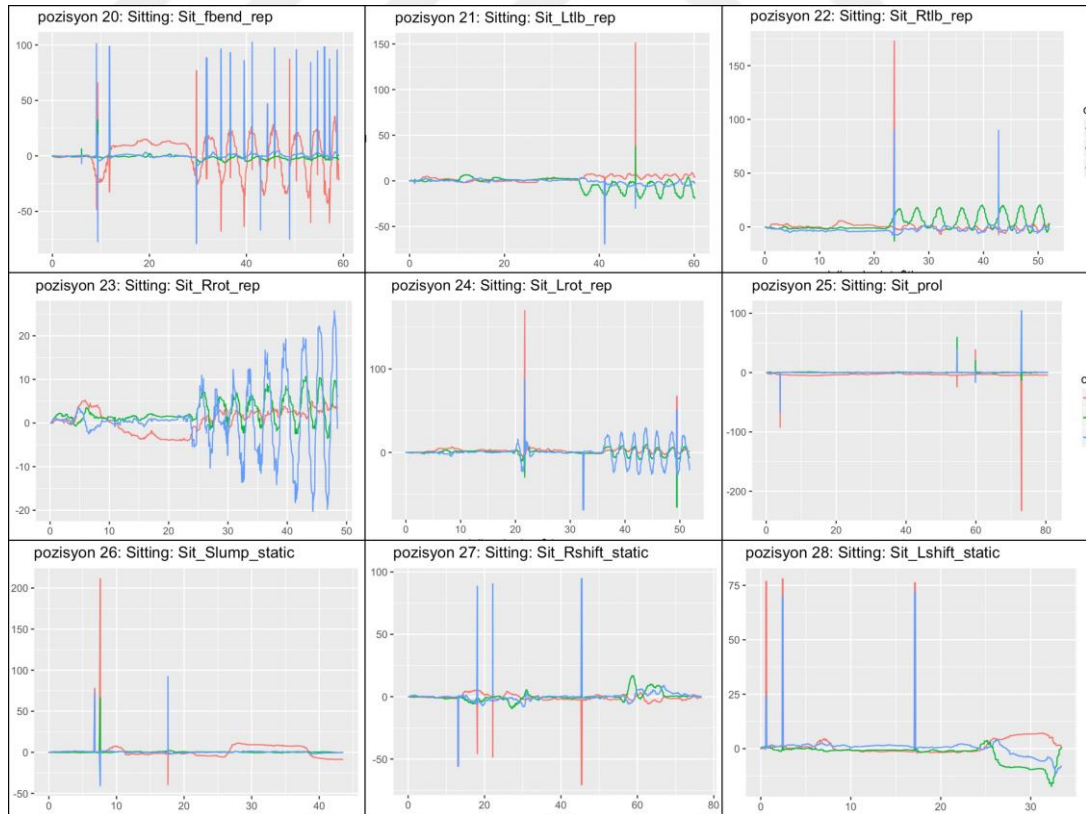
Şekil 2.6’de çizilen grafikler filtrelenmemiş çizimlerdir. Daha sonra filtrelenen pozisyonlar gösterilmiştir.

Otururken ölçülen pozisyonlar 20, 21, 22, 23, 24, 25, 26, 27, 28 numaralı pozisyonlardır. Bu pozisyonlar; 20 numaralı “Sit_fbend_rep” pozisyonu, 21 numaralı “Sit_Ltlb_rep” pozisyonu, 22 numaralı “Sit_Rtlb_rep” pozisyonu, 23 numaralı “Sit_Rrot_rep” pozisyonu, 24 numaralı “Sit_Lrot_rep” pozisyonu, 25 numaralı “Sit_prol” pozisyonu”, 26 numaralı “Sit_Slump_static” pozisyonu, 27 numaralı “Sit_Rshift_static” pozisyonu ve 28 numaralı “Sit_Lshift_static” pozisyonudur.

Tablo 2.6: Otururken Kaydedilen Pozisyonlara Ait Yaklaşık Süre ve Kayıt Sayısı Bilgileri

Otururken		
Pozisyon	Süresi	Kayıt Sayısı
Sit_fbend_rep	59 saniye	2810
Sit_Ltlb_rep	1 dakika 1 saniye	2768
Sit_Rtlb_rep	52 saniye	2528
Sit_Rrot_rep	48 saniye	2329
Sit_Lrot_rep	52 saniye	2718
Sit_prol	1 dakika 20 saniye	3504
Sit_Slump_static	43 saniye	2048
Sit_RShift_rep	1 dakika 18 saniye	3559
Sit_LShift_rep	34 saniye	1635

Otururken ölçülmüş olan pozisyonlardan; 20 numaralı “Sit_fbend_rep” pozisyonu, 21 numaralı “Sit_Ltlb_rep” pozisyonu, 22 numaralı “Sit_Rtlb_rep” pozisyonu, 23 numaralı “Sit_Rrot_rep” pozisyonu, 24 numaralı “Sit_Lrot_rep” pozisyonu tekrarlı olarak ölçülmüştür. Otururken ölçülmüş, işlenmemiş, ham (gürültülü) pozisyonların, x y z eksenlerindeki Euler açılarına ait grafikleri Şekil 2.7 gibidir.



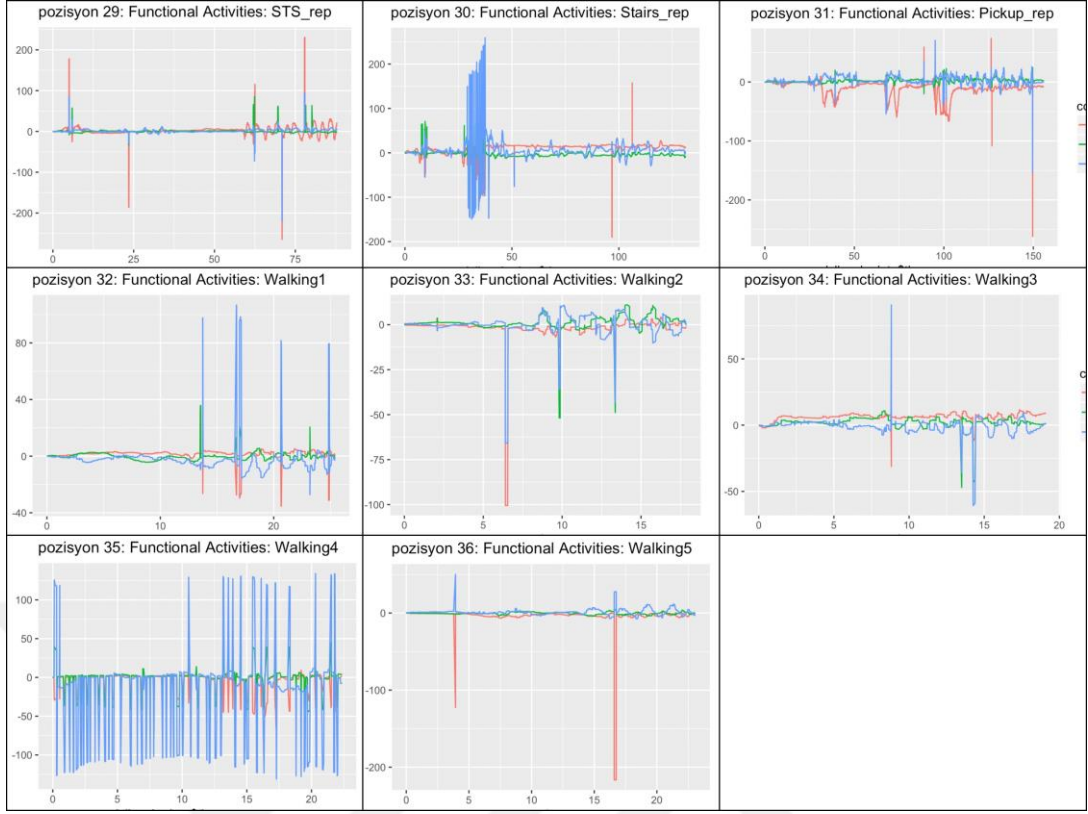
Şekil 2.7: Otururken kaydedilmiş (gürültülü) pozisyonların grafiği

Fonksiyonel aktiviteler esnasında ölçülen pozisyonlar 29, 30, 31, 31, 33, 34, 35, 36 numaralı pozisyonlardır. Bu pozisyonlar; 29 numaralı “STS_rep” pozisyonu, 30 numaralı “Stairs_rep” pozisyonu, 31 numaralı “Pickup_rep” pozisyonu, 32 numaralı “Walking1” pozisyonu, 33 numaralı “Walking2” pozisyonu, 34 numaralı “Walking3” pozisyonu, 35 numaralı “Walking4” pozisyonu ve 36 numaralı “Walking5” pozisyonudur.

Tablo 2.7: Fonksiyonel Aktiviteler Sırasında Kaydedilen Pozisyonlara Ait Yaklaşık Süre Bilgileri

Fonksiyonel Aktiviteler		
Pozisyon	Süresi	Kayıt Sayısı
STS_rep	1 dakika 27 saniye	4026
Stairs_rep	2 dakika 12 saniye	5929
Pickup_rep	2 dakika 36 saniye	7173
Walking1	26 saniye	1309
Walking2	18 saniye	827
Walking3	19 saniye	932
Walking4	22 saniye	1076
Walking5	24 saniye	1083

Fonksiyonel aktiviteler esnasında ölçülen 29 numaralı “STS_rep” pozisyonu, 30 numaralı “Stairs_rep” pozisyonu, 31 numaralı “Pickup_rep” pozisyonu tekrarlı olarak uygulanmıştır. Fonksiyonel aktiviteler sırasında ölçülmüş, işlenmemiş, ham (gürültülü) pozisyonların, x y z eksenlerindeki Euler açılarına ait grafikleri Şekil 2.8 gibidir.



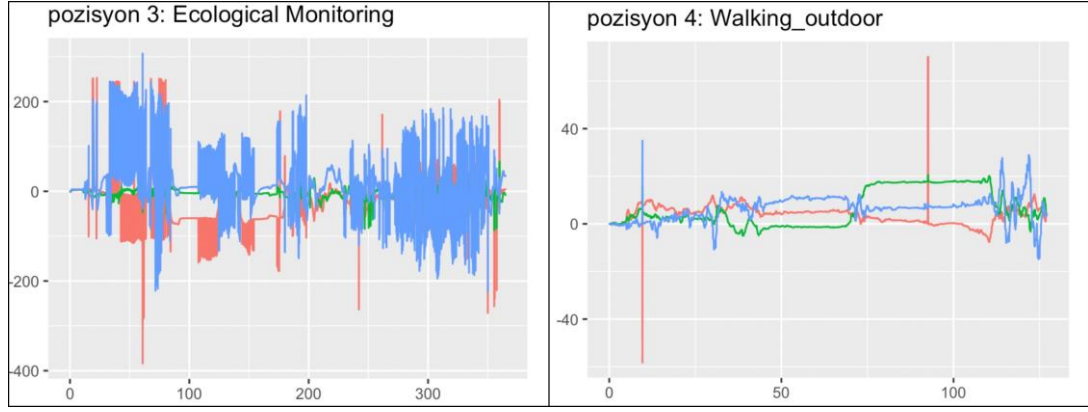
Şekil 2.8: Fonksiyonel aktiviteler sırasında kaydedilmiş (gürültülü) pozisyonların grafiği

Ayaktayken, otururken ve fonksiyonel bazı aktiviteler sırasında ölçülen verilerden farklı olarak iki pozisyon daha ölçülmüştür. Bu pozisyonlar 3 numaralı “Ecological Monitoring” pozisyonu ve 4 numaralı “Walking_outdoor” pozisyonlarıdır.

Tablo 2.8: Diğer Aktiviteler Sırasında Kaydedilen Pozisyonların Yaklaşık Süre Bilgileri

Ekolojik İzleme-Yürüyüş		
Pozisyon	Süresi	Kayıt Sayısı
Ecological Monitoring	6 dakika 6 saniye	16270
Walking_outdoor	2 dakika 9 saniye	5377

Ecological monitoring ve walking outdoor aktiviteleri sırasında ölçülmüş, işlenmemiş, ham (gürültülü) pozisyonların, x y z eksenlerindeki Euler açılarına ait grafikleri Şekil 2.9 gibidir.



Şekil 2.9: Ecological monitoring ve Walking outdoor aktiviteleri sırasında kaydedilmiş (gürültülü) pozisyonların grafiği

2.3.2 Gürültülü Verinin Filtrenmesi

Sensörlerden okunan değerlerde gürültü veya eksik veri olduğu için Sensörlerden okunan değerlerde gürültü veya eksik veri olduğu için öncelikle veri üzerinde bazı işlemler yapmamız gerekti. Bu nedenle ilk adım olarak veri setinde “NA” değer olan hücelere sahip satırlar silindi. Bir satıra ait tek bir hücrede bile “NA” değer bulunması istenmeyen durum olduğu için o satır bütünüyle silinmiş oldu.

Ham verideki gürültüye sadece boş hücreler veya “NA” değerler sebep olmamaktadır. Aynı zamanda veri setinde ardışık olarak gelen kayıtlarda aşırı büyük atlamalar görülmektedir. Bu durumu düzeltmek için veri setine “Medyan Filtresi” uygulanarak sorun çözülmüştür.

2.3.2.1 Medyan Filtre

Görüntü ve sinyal işleme konularında, gürültü temizlemek için kullanılan yöntemlerden birisidir. Amaç belirli bir pencere aralığındaki sayıların ortancasını (medyan) alarak aşırı büyük atlamaları kaldırmaktır. Medyan Filtre, genellikle görüntüden veya sinyalden paraziti gidermek için kullanılan doğrusal olmayan bir dijital filtreleme tekniği olup bu tür gürültü azaltma, daha sonraki işlemlerin sonuçlarını iyileştirmek için tipik bir ön işleme aşamasıdır.

Medyan filtresinin ana fikri, her girişi komşu girişlerin medyanı ile değiştirerek, sinyal girişi ile giriş yapmaktır. Komşu değerlerdeki paterne, giriş sırasına göre tüm sinyal boyunca kayan “pencere” denir. 1D sinyalleri için, en belirgin pencere önceki ve sonraki girişlerden sadece birkaçıdır, buna karşın görüntüler gibi 2D (veya daha yüksek boyutlu) sinyaller için daha karmaşık pencere desenleri ("kutu" veya "çapraz" desenler) mümkündür. Pencerenin tek sayıda olması durumunda, ortancayı

tanımlamak kolaydır. Bu nedenle kullandığımız fonksiyonda pencere sayısı belli bir denkleme göre tek sayı olarak hesaplandı. Değerlerin medyanı, penceredeki tüm girişler sayısal olarak sıralandıktan sonra orta değer olarak alınır. Pencere sayısı 2.10 numaralı denklemle hesaplanmaktadır:

$$k < -1 + 2 \times \min((nrow(data) - 1) \% / \%2, ceiling(0.1 \times nrow(data))) \quad (2.10)$$

2.3.3 Filtrelenmiş Verinin Görselleştirilmesi

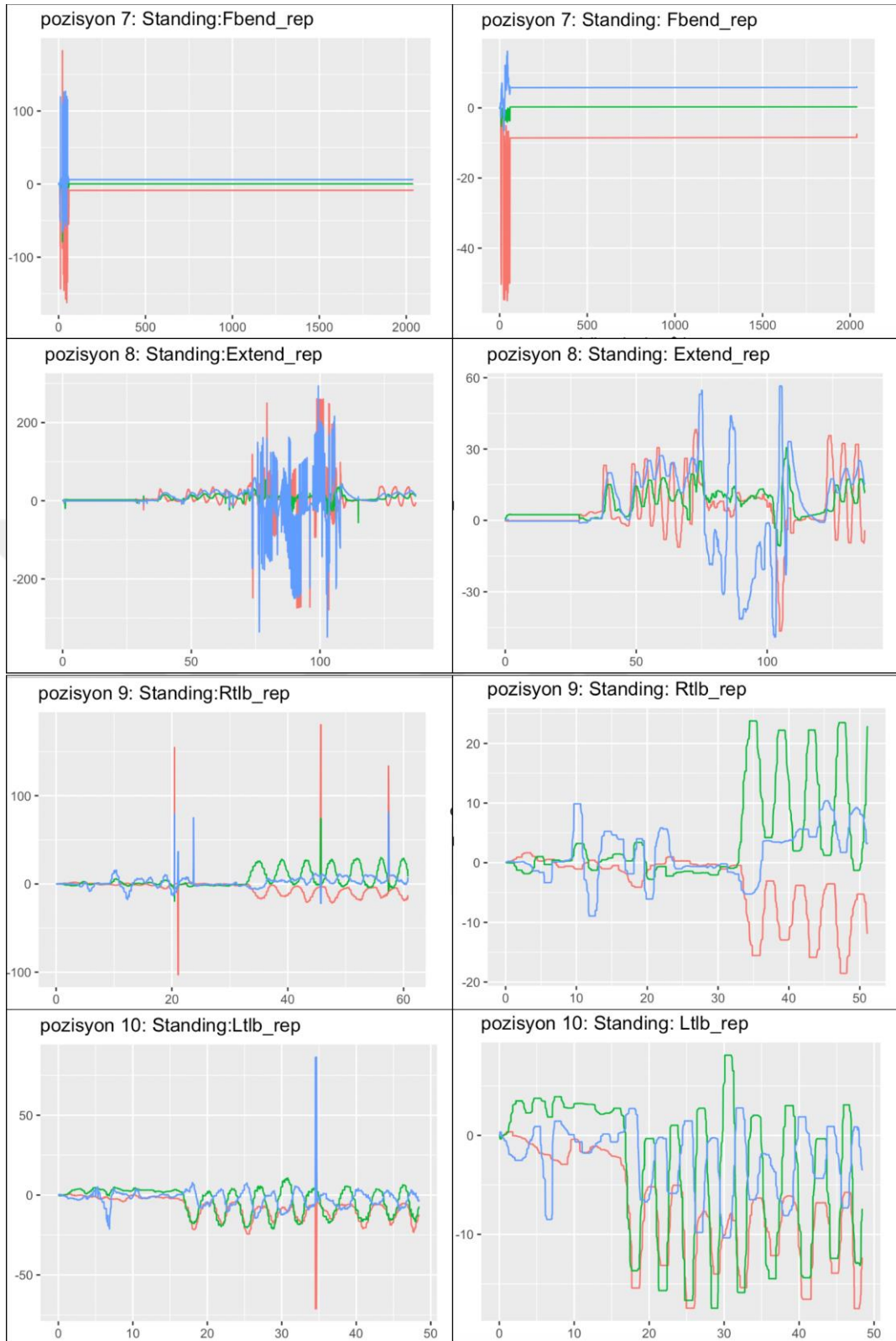
Sensörlerden gelen ham verideki gürültülerin düzeltilmesi için medyan filtre uygulandı.

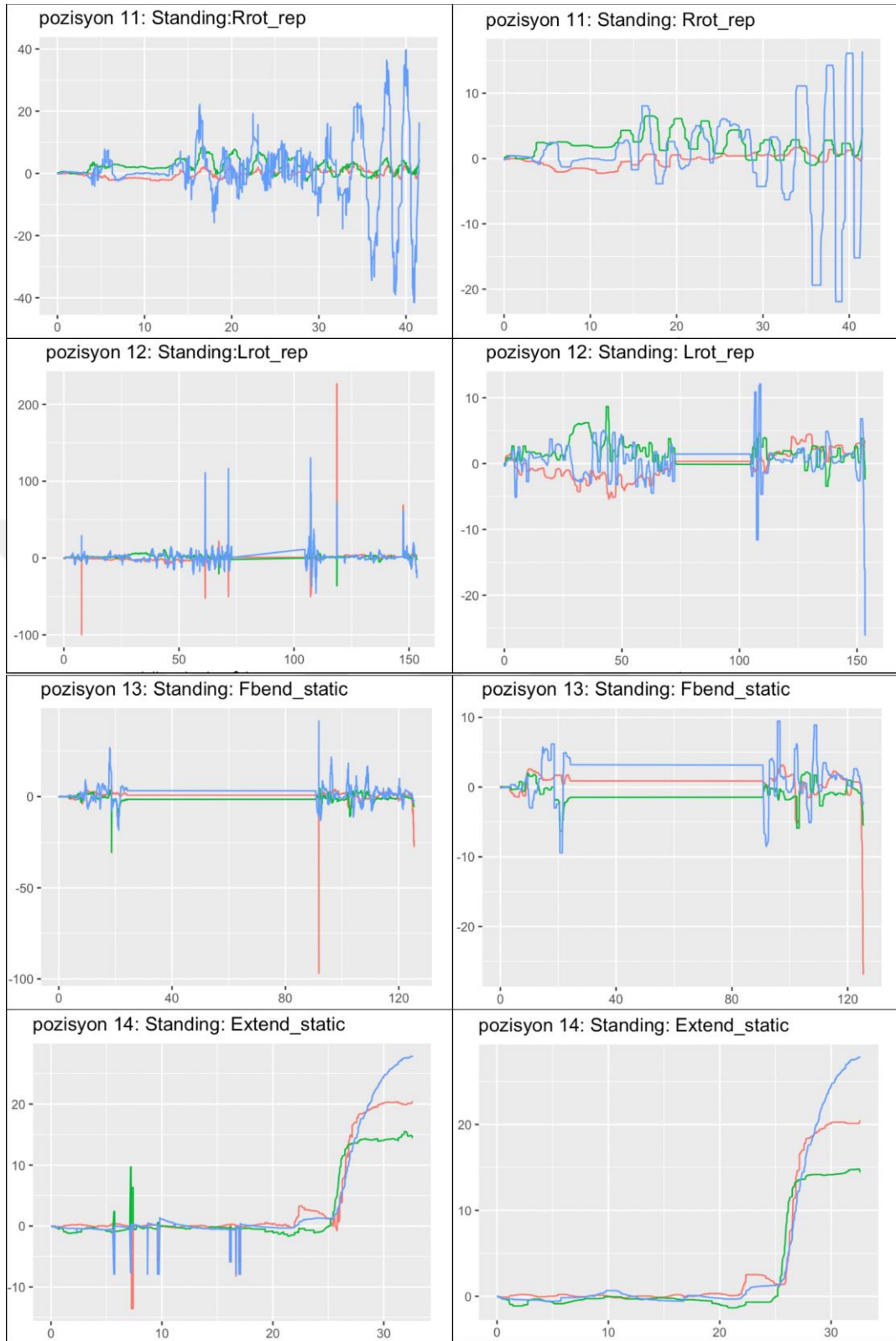
time_sequence	x	y	z
0.000	-0.01902127	-0.004366691	-0.2484668
0.031	-0.03406948	-0.023599294	-0.2606156
0.032	-0.03406948	-0.023599294	-0.2823400
0.033	-0.01902127	-0.004366691	-0.2606156
0.569	0.00000000	0.000000000	-0.2484668
0.570	0.13056226	0.025901649	0.0000000
1.110	1.59757645	2.799107353	2.5562005
1.111	1.59793880	2.864843334	2.5562005
1.112	1.60003542	2.921012277	2.5562005
1.112	1.60953040	2.949881146	2.6076028
1.139	1.61319787	2.975299386	2.6302340
1.260	1.61585676	2.979369812	2.6409769

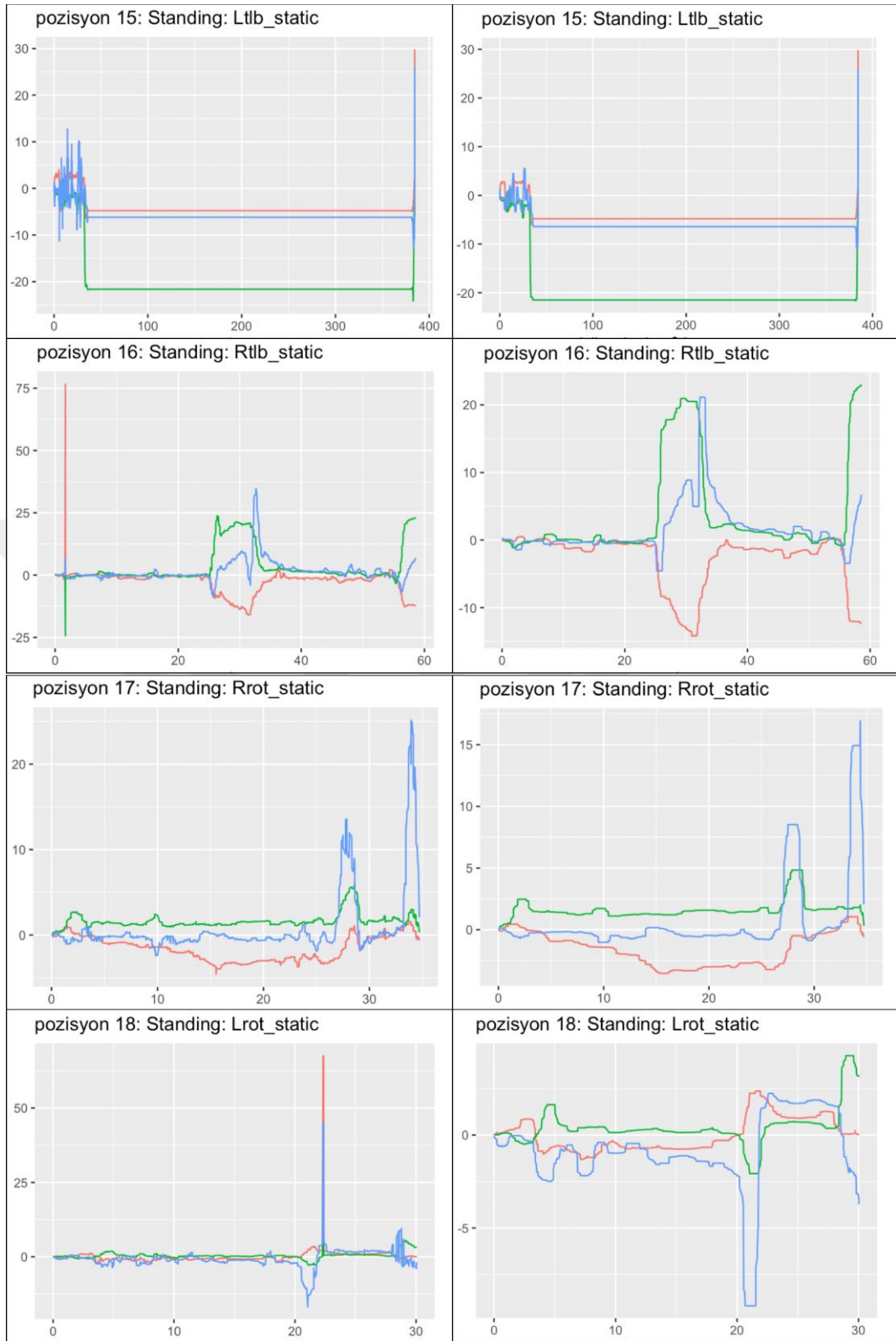
Şekil 2.10: Görselleştirilen filtrelenmiş veri örneği

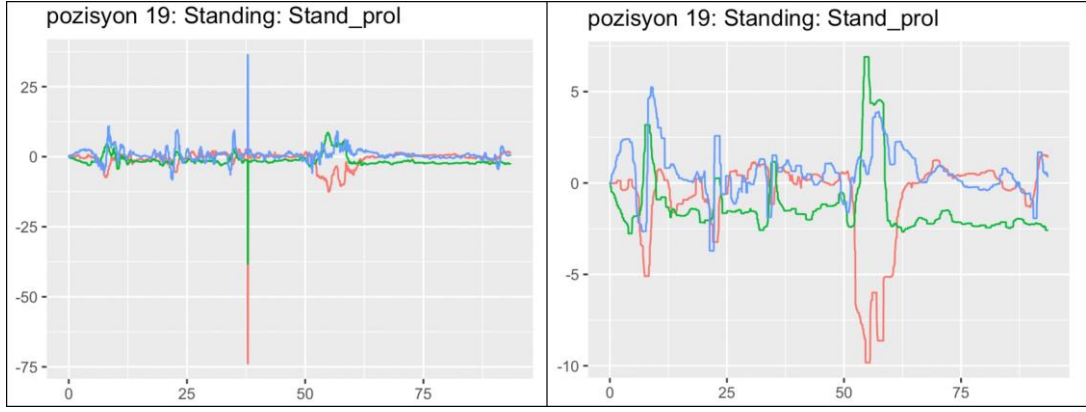
Medyan filtresinde kullanılacak pencere sayısı, 2.10 numaralı denklemde verilen formüle göre 81 olarak hesaplanmış ve uygulanmıştır. Filtre uygulandıktan sonra elde edilen veri setinde aşırı atlama olan değerler çıkartılmış ve gürültü giderilmiştir.

Ayaktayken ölçülmüş, filtrelenmiş, ham (gürültülü) pozisyonların, x,y,z eksenlerindeki Euler açılarına ait grafikler Şekil 2.11'deki gibidir. Filtrelenmemiş ve filtrelenmiş grafik görseller karşılaştırılarak medyan filtresi sonucu gösterilmiştir. Şekilde solda yer alan grafikler medyan filtresi uygulanmamış ham veriden elde edilmiş olup soldaki grafikler medyan filtre uygulanmış verilerle çizilen grafiklerdir.



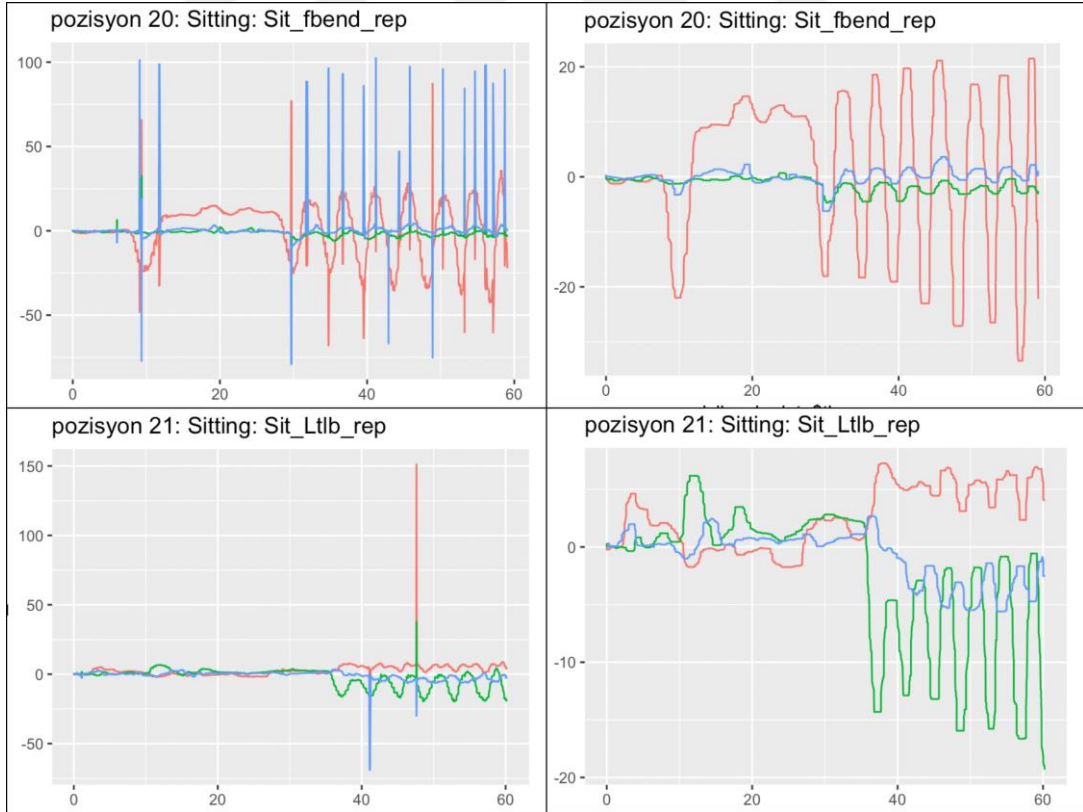


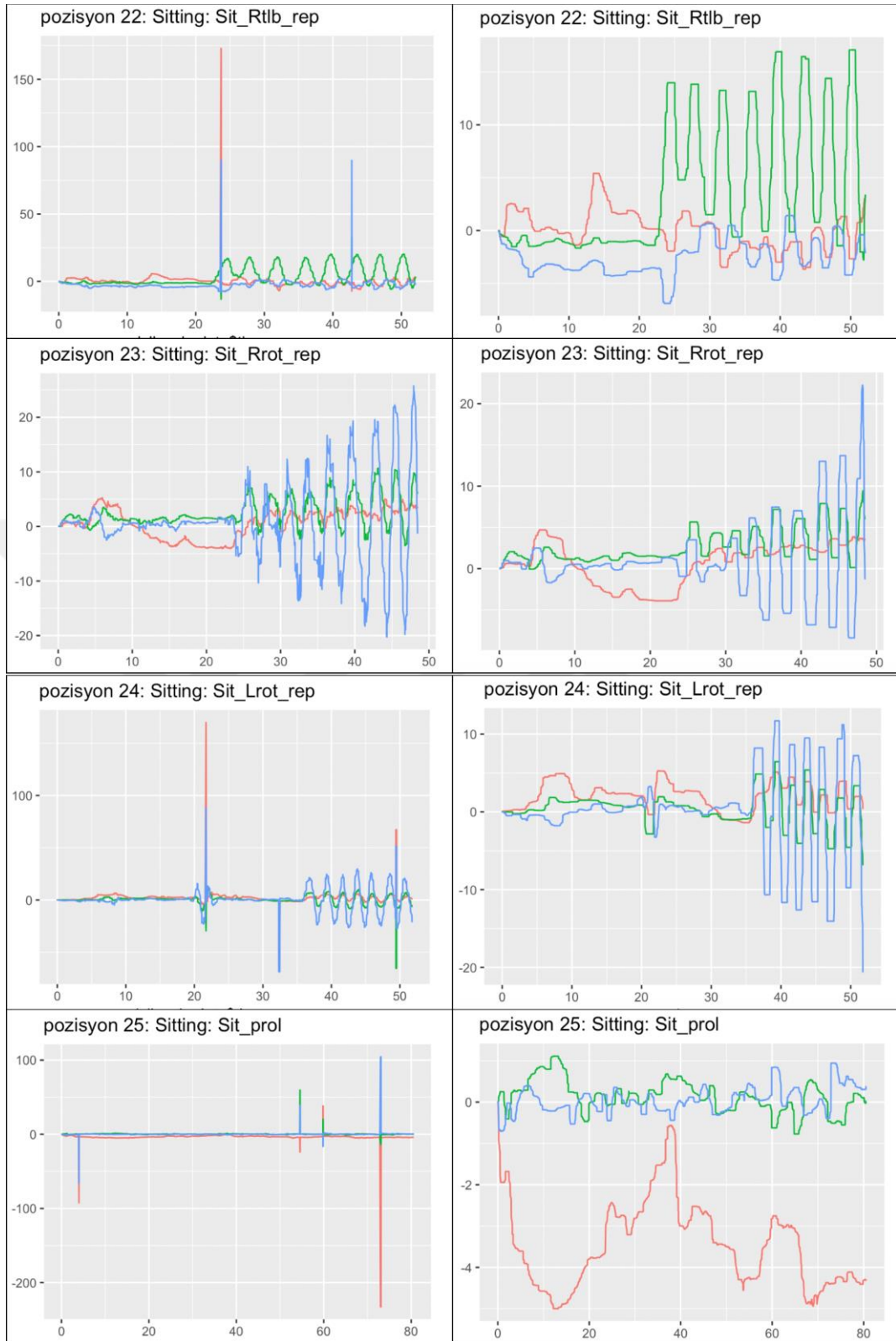


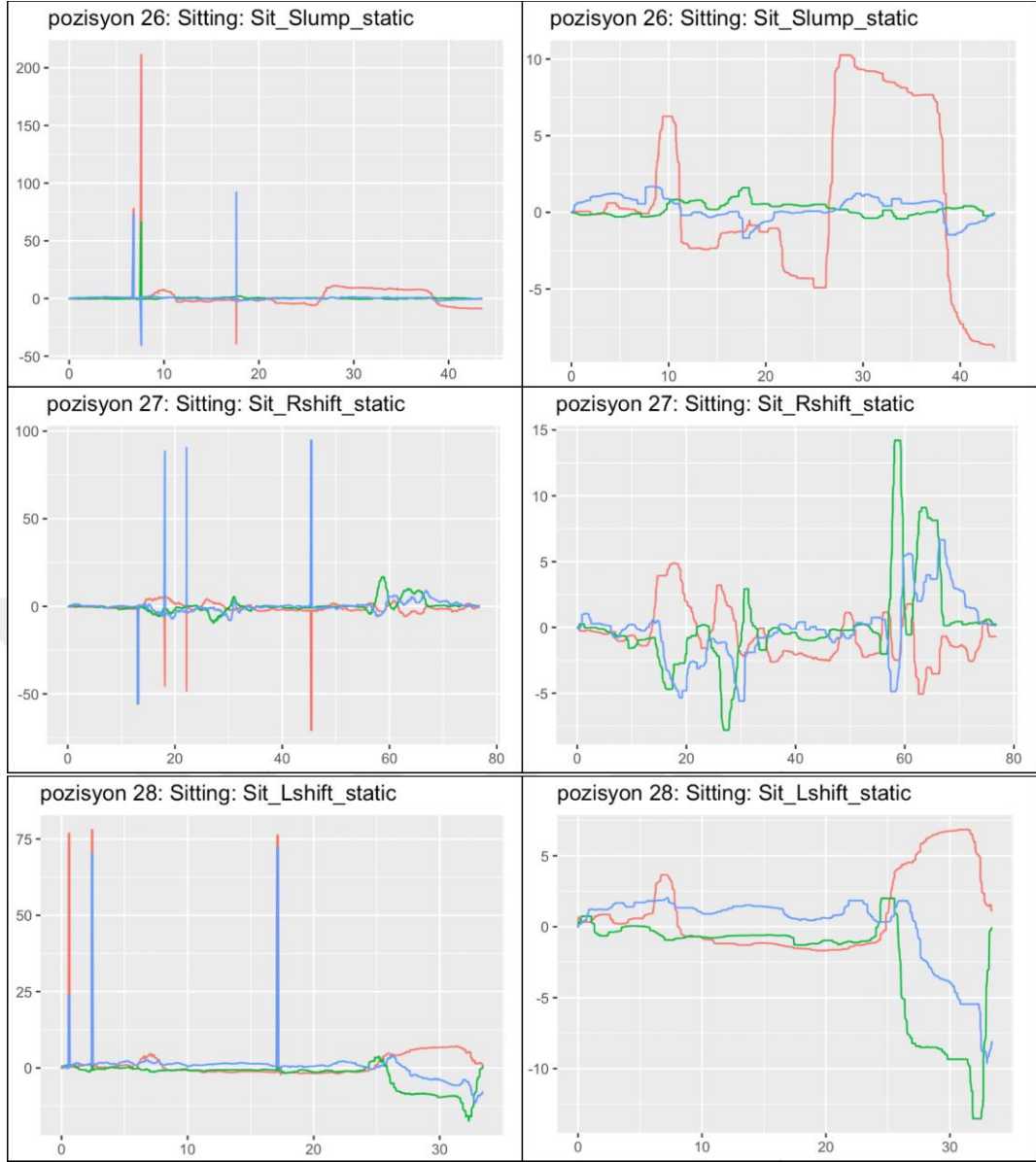


Şekil 2.11: Ayaktayken kaydedilmiş pozisyonların filtrelenmiş grafiği

Otururken ölçülmüş, filtrelenmiş pozisyonların, x,y,z eksenlerindeki Euler açılarına ait grafikleri aşağıdaki Şekil 2.12'deki gibidir. Filtrelenmemiş ve filtrelenmiş grafik görseller karşılaştırılarak median filtresi sonucu gösterilmiştir. Şekilde solda yer alan grafikler medyan filtresi uygulanmamış ham veriden elde edilmiş olup soldaki grafikler medyan filtre uygulanmış verilerle çizilen grafiklerdir.

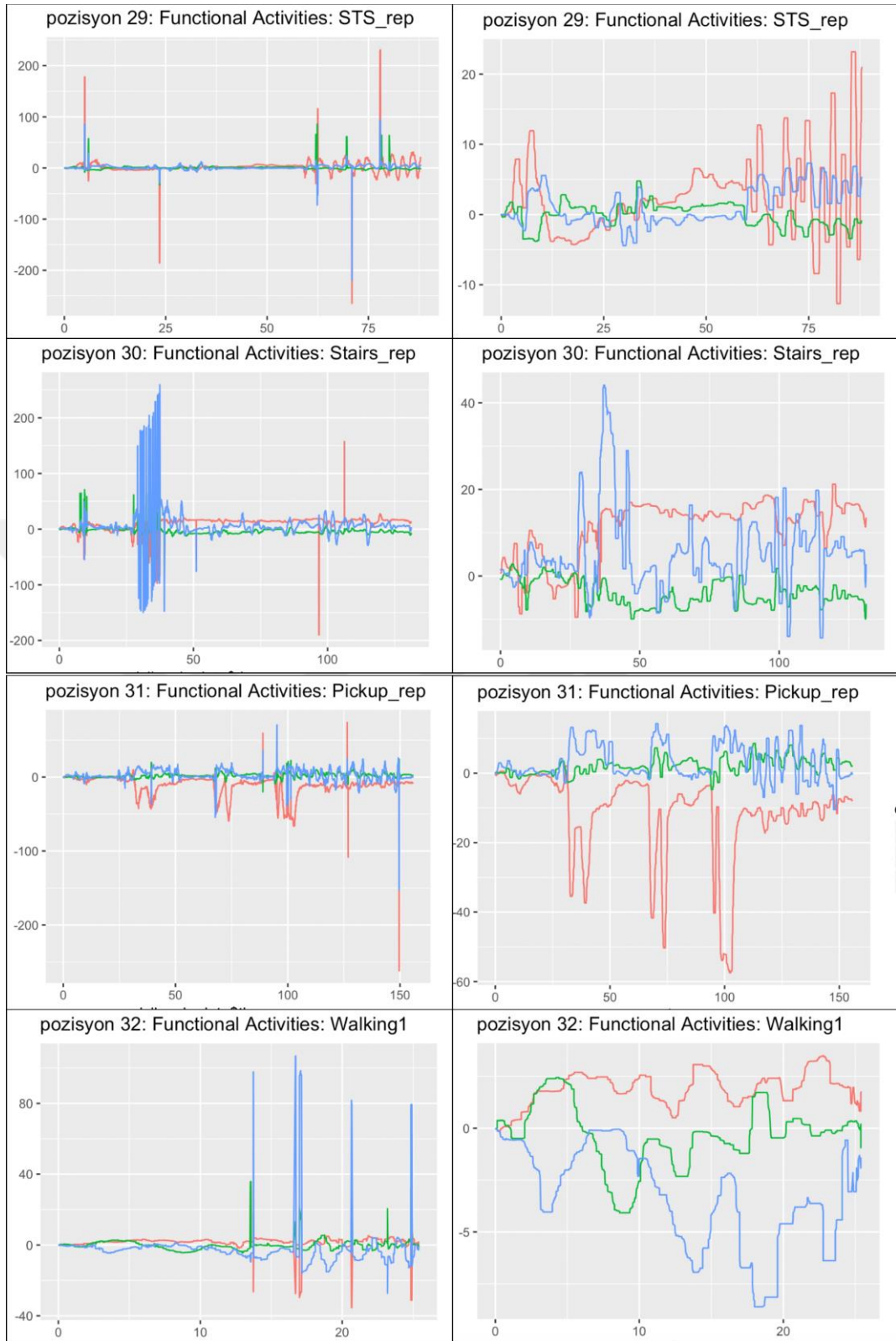


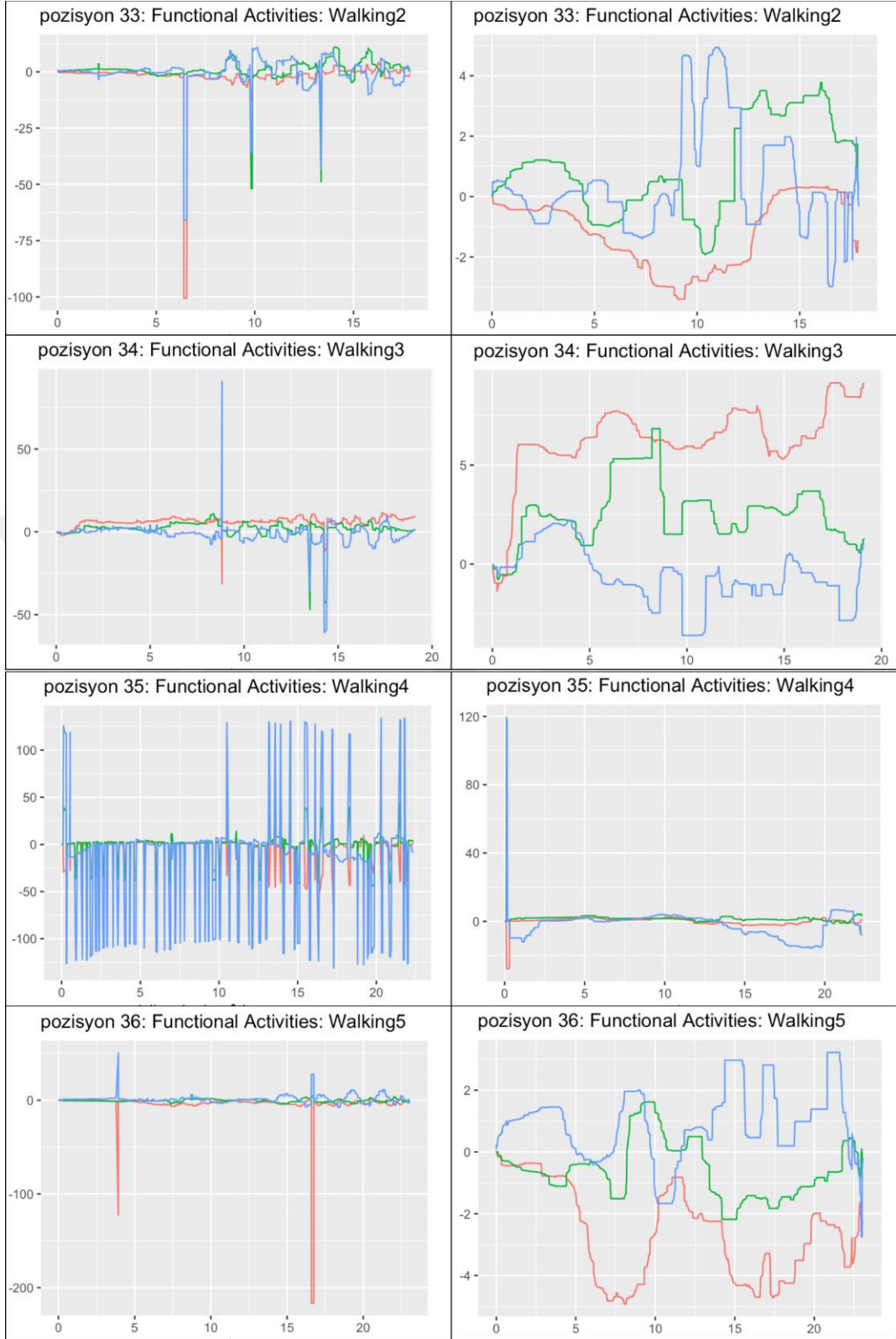




Şekil 2.12: Otururken kaydedilmiş pozisyonların filtrelenmiş grafiği

Fonksiyonel aktiviteler sırasında ölçülmüş, filtrelenmiş pozisyonların, x,y,z eksenlerindeki Euler açılarına ait grafikleri Şekil 2.13'deki gibidir. Filtrelenmemiş ve filtrelenmiş grafik görseller karşılaştırılarak medyan filtresi sonucu gösterilmiştir. Figürde solda yer alan grafikler medyan filtresi uygulanmamış ham veriden elde edilmiş olup soldaki grafikler medyan filtre uygulanmış verilerle çizilen grafiklerdir.

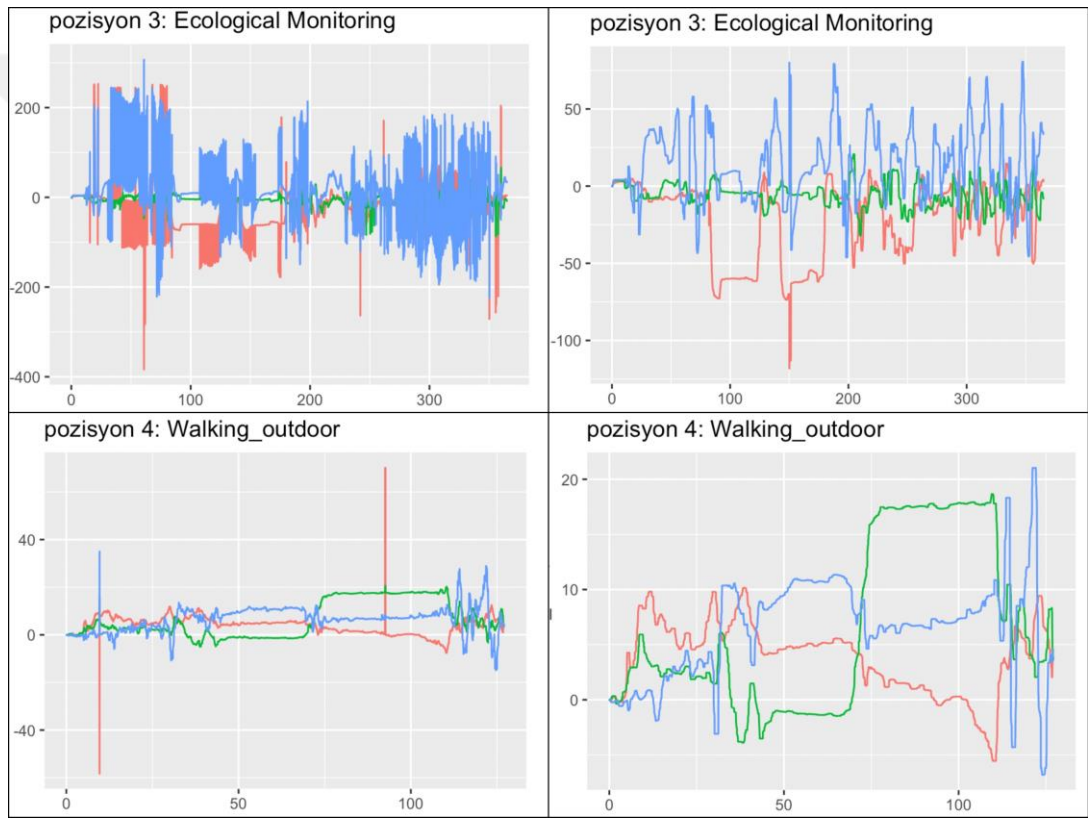




Şekil 2.13: Fonksiyonel aktiviteler sırasında kaydedilmiş pozisyonların filtrelenmiş grafiği

Veri setindeki, 3 grup altında topladığımız pozisyonlar dışında 3 numaralı “Ecological Monitoring” pozisyonu ve 4 numaralı “Walking_outdoor” pozisyonu da bulunmaktadır.

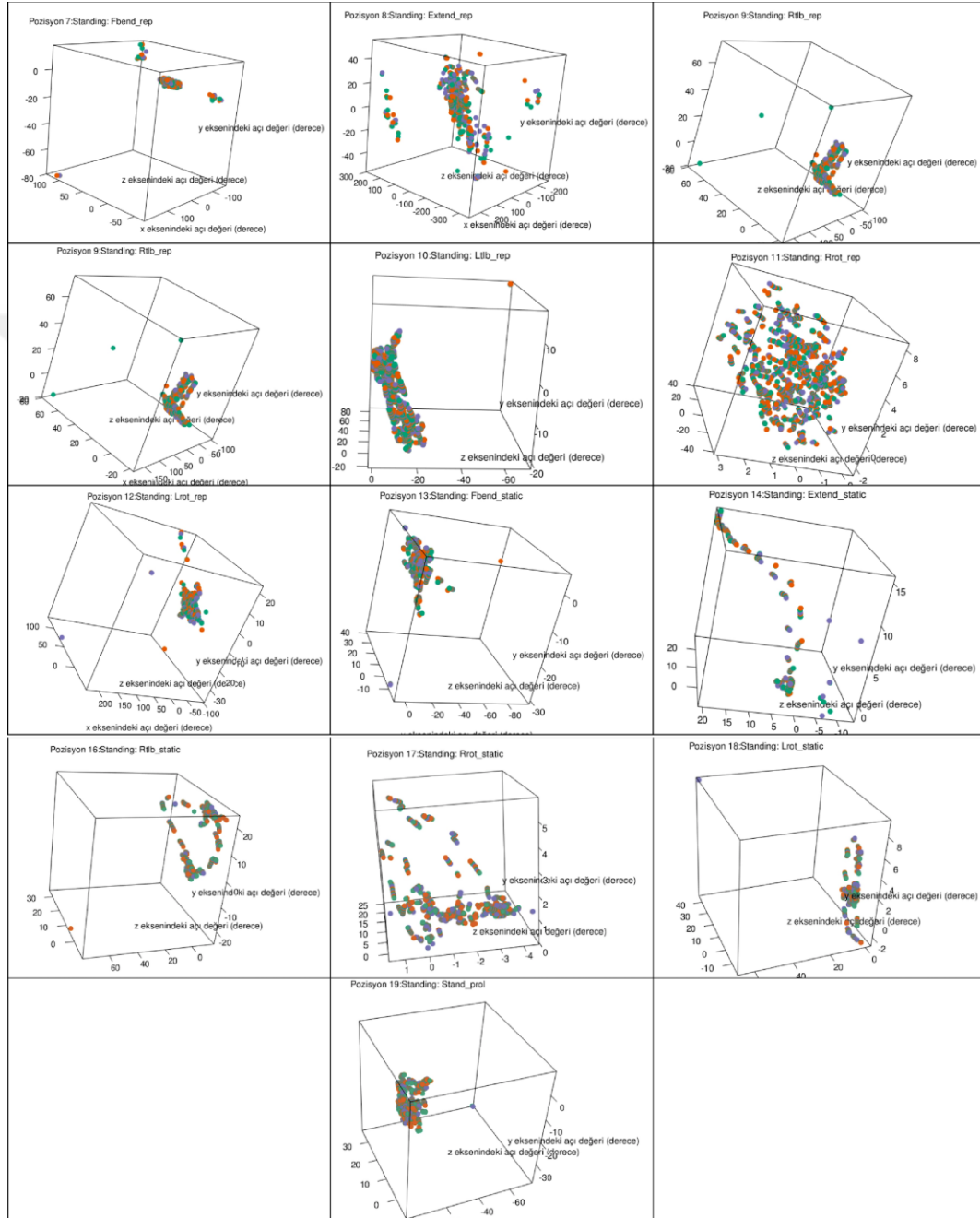
Ecological monitoring ve walking outdoor aktiviteleri sırasında ölçülmüş, filtrelenmiş pozisyonların, x,y,z eksenlerindeki Euler açılarına ait grafikleri aşağıdaki Şekil 2.14’deki gibidir. Filtrelenmemiş ve filtrelenmiş grafik görseller karşılaştırılarak median filtresi sonucu gösterilmiştir. Şekilde solda yer alan grafikler median filtresi uygulanmamış ham veriden elde edilmiş olup soldaki grafikler median filtre uygulanmış verilerle çizilen grafiklerdir.



Şekil 2.14: Ecological monitoring ve walking outdoor aktiviteleri sırasında ölçülmüş pozisyonların filtrelenmiş grafiği

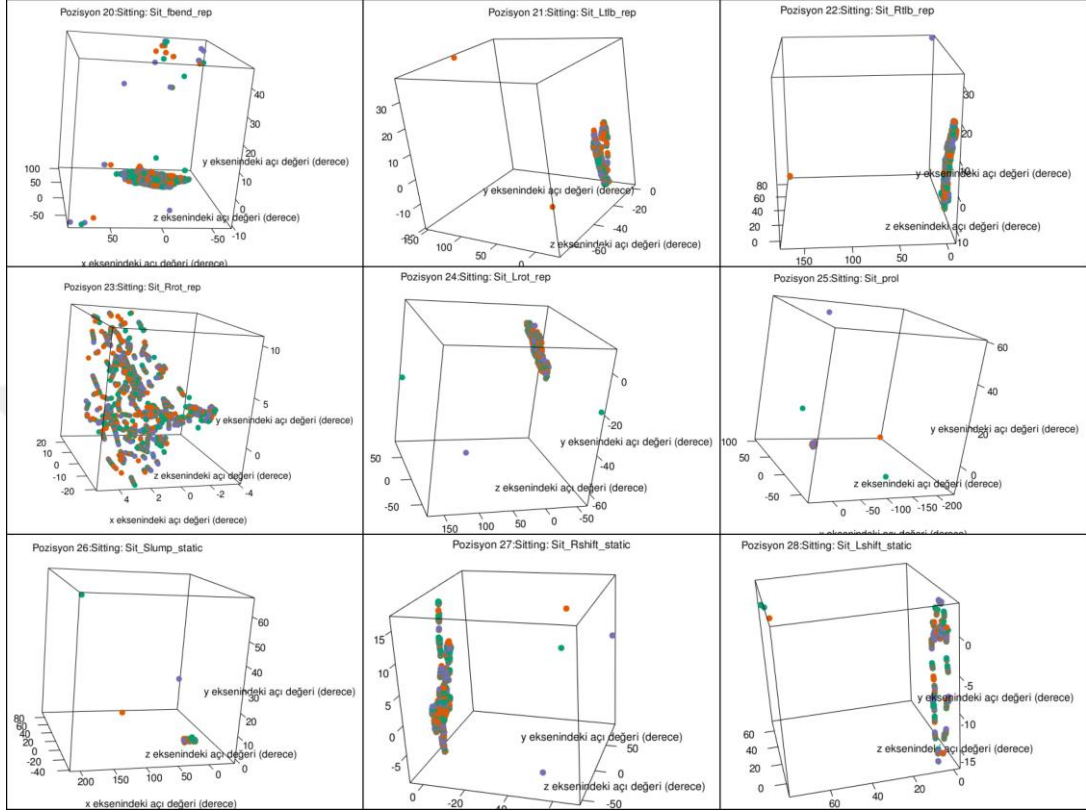
2.3.3 Verinin 3 Boyutlu Dağılım Grafiklerinin Çizilmesi

Ayaktayken ölçülmüş pozisyonlara ait dağılım grafikleri Şekil 2.15'deki gibidir. Grafikte x, y, z eksenlerindeki Euler açı değerlerini 3 boyutlu dağılımını göstermek amaçlanmaktadır.



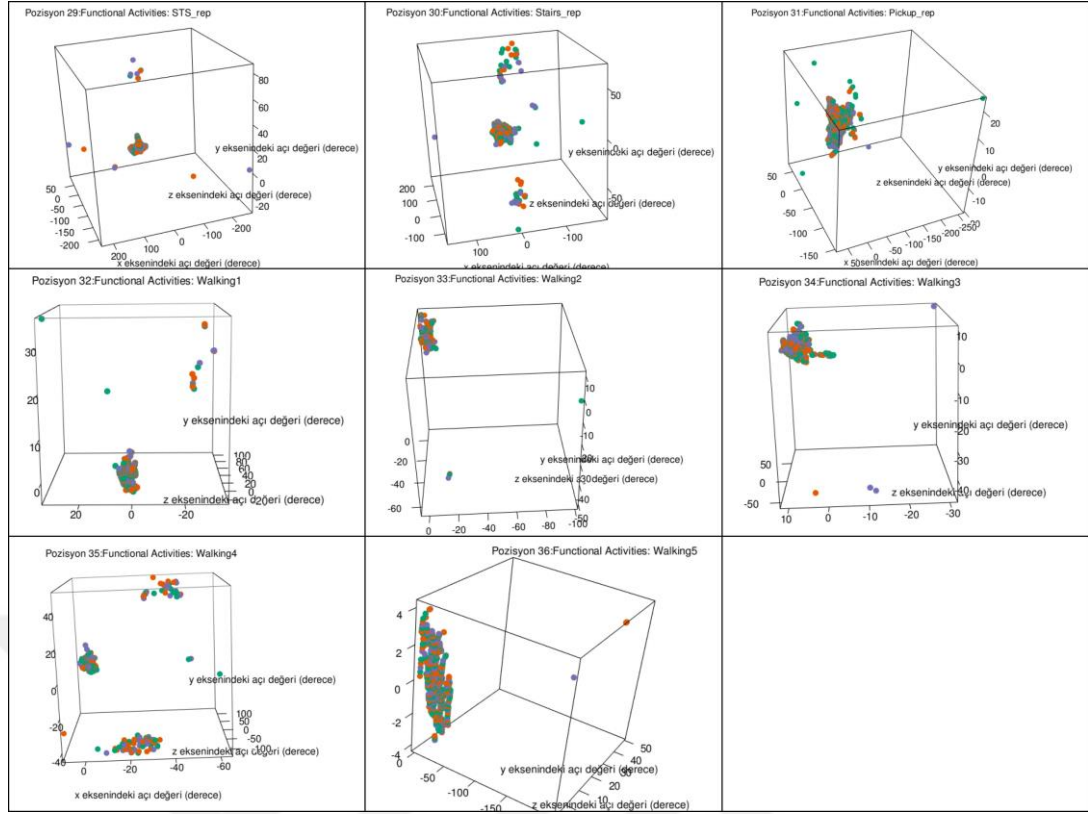
Şekil 2.15: Ayaktayken kaydedilen pozisyonların 3 boyutlu dağılım grafikleri

Otururken ölçülmüş pozisyonlara ait dağılım grafikleri Şekil 2.16'deki gibidir. Grafikte x, y, z eksenlerindeki Euler açı değerlerini 3 boyutlu dağılımını göstermek amaçlanmaktadır.



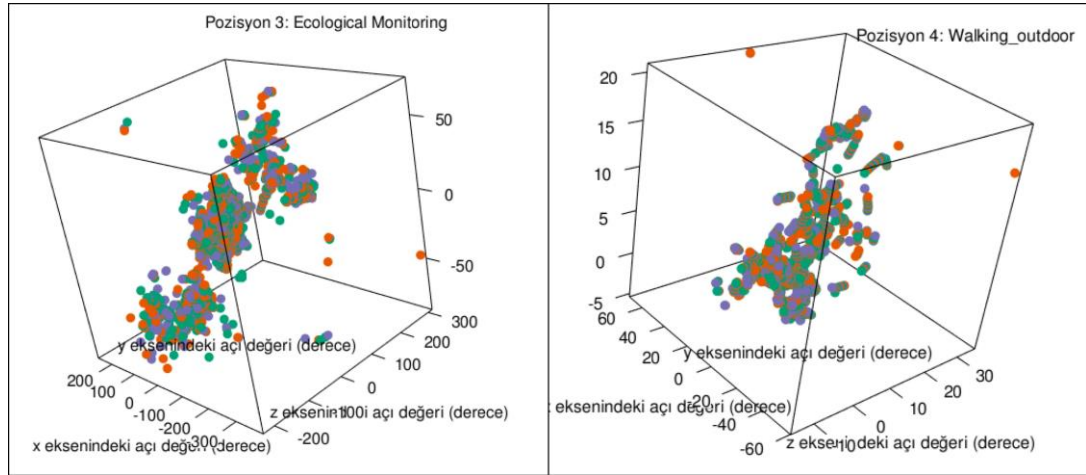
Şekil 2.16: Otururken kaydedilen pozisyonların 3 boyutlu dağılım grafikleri

Fonksiyonel aktiviteler sırasında ölçülmüş pozisyonlara ait dağılım grafikleri Şekil 2.17'deki gibidir. Grafikte x, y, z eksenlerindeki Euler açı değerlerini 3 boyutlu dağılımını göstermek amaçlanmaktadır.



Şekil 2.17 Fonksiyonel aktiviteler sırasında ölçülmüş pozisyonların 3 boyutlu dağılım grafikleri

Otururken, ayakta ve bazı fonksiyonel aktiviteler sırasında ölçülmüş pozisyonlar- dan ayrı olarak ölçülen “Ecological Monitoring” ve “Walking_outdoor” pozisyonlarına ait dağılım grafikleri Şekil 2.18’deki gibidir. Grafikte x, y, z eksenlerindeki Euler açı değerlerini 3 boyutlu dağılımını göstermek amaçlanmaktadır.



Şekil 2.18 Ecological monitoring ve Walking_outdoor aktiviteleri sırasında ölçülmüş pozisyonların 3 boyutlu dağılım grafikleri

2.4 VERİNİN KÜMELENMESİ

Verinin görselleştirme işlemi tamamlandıktan sonra veri setinde kümeleme işlemi denendi. Kümeleme işlemi için k-means kümeleme algoritması uygulandı. K-means kümeleme algoritması, veri madenciliğinde en çok kullanılan kümeleme algoritmalarından biridir. Algoritmanın amacı, veriyi daha küçük veya alt kümelere ayırmak olup istatistiksel olarak benzer nitelikteki kayıtları aynı grup altında toplamaktır. Veriler kümelere ayrılırken kullanılan matematiksel yöntem her sınıf için merkez belirlenen noktaya uzaklığa (aynı zamanda bu hata miktarıdır) göre yeni kümelerin yerleştirilmesidir. Öncelikle iki farklı pozisyon verisini birleştirerek küme sayısı iki olarak belirlendi.

Algoritma 4: K-means Algoritmasının Uygulanması

```
1: number_of_cluster ← 2
2: n ← length(clustering_data)
3: kmeans(clustering_data,number_of_cluster, iter.max=100000)$centers
```

Algoritma 4, küme sayısının önceden belirlenmiş olması durumunda uygulanmıştır. Algoritmada merkez sayısı olarak küme sayısı seçildi ve 1000 iterasyon ile çalıştırıldı. K-means algoritması sonucu olarak elde edilen değerler aşağıdaki gibidir.

x	y	z
-7.687502	2.045549	2.134151
-40.214332	1.857594	9.869710

Şekil 2.19: İki farklı pozisyona uygulanmış K-means algoritması sonucu

Şekil 2.19'deki Kmeans algoritması 1000 iterasyonla uygulanmış ve sonuç olarak iki farklı pozisyondan elde edilen, x y ve z eksenlerindeki Euler açılarının merkezlerinin belirlendiği görülmüştür. Bu örnek için ilk pozisyonun x ekseninde aldığı Euler x açılarının merkezinin -7.687502 derece olduğu anlaşılmaktadır. Aynı şekilde Euler y açılarının merkezi 2.045549 derece ve Euler z açılarının merkezi 2.134151 derece olarak belirlenmiştir. İkinci pozisyona ait kümeleneş x, y, z açılarının merkez değerleri de ikinci satırda yer almaktadır.

2.4.1 Verideki Küme Sayısının Belirlenmesi

Şekil 2.19'deki örnekte küme sayısı önceden belirlenmiştir. Fakat küme sayısı belirli olmayan veri setleri için küme sayısını belirleyen metotlar vardır. Bunlar direkt yöntemler veya istatistiksel test yöntemleri olarak ikiye ayrılır:

- Direkt yöntemler: Elbow ve Silhouette metotlarıdır.
- İstatistiksel test yöntemi ise: Gap Statistic metodudur.

Elbow, Silhouette ve Gap Statistic metodları ayrı ayrı denenerak test edilmiş ve algoritma sonucunda oluşan grafikler yorumlanarak; küme sayısı belirlemede en iyi sonuç veren yöntem belirlenmiştir.

Silhouette Metot: Kümedeki her noktanın komşu kümelerdeki noktalara ne kadar yakın olduğunu ölçerek küme sayısını bulmaya yarayan metottur.

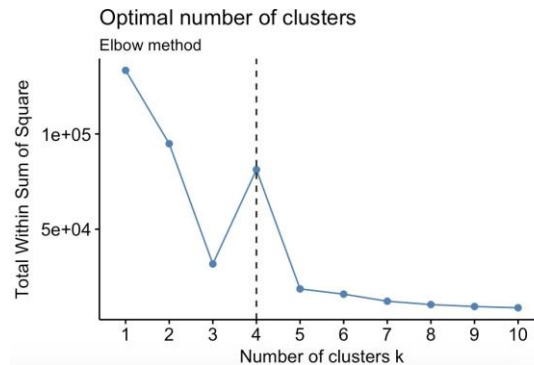
Elbow Metot: Bir veri kümesinde uygun sayıda kümenin bulunmasına yardımcı olmak için tasarlanan küme analizindeki tutarlılığın yorumlanması ve doğrulanması yöntemidir.

Gap Statistic Metot: Denetlenmesiz bir kümeleme için "doğru" küme sayısına (k) yaklaşmak için bir yöntemdir. Bu yöntemde, küme içindeki değerlerin kareler toplamı değerlendirilir.

Algoritma 5: Elbow Metodu (K-means Algoritması küme sayısının belirlenmesi)

```
1: fviz_nbclust(clustering_data[,3:5], kmeans, method = "wss") +  
2: geom_vline(xintercept = 4, linetype = 2)+  
3: labs(subtitle = "Elbow method")
```

Algoritma 5'teki Elbow metodunun uygulama sonucunda küme sayısını gösteren grafik aşağıdaki gibidir.



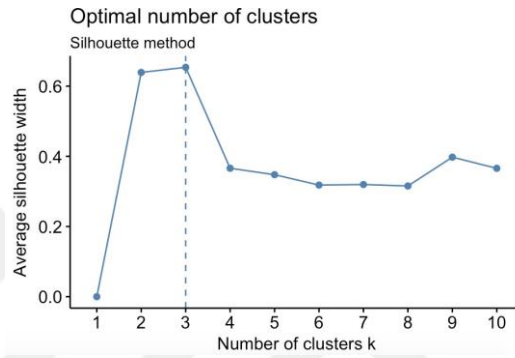
Şekil 2.20: Elbow metodu sonucuna göre verideki küme sayısını gösteren grafik

Şekil 2.20’de görüldüğü gibi veri setinde 3 farklı pozisyon bulunmasına rağmen Elbow metodu veride 4 pozisyon kümesi bulmuştur.

Algoritma 6: Silhouette Metodu (K-means Algoritması küme sayısının belirlenmesi)

- 1: fviz_nbclust(clustering[,3:5], kmeans, method = "silhouette") +
 - 2: geom_vline(xintercept = 4, linetype = 2)+
 - 3: labs(subtitle = "Silhouette method")
-

Algoritma 6’daki Silhouette metodunu uygulama sonucunda küme sayısını gösteren grafik aşağıdaki gibidir.



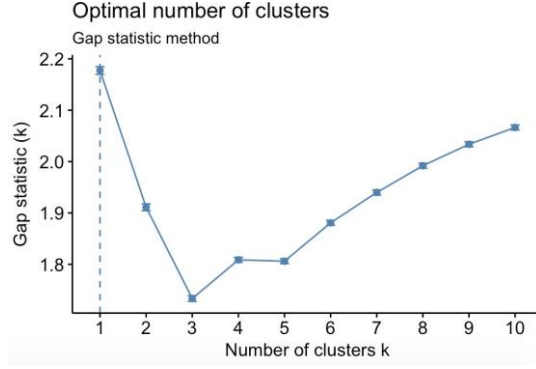
Şekil 2.21: Silhouette metodu sonucuna göre küme sayısını gösteren grafik

Şekil 2.21’de görüldüğü gibi veri setinde 3 farklı pozisyon bulunmaktadır ve Silhouette metodu da veride 3 pozisyon kümesi bulmuştur.

Algoritma 7: Gap Statistic method (K-means Algoritması küme sayısının belirlenmesi)

- 1: set.seed(123)
 - 2: fviz_nbclust(clustering[,3:5], kmeans, nstart = 25, method = "gap_stat", nboot = 500) +
 - 3: labs(subtitle = "Gap statistic method")
-

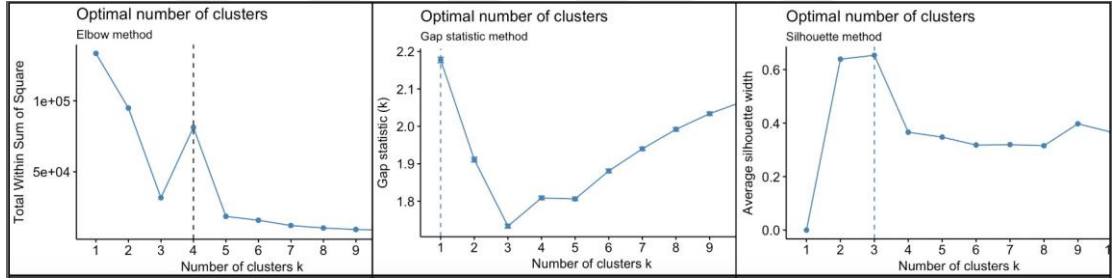
Algoritma 7’deki Gap Statistic methodunun uygulama sonucunda küme sayısını gösteren grafik aşağıdaki gibidir.



Şekil 2.22: Gap Statistic metodu sonucuna göre verideki küme sayısını gösteren grafik

Şekil 2.22’de görüldüğü gibi veri setinde 3 farklı pozisyon bulunmasına rağmen Gap Statistic metodu veride 1 pozisyon kümesi bulmuştur.

Şekil 2.20’de bulunan Elbow metodu sonucu, Şekil 2.21’de bulunan Silhouette metodu sonucu, Şekil 2.22’de bulunan Gap Statistic metodu sonucu karşılaştırıldığında en doğru sonucu Silhouette metodunun verdiği görülmektedir.



Şekil 2.23: Elbow-Gap Statistic-Sihouette metotları sonuçlarının karşılaştırılması

ÜÇÜNCÜ BÖLÜM

3.1 DERİN ÖĞRENME TABANLI AKTİVİTE TANIMA

3.1.1 Derin Öğrenme Metodolojisi

Vücut kinematiği verileri üzerinde temizleme, filtreleme, görselleştirme ve kümeleme metotları uygulandıktan sonra son olarak bir makine öğrenme yöntemi olan derin öğrenme yöntemi uygulanacaktır. Derin öğrenme, yapay sinir ağlarını kullanmaktadır. Yapay sinir ağları, tıpkı insan beyni gibi nöronlardan oluşur. Tüm nöronlar birbirine bağlıdır ve çıktıyı etkilemektedir. Nöronlar temel olarak 3 katmandan oluşmaktadır:

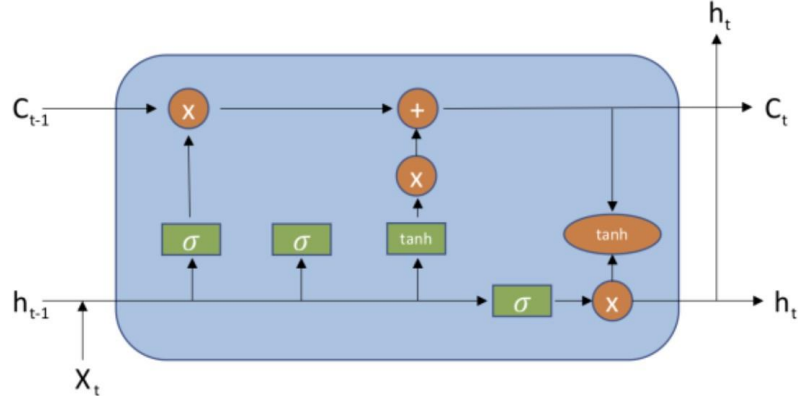
- 1. Giriş Katmanı:** Giriş verilerinin bulunduğu katmandır. Giriş verilerinin ilk gizli katmana iletir.
- 2. Gizli Katman (lar):** Girdilerimizde matematiksel hesaplamalar yapar. Yapay sinir ağları oluşturmadaki zorluklardan biri, her bir katman için nöronların sayısının yanı sıra gizli katmanların sayısına da karar vermektir. Derin öğrenmedeki, “derin” ifadesi birden çok gizli katmana sahip olmayı ifade eder.
- 3. Çıkış Katmanı:** Çıkış verilerinin oluşturulduğu katmandır.

Derin öğrenme çıkış katmanındaki çıkış verilerinin tahmini ile ilgilenmektedir. Derin öğrenmenin en zor kısımlarından biri yapay sinir ağını eğitmektir. Çünkü derin öğrenme uygulayabilmek için çok büyük bir veriye ve çok fazla miktarda hesaplama gücüne ihtiyaç vardır.

Bu çalışmada sensörlerinden verilerinden elde edilen vücut kinematiği izleme verileri için TensorFlow’da uygulanan bir LSTM (Long Short-Term Memory) Sinir Ağı eğitilmesi uygulanacaktır.

LSTM sinir ağının genel şeması aşağıda verilmiştir.

Şekil 3.1’de verilen LSTM şemasında mavi renkli kutular puanlama işlemini gerçekleştirme olup sarı renkli şekiller katmanları ifade etmektedir.



Şekil 3.1: LSTM sinir ağı şeması

3.1.2 Vücut Kinematığı Verileri Üzerinde Derin Öğrenme Uygulaması

Bu çalışmada, derin öğrenme uygulamaları Python programlama dilinde gerçekleştirilmiştir. Kullanılan başlıca kütüphaneler: numpy, pandas, tensorflow ve seaborn kütüphaneleridir.

Numpy kütüphanesi: Matematiksel işlemleri içeren bir kütüphane olup matrisler ve çok boyutlu dizilerle işlem yapmak için kullanılmaktadır.

Pandas kütüphanesi: Veri analiz ve işleminde kullanılan kütüphanedir. Asıl işi zaman serileri ve sayısal tablolar üzerinde işlem yapmaktır.

Tensorflow kütüphanesi: Veri akışı ve değiştirilebilir dataflow programlama için kullanılan bir kütüphanedir. Makine öğrenmesi uygulamalarında sinir ağları üzerinde uygulanabilmektedir.

Seaborn kütüphanesi: İçerik olarak zengin, dikkat çekici istatistiksel grafikler elde etmek için kullanılan kütüphanedir. Özellikle matplotlib kütüphanesi üzerine inşa edilmiştir.

Öncelikle çalışmamızda kullandığımız veri setine pozisyonların numaraları numerik olarak yeni oluşturduğumuz “class” sütununa eklendi. Ayrıca pozisyon numaralarının isimleri de karakter olarak yeni oluşturduğumuz “activities” sütununa eklendi. Düzenlenen veri seti Şekil 3.2’deki gibidir.

Fbend_rep	7	0.061	-0.0026774571	0.006780465	-0.04978697
Fbend_rep	7	0.090	-0.0004084963	0.103928704	-0.09517221
Fbend_rep	7	0.091	0.0005027139	0.106410099	-0.09758568
Fbend_rep	7	0.119	0.0036351639	0.107317985	-0.09829972
Fbend_rep	7	0.120	0.0036351639	0.107502550	-0.10195900
Fbend_rep	7	0.209	0.0036351639	0.107317985	-0.10726079
Fbend_rep	7	0.211	0.0005027139	0.106410099	-0.10726079
Fbend_rep	7	0.270	0.0003271739	0.107317985	-0.10780485
Fbend_rep	7	0.271	0.0005027139	0.107502550	-0.10924501
Fbend_rep	7	0.272	0.0033384701	0.107850691	-0.10949979
Fbend_rep	7	0.299	0.0036351639	0.107883800	-0.11082485
Fbend_rep	7	0.300	0.0036351639	0.107896756	-0.11222430

Şekil 3.2: Derin Öğrenme modeli için yeniden düzenlenmiş (filtrelenmiş) vücut kinematığı verisi

Modeli oluşturmak için hazırladığımız veri setinin sütun isimleri: COLUMN_NAMES = ['class', 'activity', 'timestamp', 'x-axis', 'y-axis', 'z-axis'] şeklindedir.

Veri setindeki pozisyon numarası bilgisi veriyi etiketlemek için kullanılacaktır. Bu nedenle etiketler aşağıdaki şekilde tanımlanmıştır:

```
LABELS = [ 'Ecological_Monitoring', 'Walking_outdoor', 'Fbend_rep', 'Extend_rep', 'Rtlb_rep', 'Ltlb_rep', 'Rrot_rep', 'Fbend_static', 'Extend_static', 'Ltlb_static', 'Rtlb_static', 'Rrot_static', 'Lrot_static', 'Stand_prol', 'Sit_fbend_rep', 'Sit_Ltlb_rep', 'Sit_Rtlb_rep', 'Sit_Rrot_rep', 'Sit_prol', 'Sit_Slump_static', 'Sit_Rshift_static', 'Sit_Lshift_static', 'STS_rep', 'Stairs_rep', 'Pickup_rep', 'Walking1', 'Walking2', 'Walking3', 'Walking4' ]
```

Modelde yer alacak class sayısı pozisyon sayısına eşitlenerek 29 olarak belirlenirken, 3 farklı veri (x, y ve z eksenlerindeki Euler açı değerleri) olduğu için modeli oluştururken özellik sayısı 3'e eşitlenmiştir.

Kullandığımız modelde, bütün bir veri kümesinin sinir ağında hem bir kez hem ileri hem geri geçirildiği zaman olarak tanımlanan, eğitim turu (epoch) sayısı başlangıçta 50 olarak denenmiş sonrasında eğitim turu sayısı artırılarak sonuçları karşılaştırılmıştır. Derin Öğrenme yöntemine ait parametre ve hiper parametrelerin tanımlanması Şekil 3.3'deki gibidir.


```
# Data preprocessing
TIME_STEP = 100

# Model
N_CLASSES = 29
N_FEATURES = 3 # x-acceleration, y-acceleration, z-acceleration

# Hyperparameters
N_LSTM_LAYERS = 2
N_EPOCHS = 100
L2_LOSS = 0.0015
LEARNING_RATE = 0.0025

# Hyperparameters optimized
SEGMENT_TIME_SIZE = 180
N_HIDDEN_NEURONS = 30
BATCH_SIZE = 32
```

Şekil 3.3: Derin Öğrenme yöntemine ait parametre ve hiper parametrelerin tanımlanması

3.1.3 LSTM Sinir Ağının Oluşturulması

LSTM sinir ağı oluşturulurken giriş katmanında yığın boyutu (BATCH_SIZE), segmentlere ayrılan verinin her bir segment boyutu (SEGMENT_TIME_SIZE) ve veri setindeki özellik sayısı (N_FEATURES) bulunmaktadır.

LSTM sinir ağındaki “gizli katmanların ağırlıkları” (Weights-hidden) oluşturulurken; veri setindeki özellik sayısı (N_FEATURES) ve gizli nöron sayısı (N_HIDDEN_NEURONS) arasında normal dağılım uygulayarak rastgele değerler üretilmiştir.

LSTM sinir ağındaki “çıkış katmanı ağırlıkları” (Weights-output) oluşturulurken; veri setindeki gizli nöron sayısı (N_HIDDEN_NEURONS) ve veri setindeki sınıf sayısı (N_CLASSES) arasında normal dağılım uygulayarak rastgele değerler üretilmiştir.

LSTM sinir ağının gizli bias değeri oluşturulurken, gizli nöron sayısının (N_HIDDEN_NEURONS) normal dağılım ortalaması 1 olacak (mean=1) şekilde rastgele değerler üretilmiştir.

Benzer şekilde LSTM sinir ağının çıkış bias değeri oluşturulurken, veri setindeki sınıf sayısı için (N_CLASSES) normal dağılım uygulanarak rastgele değerler üretilmiştir.

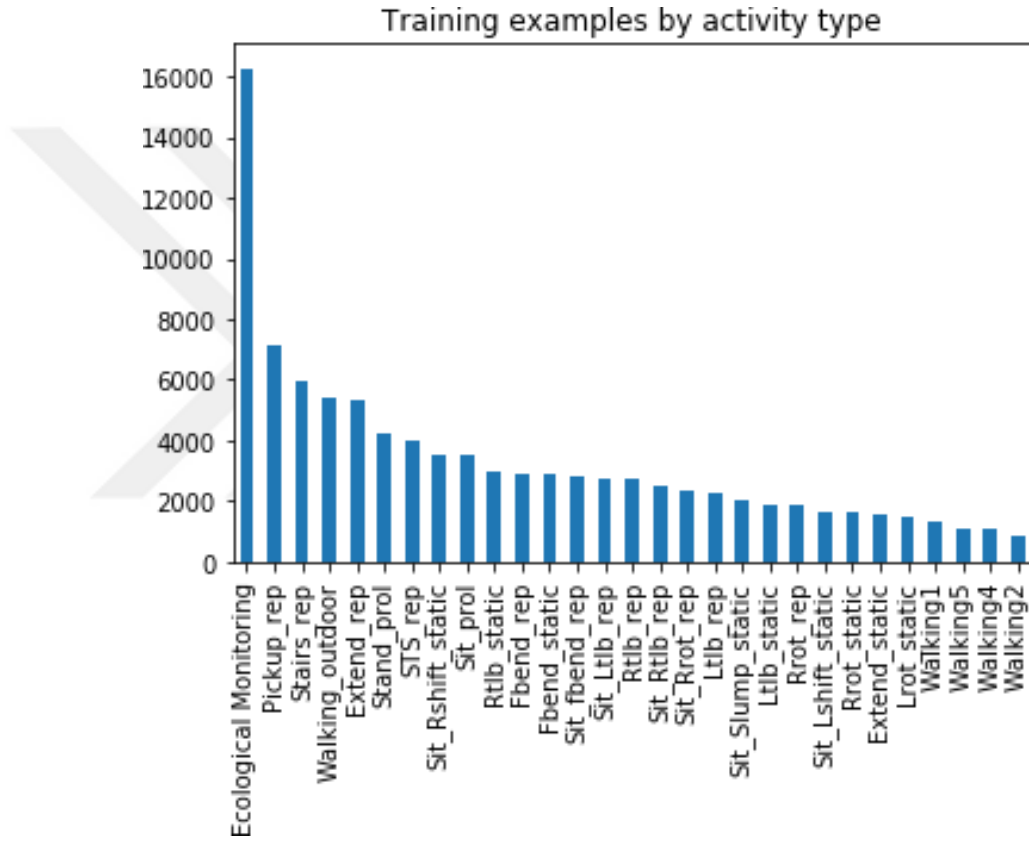
LSTM katman sayısı 2 olarak belirlenen bu modelde, iki LSTM hücresi üst üste birleştirilerek katmanlar oluşturulmuştur. Son olarak çoklu girişten tek çıkış elde

edilerek sinir ağındaki çıktısına ulaşılmıştır.

Modelimiz, her biri 64 ünite içeren 2 tam bağlantılı ve 2 LSTM katmanı (üst üste istiflenmiş) içermektedir.

3.1.4 Veri Setinin Görselleştirilmesi

Derin öğrenme metodunu uygulamak için düzenlediğimiz veri seti Python çalışma ortamına yüklenmiş ve öncelikle veri setindeki her bir aktiviteye ait kayıt sayısı Şekil 3.4'deki gibi gösterilmiştir.

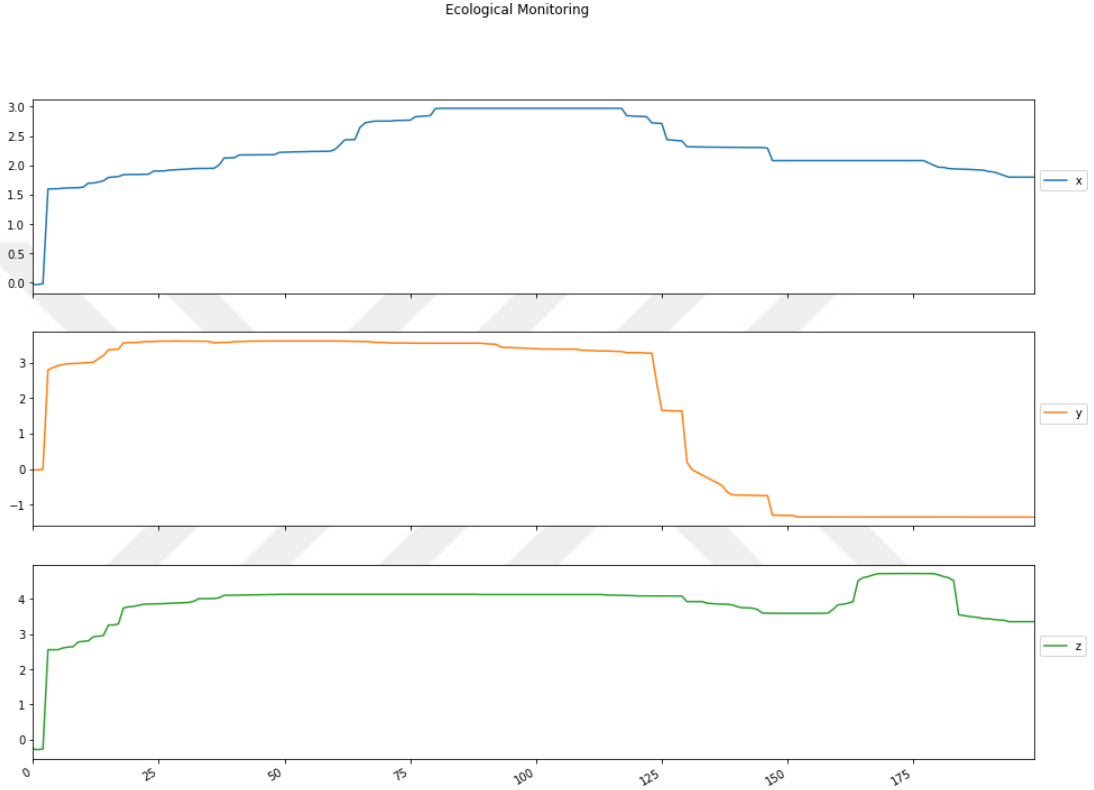


Şekil 3.4 Veri setindeki her bir aktiviteye ait kayıt sayısı

Aktivitelere ait kayıt sayıları gösterildikten sonra her bir aktiviteye ait x, y, z eksenlerindeki Euler açı değerlerinin grafiklerini çizdirek plot_activity fonksiyonu tanımlanıp, örnek olarak istenilen aktivitelere uygulandı. Örneğin "Ecological Monitoring" pozisyonuna ait x, y, z eksenlerindeki Euler açı değerleri Şekil 3.5'deki gibi gösterilmiştir.

3.1.5 Derin Öğrenme Modelinin Oluşturulması

LSTM (2.5.1.1 numaralı bölümde ele alınmıştır) modelimiz, sabit uzunluklu dizileri eğitim verisi olarak beklemektedir. Bunları üretmek için bilinen bir yöntem kullanılacaktır. Üretilen her dizi ise 100 eğitim örneği içerir. Veri setinin % 30'u test verisi; %70'i eğitim verisi olarak kullanılmıştır.



Şekil 3.5: "Ecological Monitoring" isimli pozisyona ait x, y, z eksenlerindeki Euler açı değerleri

Modelimizde, giriş ve çıkış tensörleri ayrı ayrı tanımlamıştır. Çıkış tahminleri için tensörü uygun şekilde adlandırmalıyız. L2 düzenleyici kullanılacak ve kullandığımız bu düzenleyici, kaybedilenleri de (loss) belirtilmelidir.

Model oluşturulurken doğruluk parametrelerinin neler olduğu ve iyileştirici (optimizer) olarak "AdamOptimizer" tanımlanmıştır.

AdamOptimizer: eğitim verileri üzerinde yinelemeli ağ ağırlıklarını güncellemek için klasik stokastik gradyan iniş prosedürü yerine kullanılabilecek bir optimizasyon algoritmasıdır.

AdamOptimizer, klasik stokastik gradyan inişinden farklıdır. Bu yöntem, farklı parametreler için bireysel öğrenme oranlarını, gradyanların birinci ve ikinci değerlerinden çıkardığı tahminlerinden hesaplar. Ayrıca yöntemin yazarları, Adam'ı diğer stokastik gradyan iniş uzantılarının, Adaptive Gradient Algorithm (AdaGrad) ve Root Mean Square Propagation (RMSProp) avantajlarının birleştirilerek geliştirildiğini belirtmişlerdir.

Yöntem de kullanılan bazı hiper parametreler aşağıda verilmiştir.

Eğitim Turu (Epoch): Eğitim modeli tek seferde eğitilmediği için her seferinde tekrar eğitmek için ağırlıklar güncellenir. Eğitilen her yeni eğitim verisi için en baştan ağırlık hesabı yapılır. Bu sayede model için en uygun ağırlık değerleri hesaplanmış olur. Bu adımların her birine epoch denir.

Yığın (Parti) boyutu (Batch Size): Modeli uygularken modelin kullandığı bellek olarak tanımlanabilir. 2 ve katları şeklinde (4, 8, 16, 32, 64,..., 512) değer almaktadır.

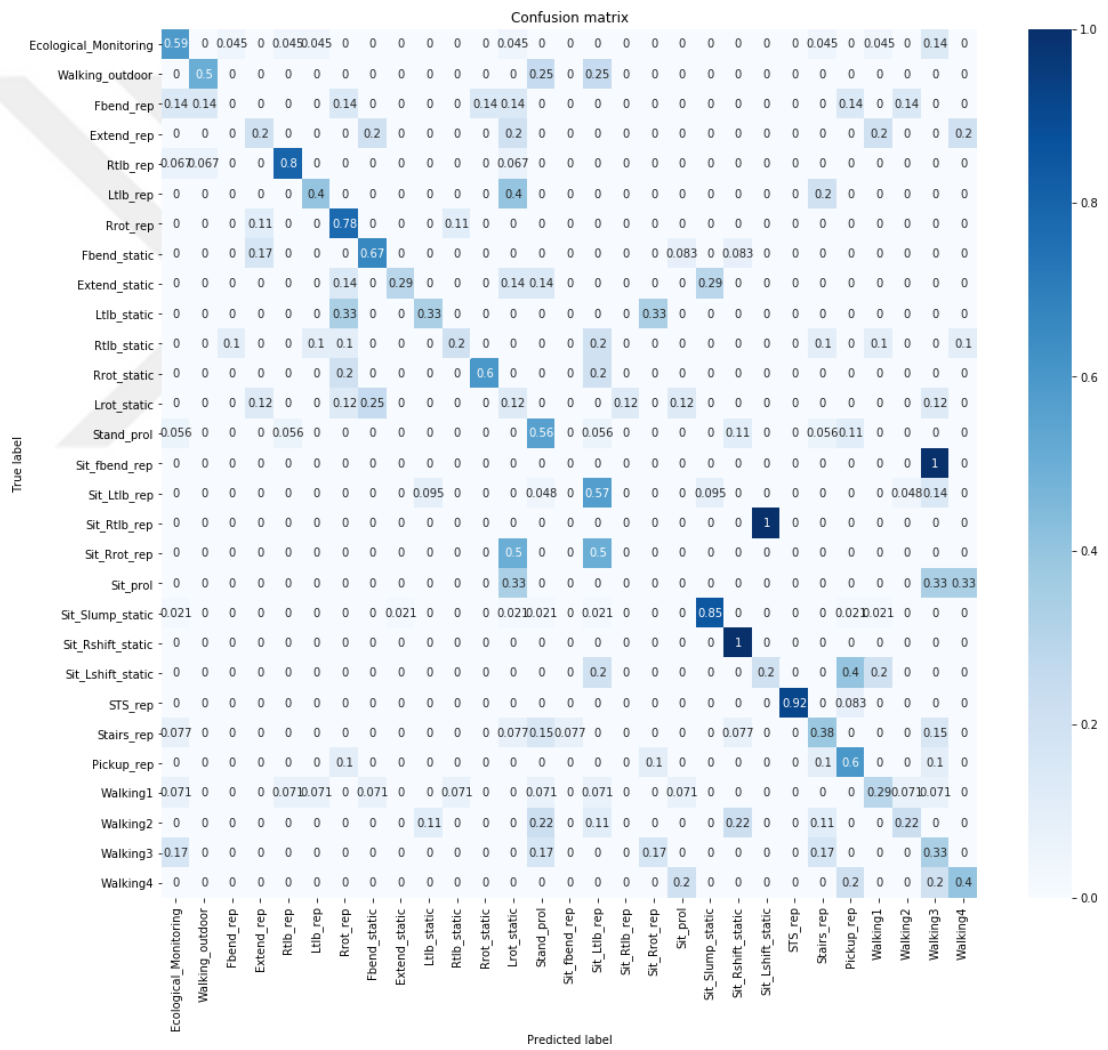
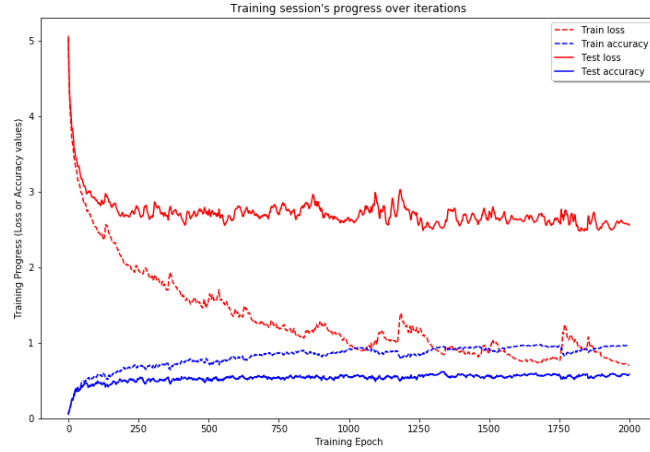
3.1.6 Derin Öğrenme Metodu Testleri

3.1.6.1 LSTM Katman Sayısının Karşılaştırılması

Bu bölümdeki testlerde LSTM katman sayısında yapılan değişiminin, testin doğruluğu (accuracy) ve testteki kayıp (loss) üzerindeki etkisini görmek amaçlanmaktadır. Derin öğrenme yöntemini diğer yapay sinir ağlarından ayıran özellikle de karmaşık problemlerde iyi sonuç vermesini sağlayan en önemli özelliklerinden biri de katman sayısıdır. Bu doğrultuda eğitim turu sayısı 100 olarak ve yığın (parti) boyutu (batch_size) 32 olacak şekilde LSTM katman sayısı arttırılarak testler yapılmıştır. Testlerde tüm pozisyonlar kullanılmış olup, modeldeki sınıf sayısı (N_CLASSES) 29 olarak belirlenmiştir.

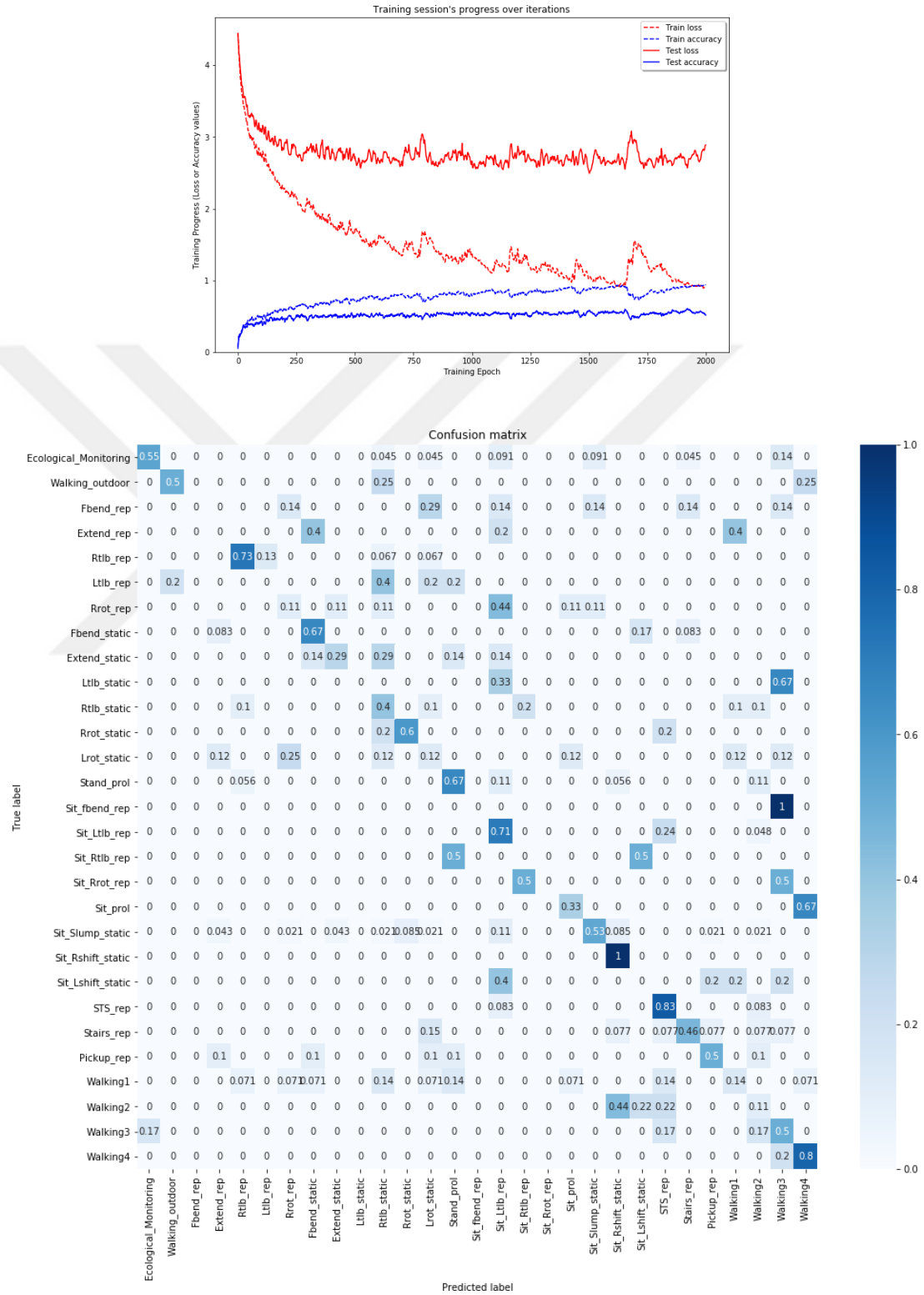
LSTM Katman sayısının doğruluk ve kayıp üzerindeki etkisini görmek için;

1. testte LSTM katman sayısı 2 olarak belirlenmiştir. Test sonucunda, test doğruluğu (accuracy) yaklaşık 0.5833 (%58.3) ve testteki kayıp (loss) ise 2.55616 olarak hesaplanmıştır.



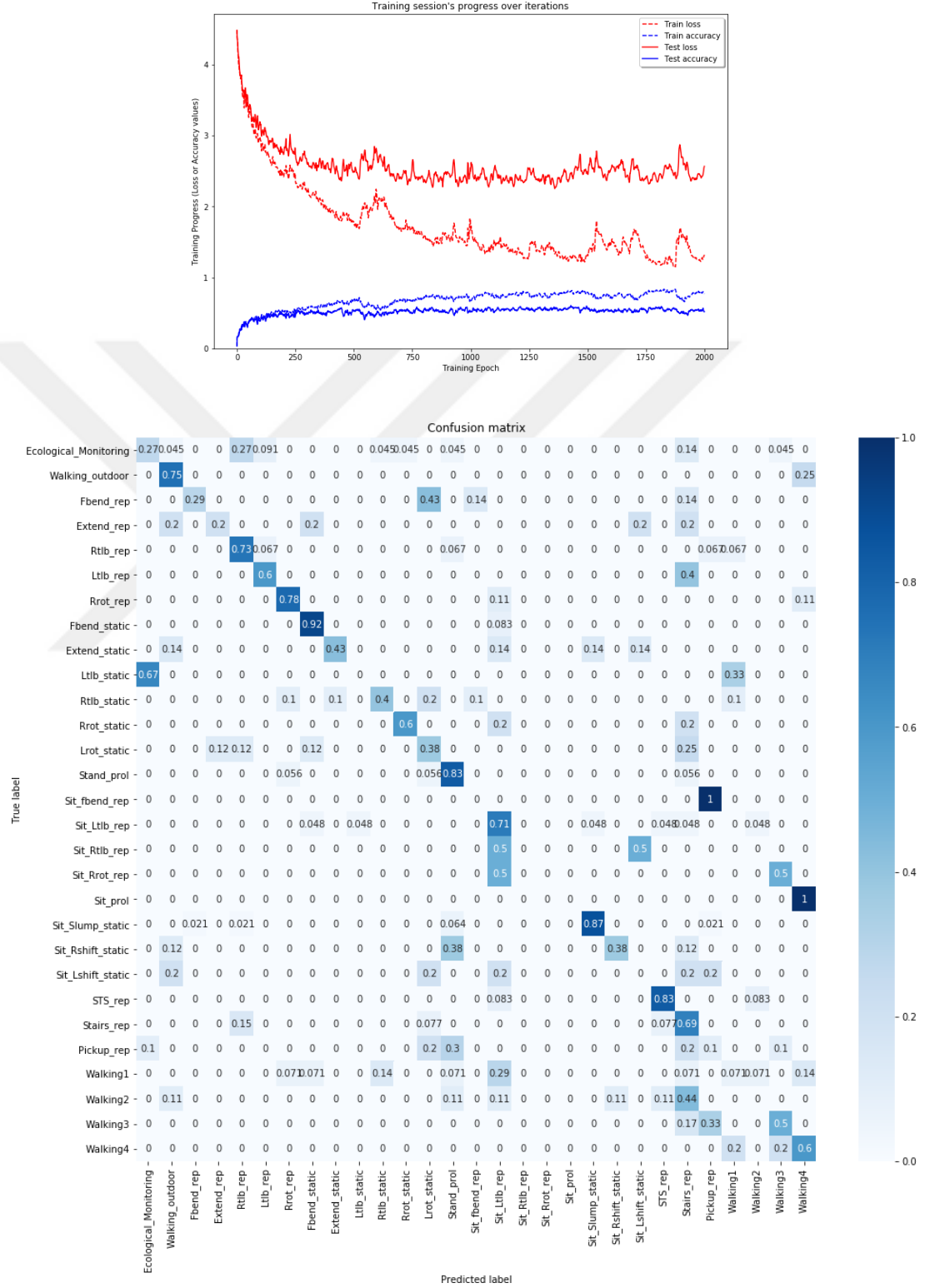
Şekil 3.6: LSTM katman sayısı 2 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi

2. testte LSTM katman sayısı arttırılarak 4 olarak belirlenmiştir. Test sonucunda, test doğruluğu (accuracy) yaklaşık 0.5486 (%54.9) ve testteki kayıp (loss) ise 2.724 olarak hesaplanmıştır.



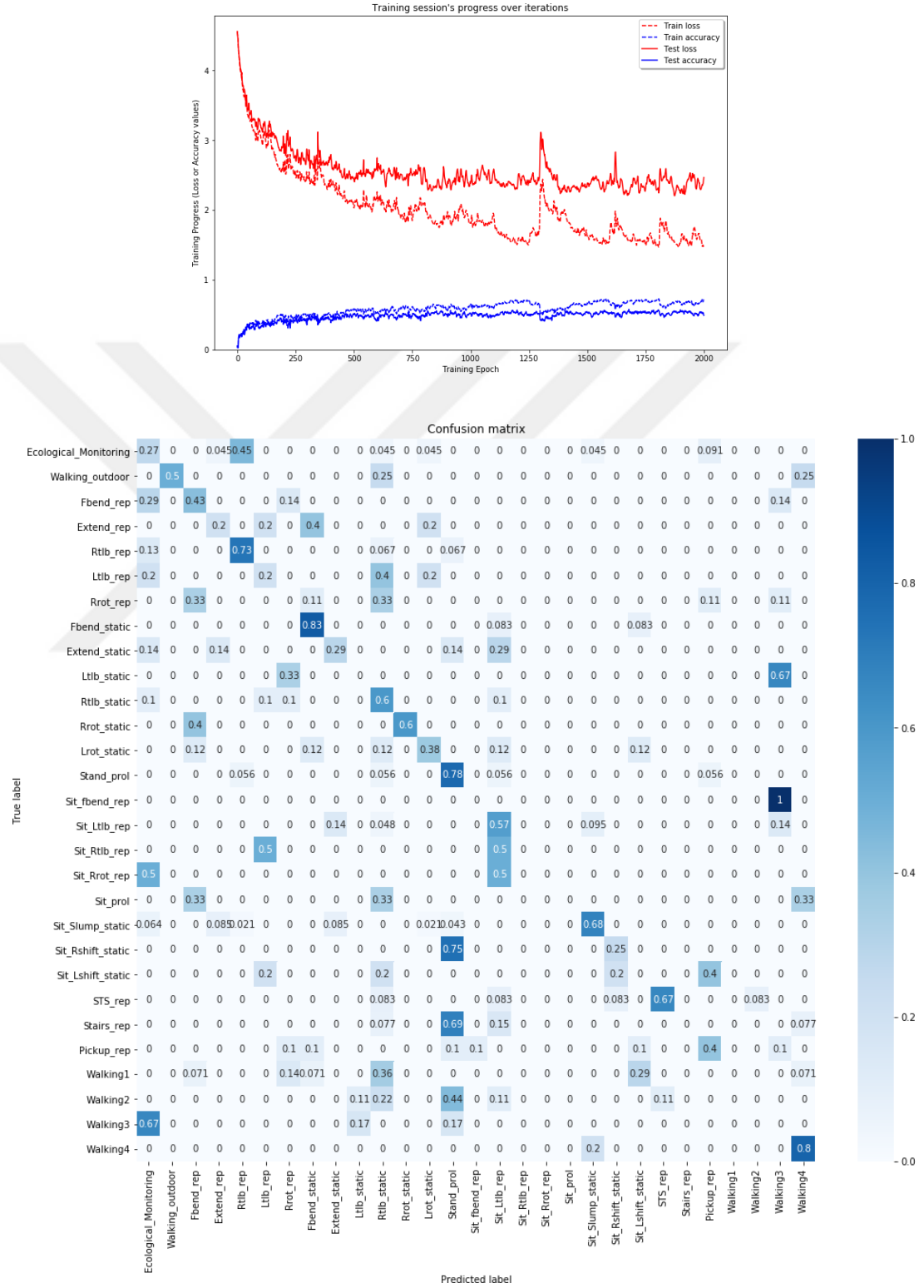
Şekil 3.7: LSTM katman sayısı 4 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi

3. testte LSTM katman sayısı artırılarak 6 olarak belirlenmiştir. Test sonucunda, test doğruluğu (accuracy) yaklaşık 0.5174 (%51.7) ve testteki kayıp (loss) ise 2.566 olarak hesaplanmıştır.



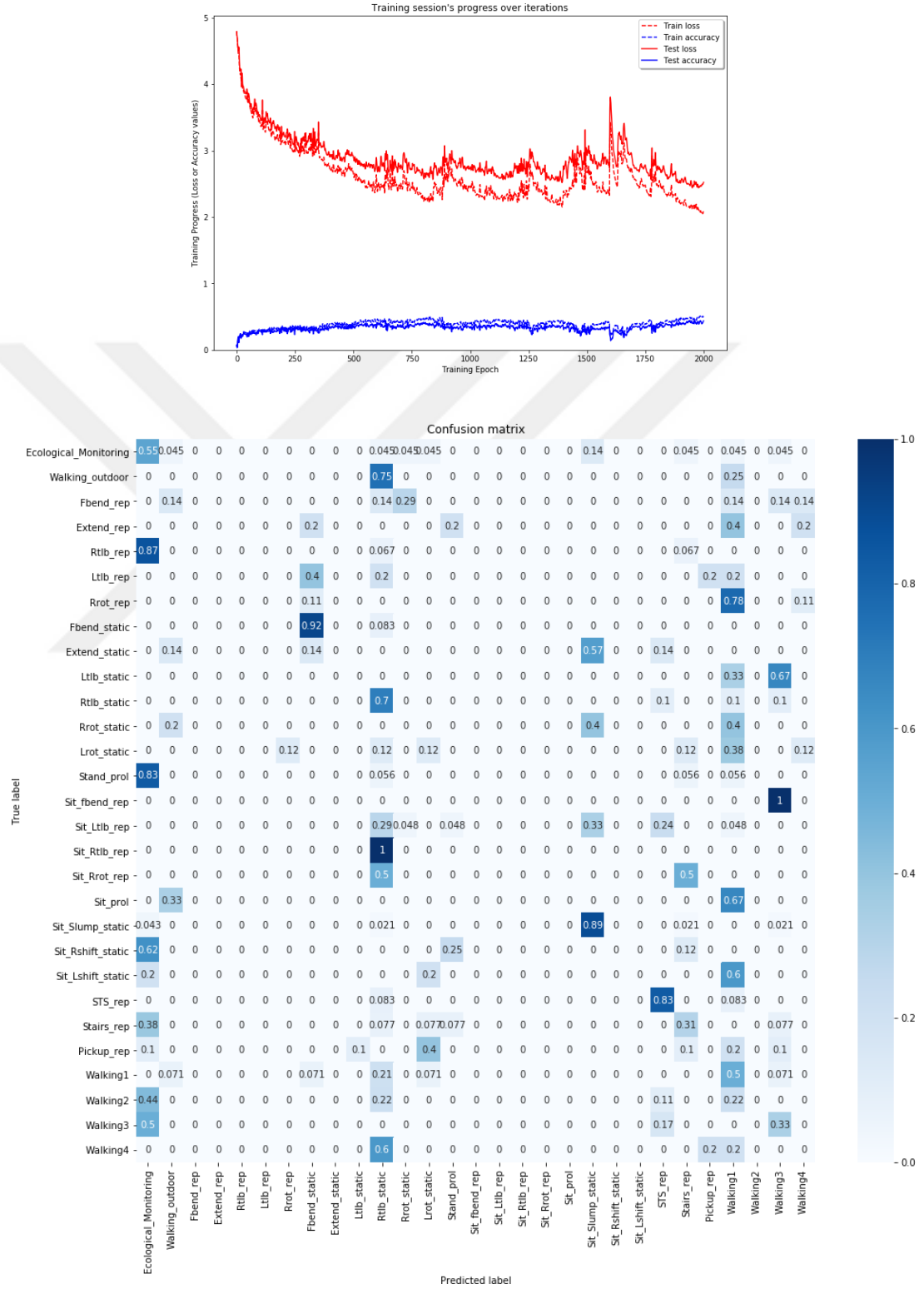
Şekil 3.8: LSTM katman sayısı 6 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi

4. testte LSTM katman sayısı artırılarak 8 olarak belirlenmiştir. Test sonucunda, test doğruluğu (accuracy) yaklaşık 0.4896 (%48.9) ve testteki kayıp (loss) ise 2.467 olarak hesaplanmıştır.



Şekil 3.9: LSTM katman sayısı 8 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi

5. testte LSTM katman sayısı artırılarak 10 olarak belirlenmiştir. Test sonucunda, test doğruluğu (accuracy) yaklaşık 0.4375 (%43.8) ve testteki kayıp (loss) ise 2.524 olarak hesaplanmıştır.



Şekil 3.10: LSTM katman sayısı 10 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi

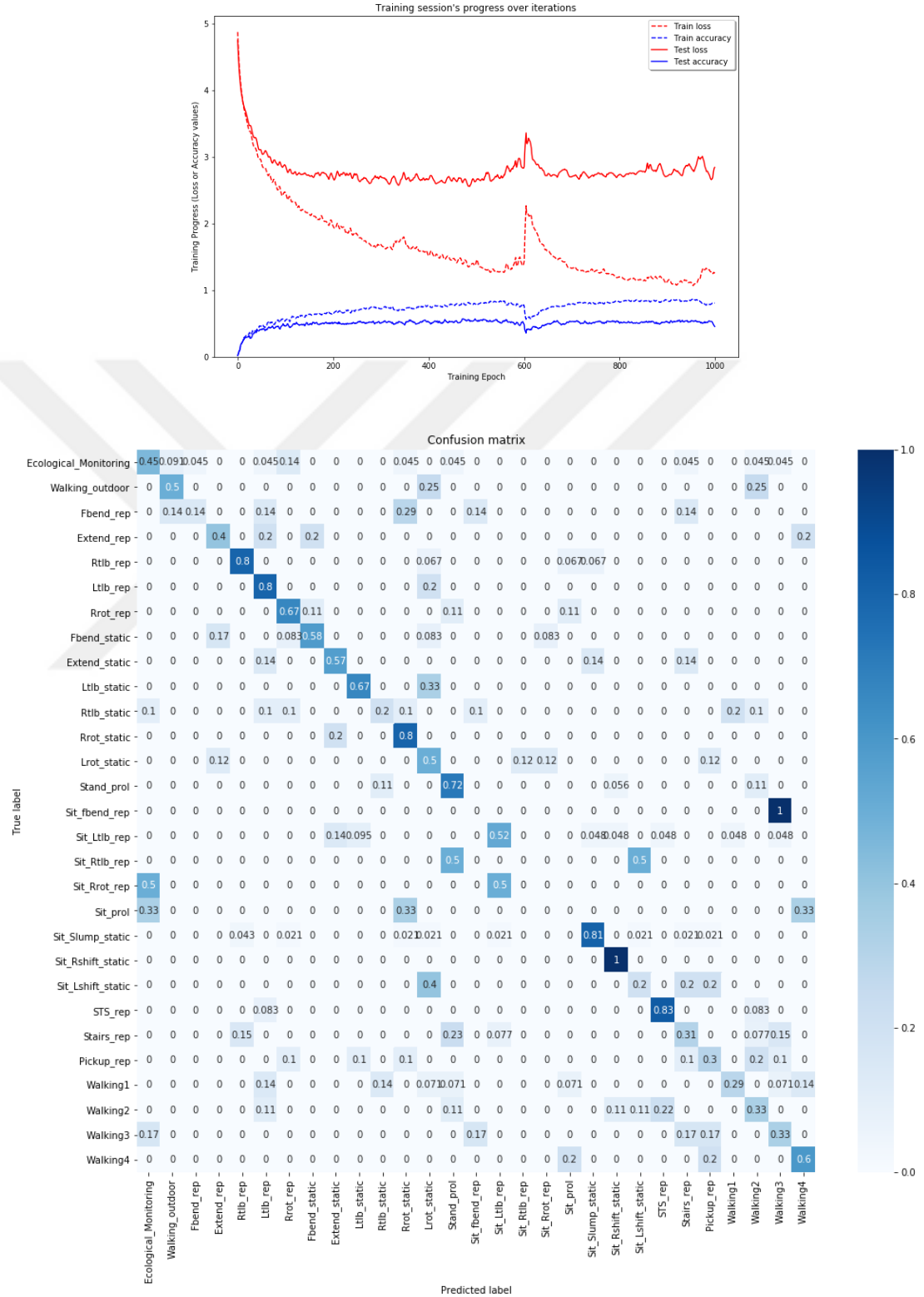
3.1.6.2 Eğitim Turu (Epoch) Sayısının Karşılaştırılması

Bu testlerde yığın boyutu (batch_size) 32 olarak sabit tutulacak, LSTM katman sayısı ise; en iyi sonuçlar 2 katmanlı LSTM modelinde alındığı için 2 olarak belirlenecek ve diğer testlerde eğitim turu sayısı arttırılacaktır. Yine tüm pozisyonlar kullanılacak, modeldeki sınıf sayısı (N_CLASSES) 29 olacaktır. Bu sayede eğitim turu (epoch) sayısının, doğruluk ve kayıp üzerindeki etkisini görmek planlanmaktadır.

Eğitim turu sayısının doğruluk ve kayıp üzerindeki etkisini görmek için;

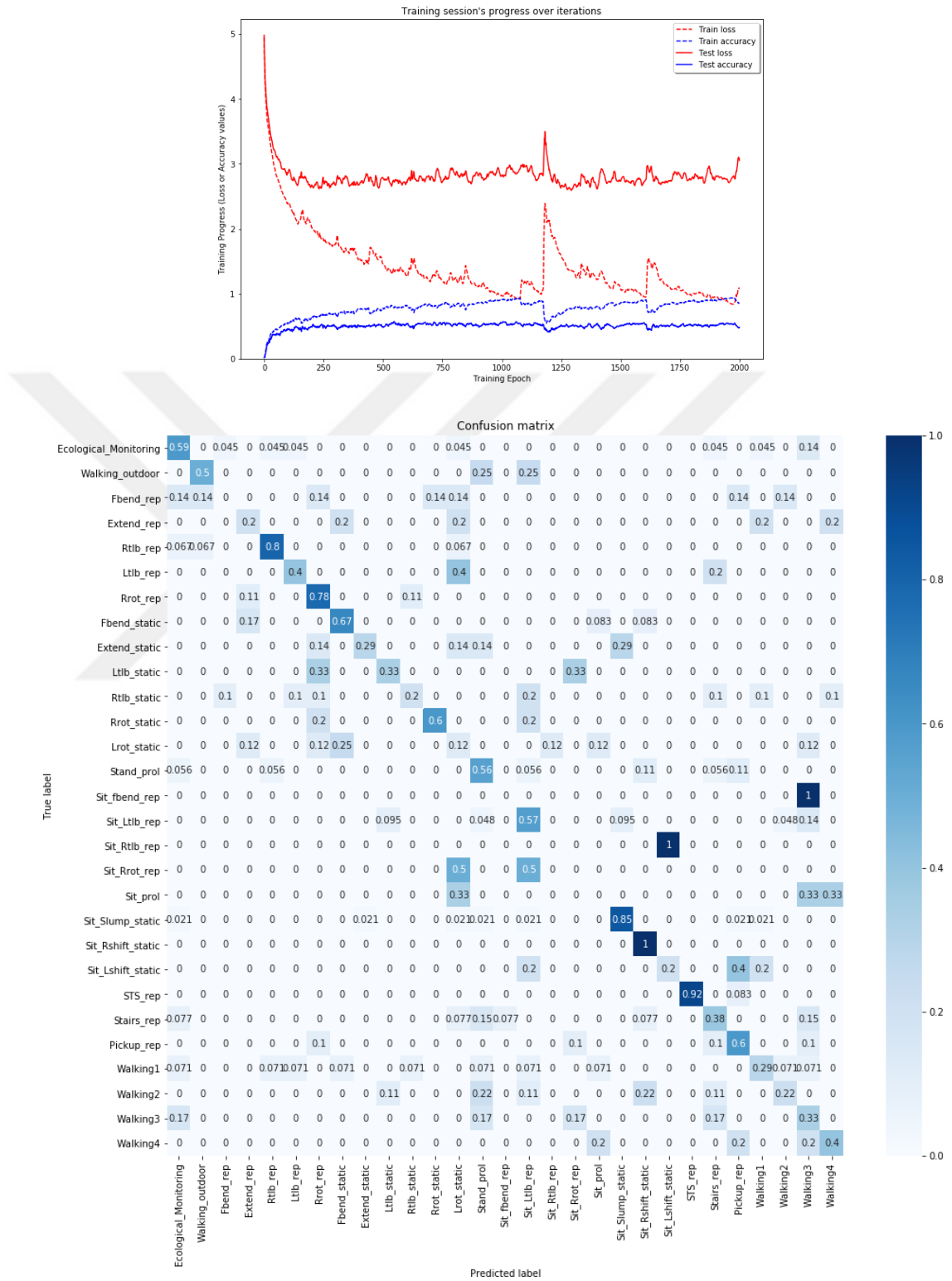


1. testte eğitim turu sayısı ilk olarak 50 olarak belirlenmiştir. Test sonucunda, test doğruluğu (accuracy) yaklaşık 0.4549 (%45.4) ve testteki kayıp (loss) ise 2.843 olarak hesaplanmıştır.



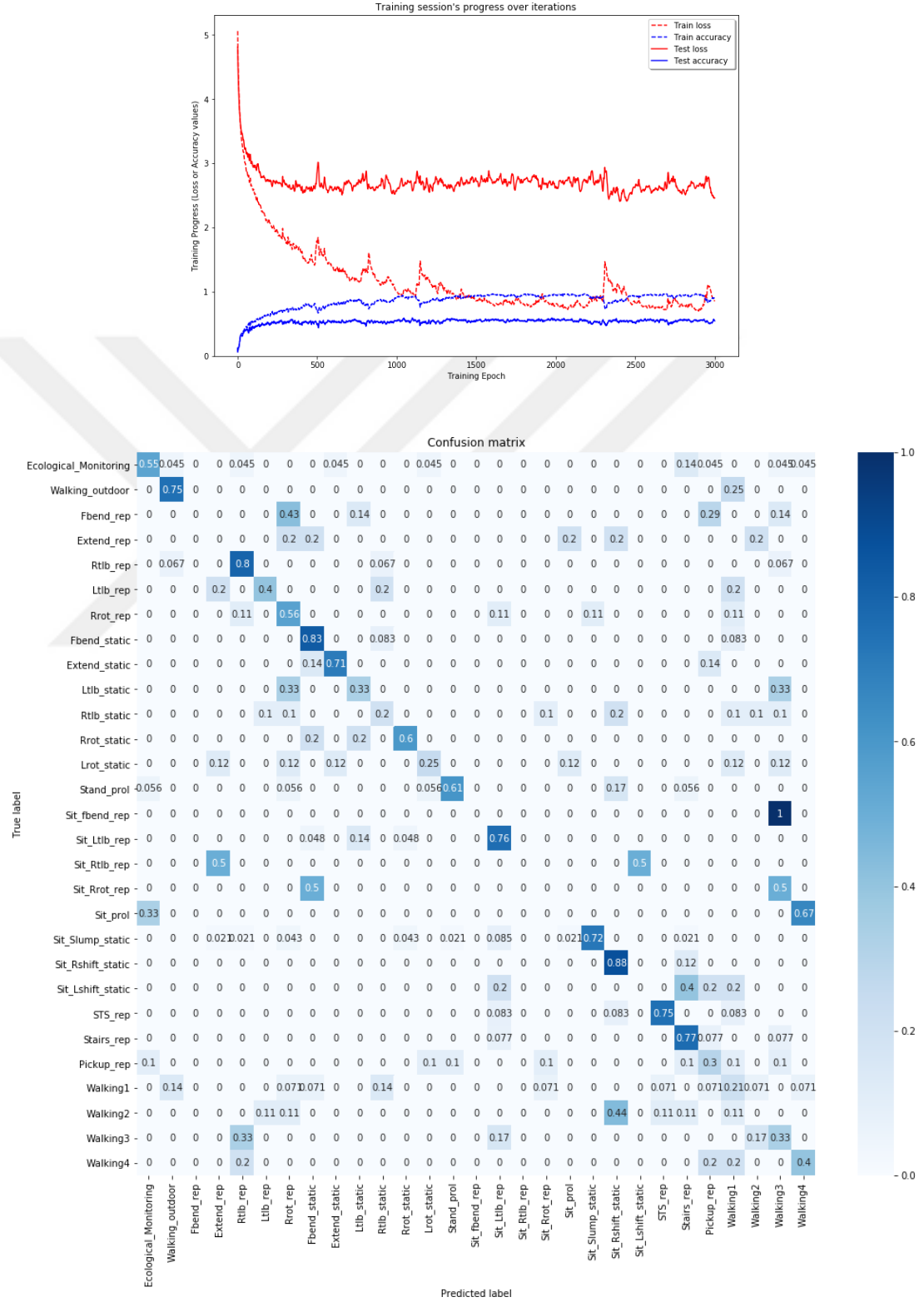
Şekil 3.11: Eğitim turu 50 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi

2. testte eğitim turu sayısı artırılarak 100 olarak belirlenmiştir. Test sonucunda, test doğruluğu (accuracy yaklaşık 0.4826 (%48.2) ve testteki kayıp (loss) ise 3.054 olarak hesaplanmıştır.



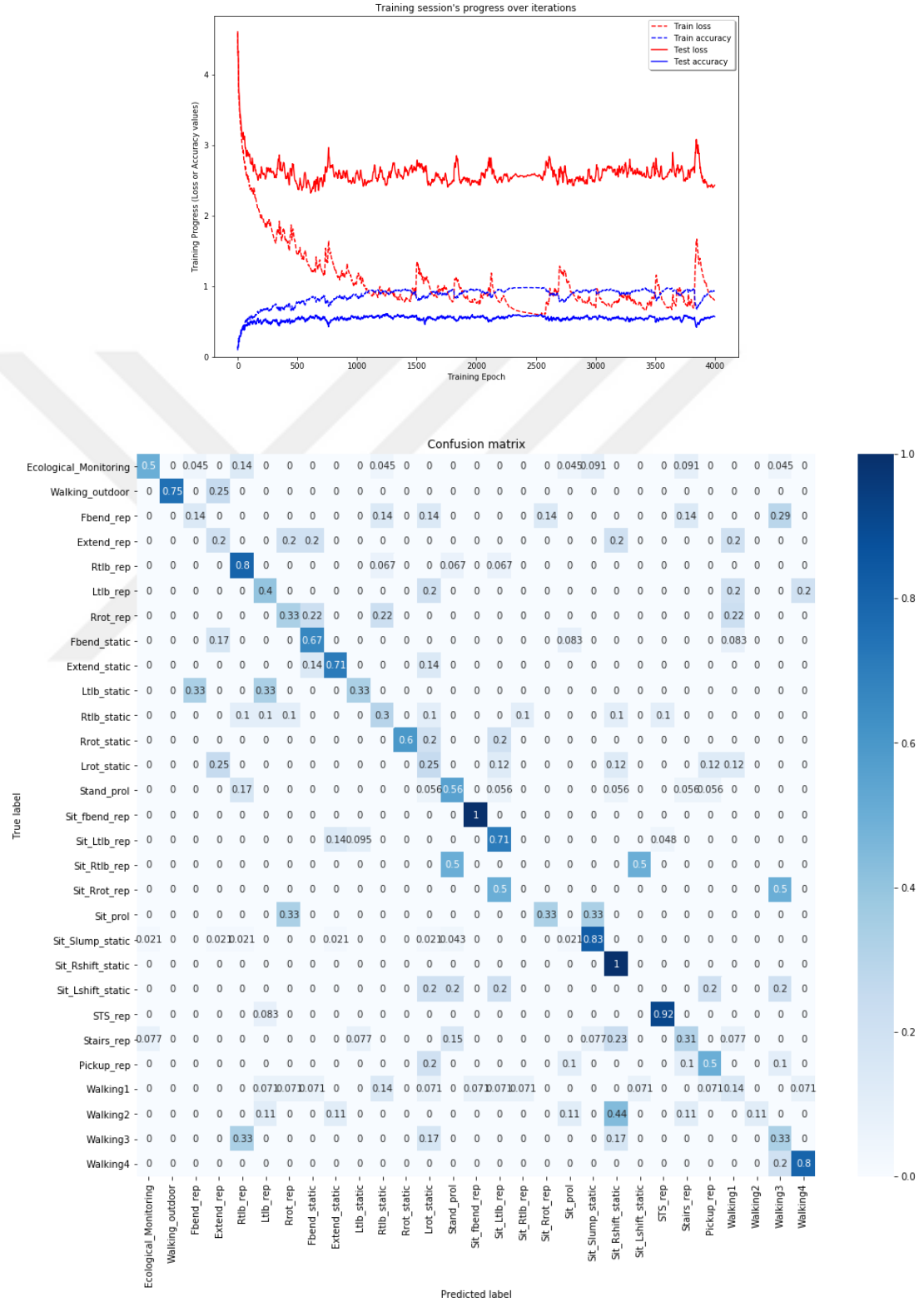
Şekil 3.12: Eğitim turu 100 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi

3. testte eğitim turu sayısı artırılarak 150 olarak belirlenmiştir. Test sonucunda, test doğruluğu (accuracy) yaklaşık 0.5451 (%54.5) ve testteki kayıp (loss) ise 2.460 olarak hesaplanmıştır.



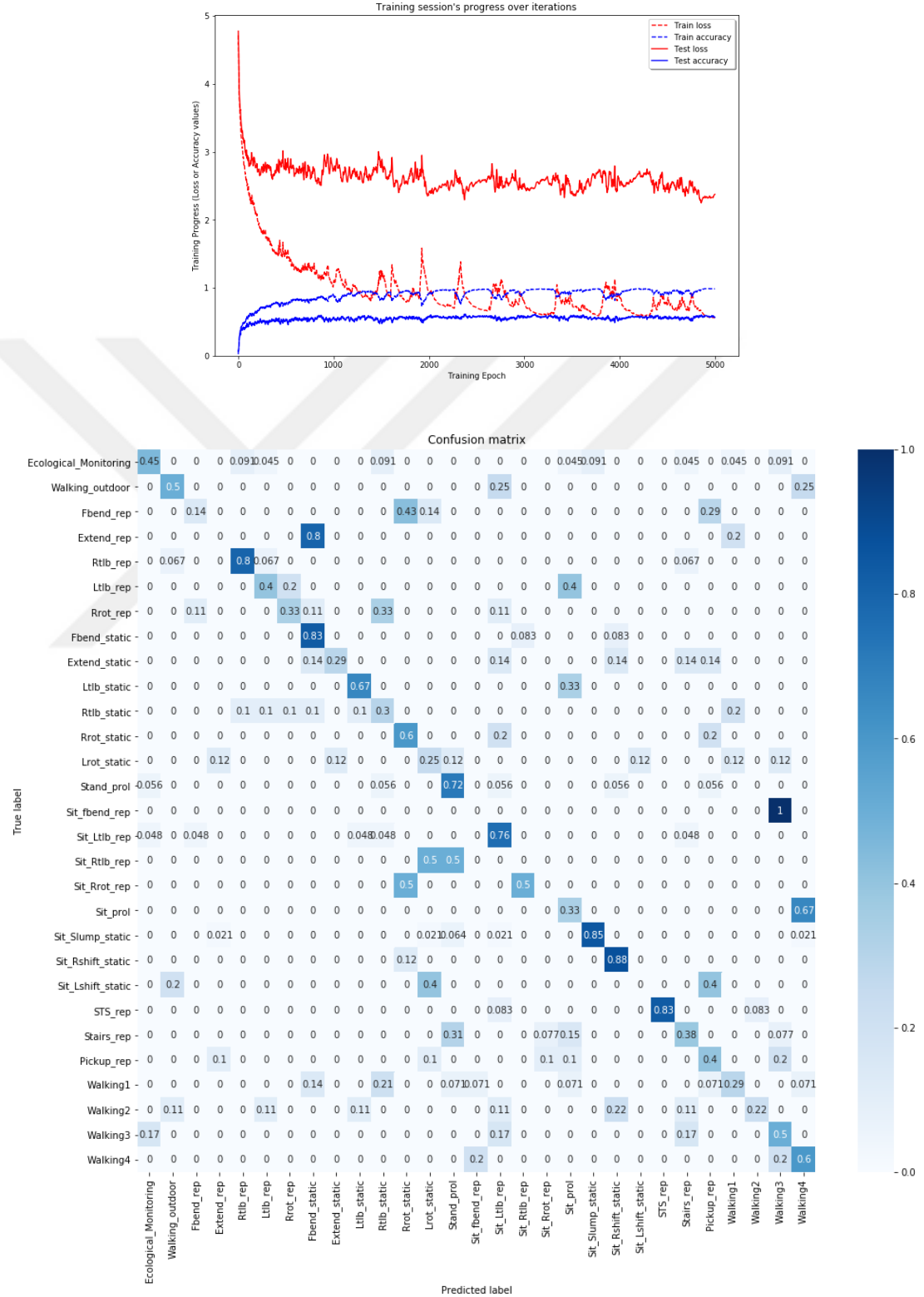
Şekil 3.13: Eğitim turu 150 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi

4. testte eğitim turu sayısı artırılarak 200 olarak belirlenmiştir. Test sonucunda, test doğruluğu (accuracy) yaklaşık 0.5659 (%56.6) ve testteki kayıp (loss) ise 2.377 olarak hesaplanmıştır.



Şekil 3.14: Eğitim turu 200 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi

5. testte eğitim turu sayısı artırılarak 250 olarak belirlenmiştir. Test sonucunda, test doğruluğu (accuracy) yaklaşık 0.5794 (%57.9) ve testteki kayıp (loss) ise 2.433 olarak hesaplanmıştır.



Şekil 3.15: Eğitim turu 250 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi

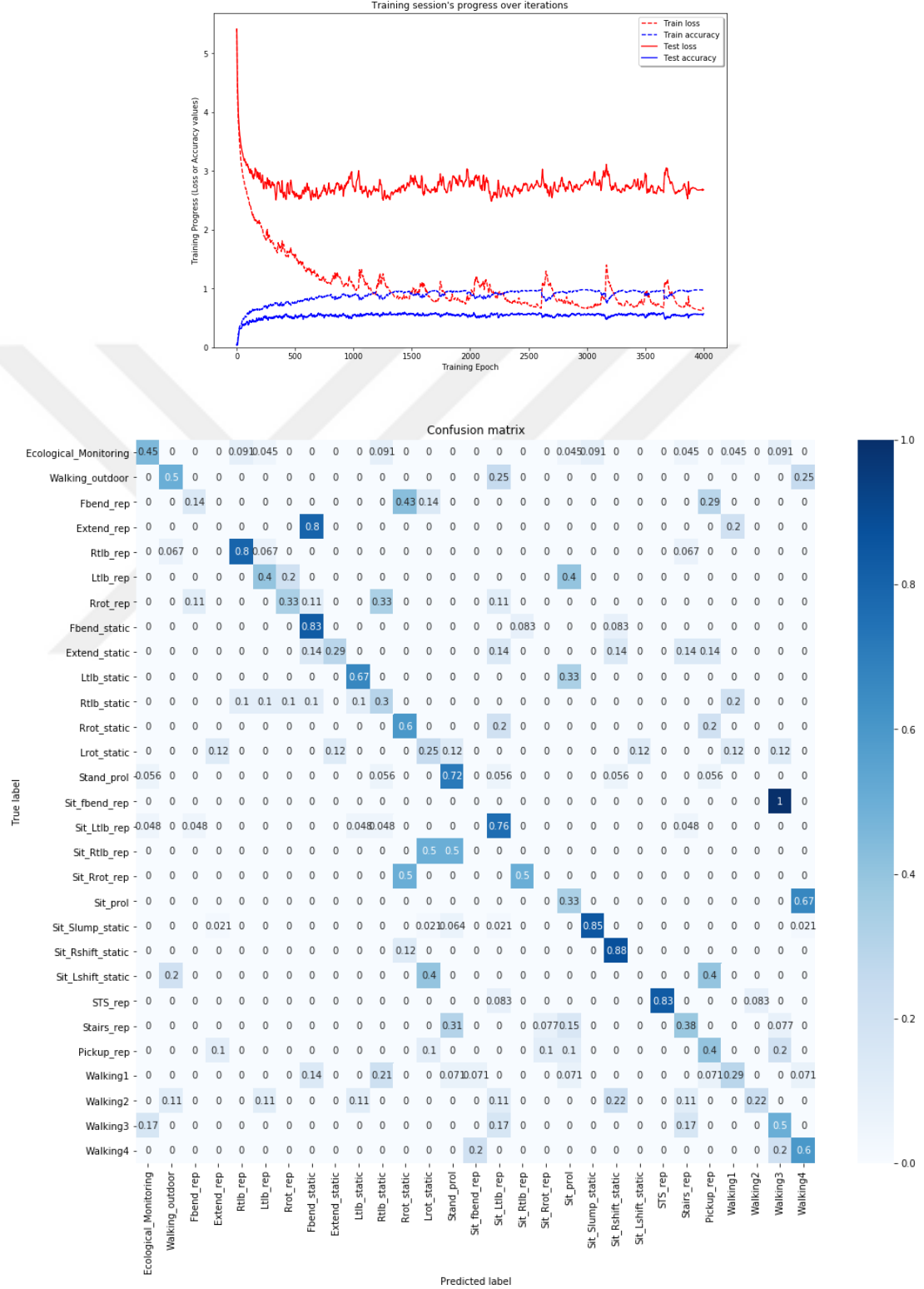
3.1.6.3 Yığın Boyutunun (Batch Size) Karşılaştırılması

Daha önceki testlerden elde edilen sonuca göre epoch sayısı 250 olduğunda en yüksek doğruluk değeri elde edildiği gözlemlenmiştir. Bu nedenle bu bölümdeki testlerde eğitim tur sayısı 250 olarak sabit tutulacak, LSTM katman sayısı ise; en iyi sonuçlar 2 katmanlı LSTM modelinde alındığı için 2 olarak belirlenecek ve diğer testlerde yığın boyutu arttırılacaktır. Yine tüm pozisyonlar kullanılacak, modeldeki sınıf sayısı (N_CLASSES) 29 olacaktır. Bu sayede yığın boyutunun, doğruluk ve kayıp üzerindeki etkisini görmek planlanmaktadır.

Yığın boyutunun doğruluk ve kayıp üzerindeki etkisini görmek için;

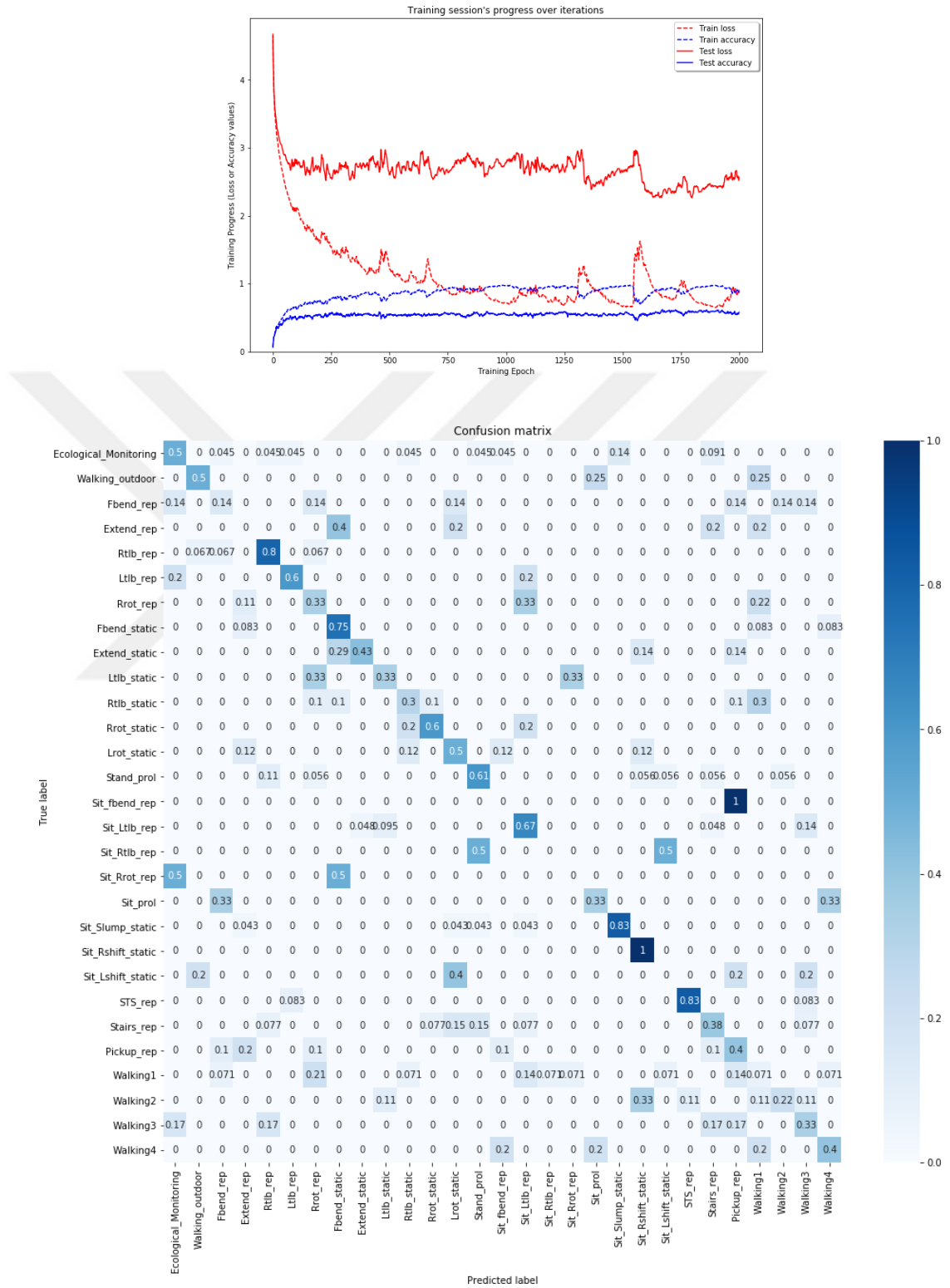


1. testte yığın boyutu ilk olarak 32 olarak belirlenmiştir. Test sonucunda, test doğruluğu (accuracy) yaklaşık 0.5818 (%58.2) ve testteki kayıp (loss) ise 2.564 olarak hesaplanmıştır.



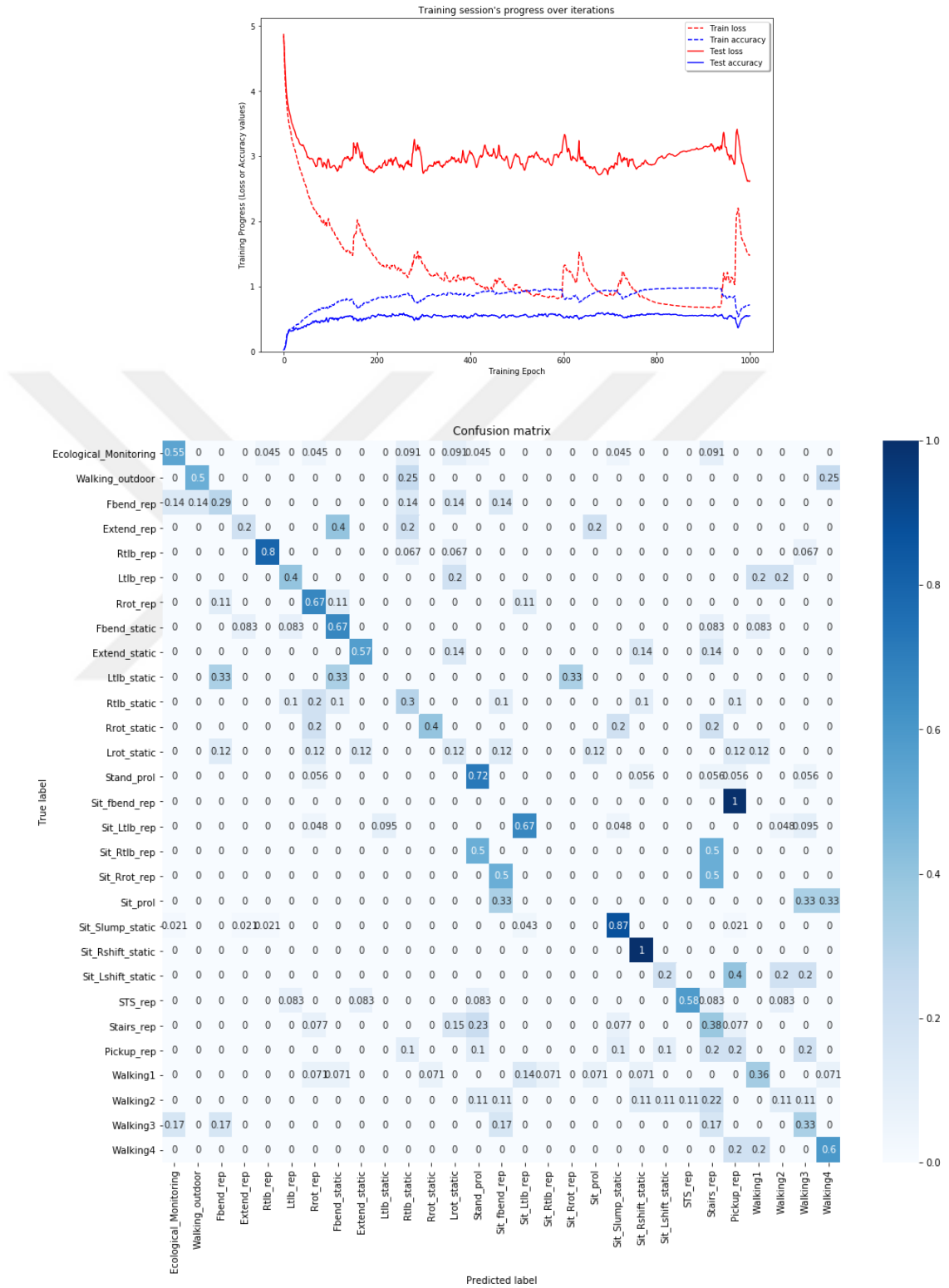
Şekil 3.16: Yığın boyutu 32 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi

2. testte yığın boyutu artırılarak 64 olarak belirlenmiştir. Test sonucunda, test doğru- luğu (accuracy) yaklaşık 0.5729 (%57.3) ve testteki kayıp (loss) ise 2.562 olarak hesaplanmıştır.



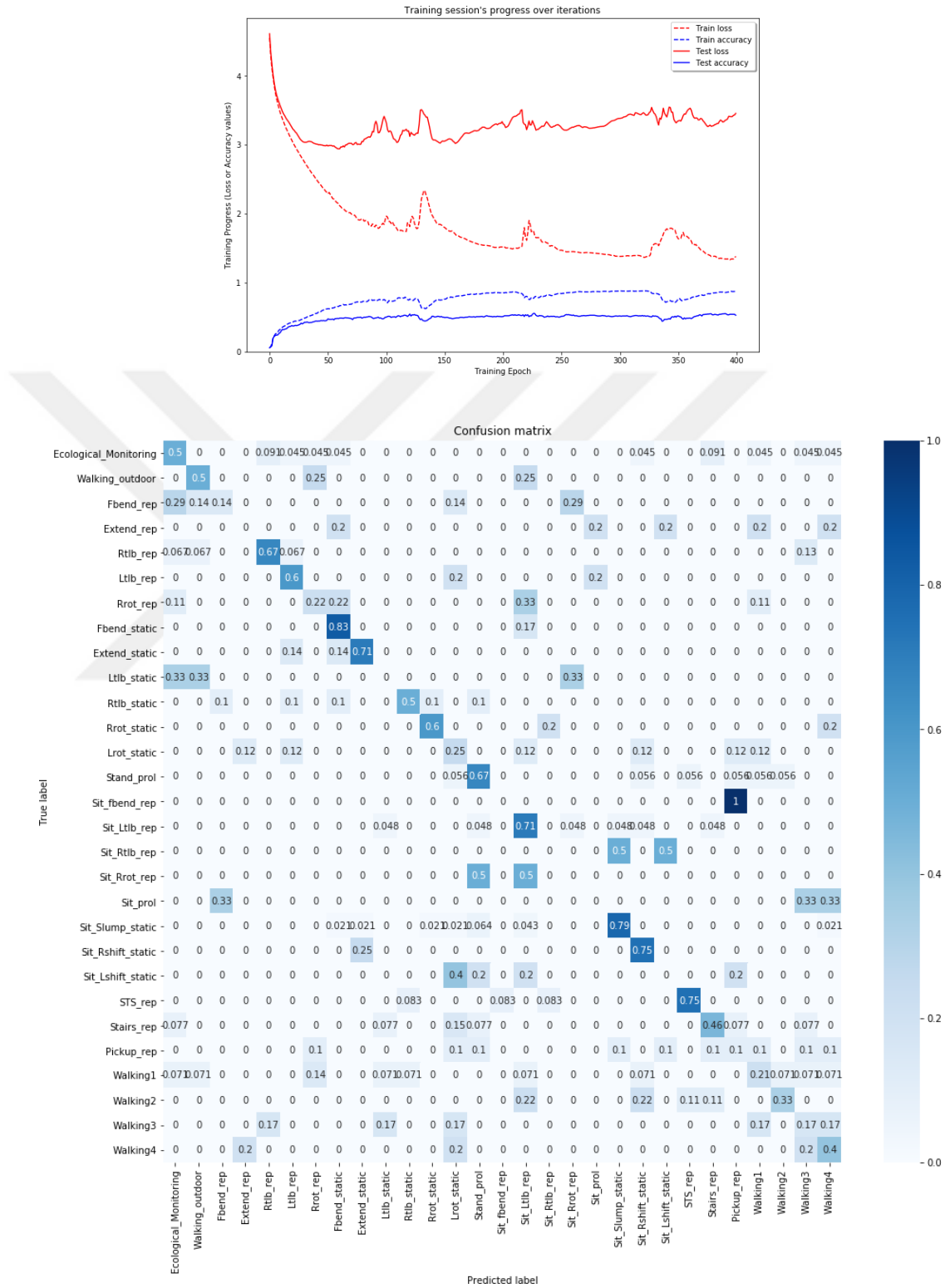
Şekil 3.17: Yığın boyutu 64 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi

3. testte yığın boyutu arttırılarak 128 olarak belirlenmiştir. Test sonucunda, test doğruluğu (accuracy) yaklaşık 0.5521 (%55.2) ve testteki kayıp (loss) ise 2.622 olarak hesaplanmıştır.



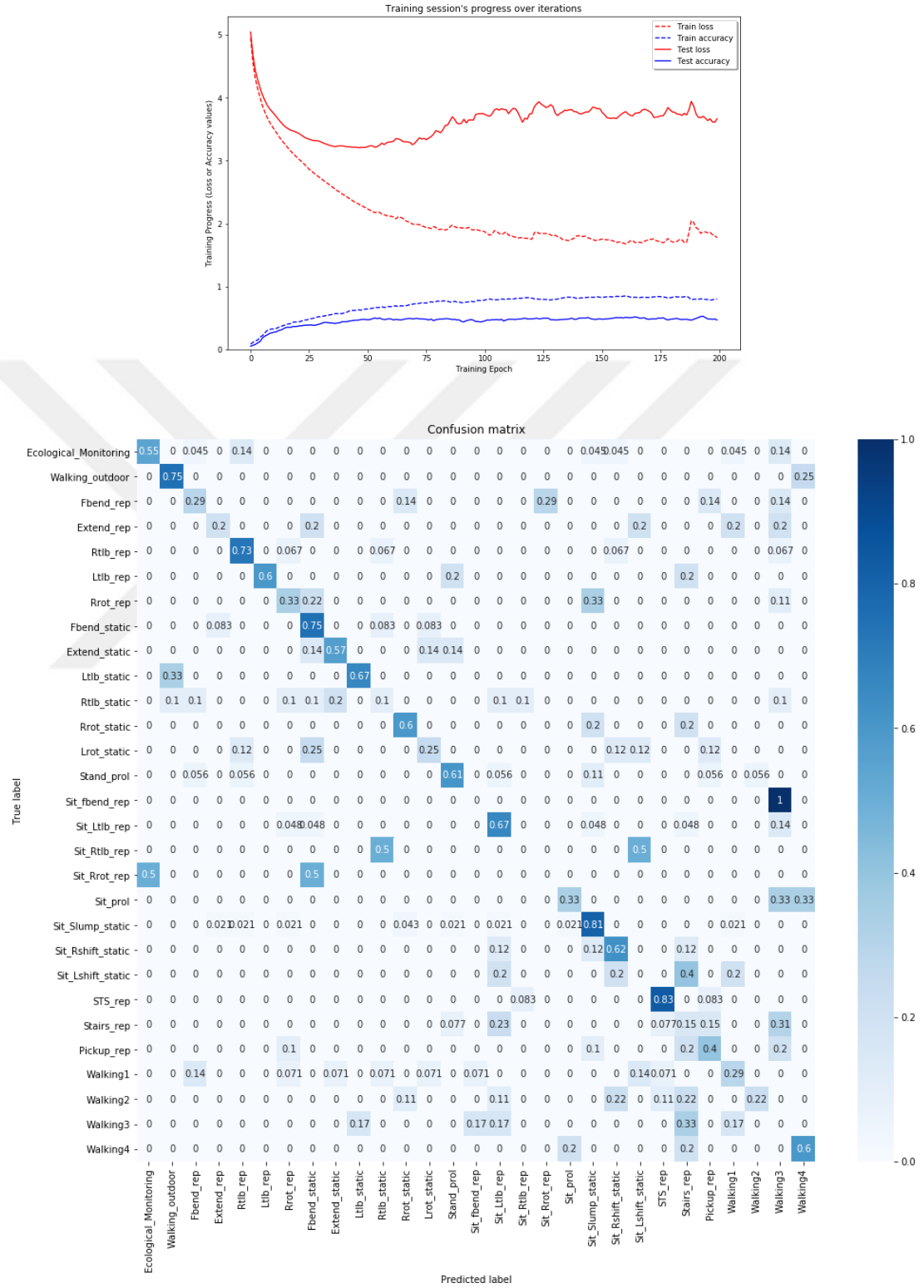
Şekil 3.18: Yığın boyutu 128 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi

4. testte yığın boyutu artırılarak 256 olarak belirlenmiştir. Test sonucunda, test doğruluğu (accuracy) yaklaşık 0.5277 (%52.8) ve testteki kayıp (loss) ise 3.456 olarak hesaplanmıştır.



Şekil 3.19: Yığın boyutu size 256 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi

5. testte yığın boyutu artırılarak 512 olarak belirlenmiştir. Test sonucunda, test doğruluğu (accuracy) yaklaşık 0.4887 (%48.9) ve testteki kayıp (loss) ise 3.667 olarak hesaplanmıştır.



Şekil 3.20: Yığın boyutu 512 olarak belirlenmiş metodun doğruluk grafiği ve karmaşıklık matrisi

3.1.7 Değerlendirme Kriterleri

Sınıflandırma sonuçlarını analiz etmek için dört farklı değerlendirme yöntemi kullanılmıştır.

1. **Doğruluk (Accuracy):** Bir ölçümün gerçek değere yakınlığını tanımlamak için kullanılır. Denklem 4.1'deki gibi hesaplanabilir.

$$Accuracy = \frac{TruePositives + TrueNegatives}{TruePositive + TrueNegatives + FalsePositives + FalseNegatives} \quad (3.1)$$

2. **Kesinlik/Hassasiyet (Precision):** Sınıflandırıcının, sezgisel olarak negatif olan bir örneği pozitif olarak etiketlememesi yeteneğidir. Denklem 4.2'deki gibi hesaplanabilir.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \quad (3.2)$$

3. **Hatırlama (Recall):** Sezgisel olarak sınıflandırıcının tüm pozitif örnekleri bulma kabiliyetidir. Denklem 4.3 ile hesaplanabilir.

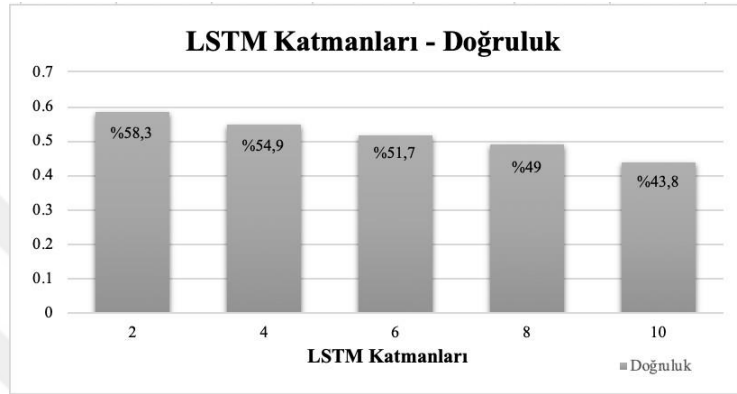
$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \quad (3.3)$$

4. **F-Measure/F-score (F1 score):** Hassasiyet ve hatırlamanın harmonik ortalamasıdır. Denklem 4.4 ile hesaplanabilir.

$$F1 = 2x \frac{Precision \times Recall}{Precision + Recall} \quad (3.4)$$

3.1.8 LSTM Sinir Ağı Eğitim Test Sonuçları

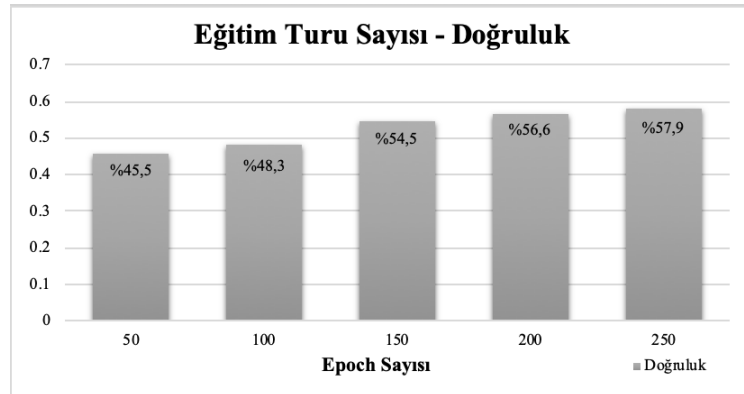
LSTM sinir ağı eğitim testleri 3 farklı hiperparametre değişimi incelenerek yapılmıştır. Bunlardan ilki LSTM sinir ağına kullanılacak katman sayısıdır. Katman sayısı başlangıç olarak 2 olarak belirlenmiş ve artırılarak modelin davranışı gözlemlenmiştir. LSTM sinir ağındaki train ve test verilerini, model %70 eğitim verisi, %30 test verisi olarak kendisi oluşturmuştur.



Şekil 3.21: LSTM katman sayısı - Doğruluk değişimi grafiği

Şekil 3.21'deki grafiğe baktığımızda çalışmamızda kullandığımız vücut kinematiği veri seti için LSTM katman sayısının artırılması modelin başarısını düşürmüş, doğruluk yüzdesi LSTM katman sayısı arttıkça azalmıştır.

Bir diğer hiperparametre ise eğitim turu sayısıdır (epoch sayısı). Eğitim turu sayısının modelin başarısı üzerindeki etkisini incelemek için öncelikle 50 eğitim turu sayısı ile başlanmış ve diğer testlerde artırılarak modelin davranışı gözlemlenmiştir.



Şekil 3.22: Eğitim Turu (Epoch) Sayısı - Doğruluk değişimi grafiği

Şekil 3.22'deki grafiği incelediğimizde çalışmamızda kullandığımız vücut kinematiği veri seti için eğitim turu sayısının artırılması modelin başarısını arttırmış, doğruluk yüzdesi eğitim turu sayısı arttıkça artmıştır. Derin öğrenme metodlarında eğitim turu sayısının yüksek olması modelin başarısını genel olarak olumlu etkilemektedir, ancak sürekli bir şekilde arttırmak her zaman modelin başarısını arttırmaz, belli bir sayıdan sonra düşüşler yaşanabilir.

Çalışmamızda işlediğimiz son derin öğrenme hiperparametresi ise yığın boyutudur. (batch_size) Yığın boyutunun GPU belleğine sığması gerektiği için yığın boyutu değeri 2 ve katları şeklinde (2, 4, 8, 16, . . . , 512) şeklinde belirlenmelidir. Bu şekilde belirlenmemişse modelin başarısında ani düşüşler yaşanabilir. Genellikle 64 ile 512 arasında değer almaktadır. Çalışmamızdaki testlerde 32'den başlayarak 2'nin katları şeklinde 512'ye kadar testler yapılarak modelin yığın boyutu artışı karşısında nasıl davrandığı gözlemlenecektir.



Şekil 3.23: Yığın Boyutu - Doğruluk değişimi grafiği

Şekil 3.23'deki grafiğe baktığımızda LSTM sinir ağını daha küçük boyutlarda yığın boyutuyla eğitmenin daha yüksek başarı sağladığı görülmektedir. yığın boyutu'nun çok büyük tanımlanması modelin eğitimini olumsuz etkilemektedir.

Tablo 3.1'de eğitim turu sayısının hassasiyet (precision), hatırlama (recall) ve F1-score değerleri üzerindeki etkisi incelenmiştir. Testlerde LSTM katman sayısı 2; yığın boyutu ise 32 olarak belirlenmiş ve sabit tutulmuş; eğitim turu sayısı ise her adımda 50'şer olarak arttırılmıştır. Test sonuçları incelendiğinde eğitim turuna bağlı olarak düzenli bir artış veya azalma görülmemektedir.

Tablo 3.1: Eğitim Turu Sayısına Göre Sınıflandırma Değerleri

Eğitim Turu Sayısı	F1-Score	Precision	Recall
50	0.4083	0.4371	0.4134
100	0.3875	0.3919	0.4159
150	0.4130	0.4348	0.4238
200	0.4071	0.4037	0.4331
250	0.4062	0.4234	0.4301

Tablo 3.2’de yığın boyutunun hassasiyet (precision), hatırlama (recall) ve F1-score değerleri üzerindeki etkisi incelenmiştir. Testlerde LSTM katman sayısı 2; eğitim turu sayısı 250 olarak belirlenmiş ve sabit tutulmuş; yığın boyutu sayısı ise her adımda 2 katına çıkarılmıştır. Test sonuçları incelendiğinde yığın boyutuna bağlı olarak düzenli bir artış veya azalma görülmemektedir.

Tablo 3.2: Yığın Boyutuna Göre Sınıflandırma Değerleri

Yığın Boyutu	F1-Score	Precision	Recall
32	0.4062	0.4234	0.4301
64	0.3266	0.3258	0.3498
128	0.3884	0.3941	0.4120
256	0.3249	0.3478	0.3354
512	0.3706	0.4091	0.3774

Tablo 3.3’de LSTM katman sayısının hassasiyet (precision), hatırlama (recall) ve F1-score değerleri üzerindeki etkisi incelenmiştir. Testlerde yığın boyutu 32; eğitim turu sayısı 100 olarak belirlenmiş ve sabit tutulmuş; LSTM katman sayısı ise her adımda 2’şer arttırılmıştır. Test sonuçları incelendiğinde LSTM katman sayısına bağlı olarak düzenli bir artış veya azalma görülmemektedir.

Tablo 3.3: LSTM Katman Sayısına Göre Sınıflandırma Değerleri

LSTM Katman Sayısı	F1-Score	Precision	Recall
2	0.3875	0.3919	0.4159
4	0.4154	0.4334	0.4411
6	0.4224	0.4339	0.4831
8	0.2862	0.3010	0.3290
10	0.1156	0.1141	0.1632



DÖRDÜNCÜ BÖLÜM

4.1 MAKİNE ÖĞRENMESİ TABANLI AKTİVİTE TANIMA

4.1.1 Makine Öğrenmesi Metodolojisi

Makine öğrenmesi, temel olarak denetimli (supervised) yani hedef değerleri belli olan ve denetimsiz (unsupervised) yani hedef değerleri belli olmayan olmak üzere iki grup öğrenme yöntemini içermektedir. Bu iki kavramın arasında yer alan ve etiketlenmemiş büyük miktarda bir veri ile etiketlenmiş küçük miktarda bir verinin beraber kullanılmasına ise yarı denetimli öğrenme denir.

Makine öğrenmesi ve veri madenciliği sık sık aynı yöntemleri kullanırlar ve bu yöntemler önemli ölçüde örtüşmektedir. Bu iki alanın ortak yönleri çok fazladır. Veri madenciliği birçok makine öğrenmesi metodunu kullanır fakat genellikle mantıksal olarak farklı hedefleri vardır. Diğer yandan makine öğrenmesi de denetimsiz öğrenme ya da öğrenici doğruluğunu geliştirmek için veri ön işleme adımı gibi veri madenciliği metodlarını kullanır.

Çalışmamızda da veri madenciliği yöntemleriyle makine öğrenmesi metodları birbirini tamamlayıcı olarak beraber kullanılmıştır.

4.1.2 Vücut Kinematiki Verileri Üzerinde Makine Öğrenmesi Uygulaması

Çalışmamızda kullanacağımız temel bazı Python kütüphaneleri vardır. Bunlar, sklearn (sci-kit learn) kütüphanesi, matplotlib kütüphanesi ve StatsModels kütüphanesidir.

Sklearn (scikit-learn) kütüphanesi: Makine öğrenmesi yöntemlerini kullanmamız için oluşturulmuş bir Python kütüphanesidir. Sklearn'den önce bilinmesi gereken Python kütüphaneleri de vardır. Bunlar veri seti üzerinde en iyi düzenlemelerimizi yapmamızı, veriyi anlamamızı ve veriyi Sklearn kütüphanesinde daha kolay işleyebilmemize yararlar. Bu kütüphanelerden bazıları derin öğrenme metodunda da kullandığımız Numpy, Pandas, Matplotlib, Seaborn, StatsModels kütüphaneleridir.

Bu bölümde derin öğrenme metodolojisinden farklı olarak Matplotlib ve StatsModels kütüphanelerini açıklanacaktır.

Matplotlib kütüphanesi: Veri görselleştirme yapabilmek için kullanılan

kütüphanedir.

StatsModels kütüphanesi: Birçok farklı istatistiksel modelin tahmininde, istatistiksel testlerin yapılmasında ve istatistiksel verilerin araştırılması için sınıflar ve işlevler sağlayan bir Python modülü veya kütüphanesi olarak tanımlanmaktadır.

- Sklearn ile Makine Öğrenmesinde verilerin;
- Ön işleme (Preprocesssing),
- Rasgele değişkenlerin sayısını azaltmak (Dimensionality Reduction),
- Parametre ve modellerin karşılaştırılması, doğrulanması ve seçilmesi (Model Selection),
- Sınıflandırılması (Classification),
- Tahmin sonuçları (Regression),
- Benzer özelliklerdeki nesnelerin kümelmesi (Clustering) gibi işlemler yapmamıza olanak sağlar.

Veri setindeki değerlerden tahminler oluşturmak için kullandığımız istatistiksel modeller şunlardır:

Regresyon Modelleri:

- 1. K-Nearest Neighbors Regression:** Yeni bir veri noktası hakkında tahminde bulunmak için, algoritma eğitim veri setindeki en yakın veri noktalarını bulur - “en yakın komşusu”.

```
from sklearn.neighbors import KNeighborsRegressor
```

- 2. Decision Tree Regression:** Karar ağacı, çok sayıda kayıt içeren bir veri kümesini, bir dizi karar kuralları uygulayarak daha küçük kümelere bölmek için kullanılan bir yapıdır. Basit karar verme adımları uygulanarak, büyük miktarlardaki kayıtları, çok küçük kayıt gruplarına bölerek kullanılan bir yapıdır.

```
from sklearn.tree import DecisionTreeRegressor
```

- 3. Random Forest Regression:** Tahminlerde bulunmak için, tüm ağaçların tahminlerini elde eder, daha sonra en çok oyu alan sınıfı tahmin eder. Karar Ağaçlarının böyle bir grubuna Rastgele Orman adı verilir ve sadeliğine rağmen, bugün mevcut en güçlü Makine Öğrenimi algoritmalarından biridir.

from sklearn.ensemble import RandomForestRegressor

- 4. Gradient Boosting Regression:** Gradient Boosting ilk iterasyonda tahminleri üreten bir “F” fonksiyonu oluşturur. Tahminler ile hedef değer arasındaki farkı hesaplar ve bu farklar için de “h” fonksiyonunu oluşturur. İkinci iterasyonda “F” ve “h” fonksiyonlarını birleştirir ve tekrar tahminler ile hedefler arasındaki fark hesaplar. Bu sayede sürekli üstüne ekleyerek “F” fonksiyonunun başarısını arttırmaya, dolayısıyla da tahminler ile hedefler arasındaki farkı sıfıra indirmeye çalışır.

from sklearn.ensemble import GradientBoostingRegressor

Elimizdeki veri setlerinin, hedef sınıflarını makinaya öğreterek, sınıfı hakkında bilgimiz olmayan verilerin sınıflarını (veri sınıflandırma işlemi) aşağıdaki istatistiksel modeller yardımıyla tamamlandı.

Sınıflandırma Modelleri:

- 1. K Nearest Neighbors Sınıflandırması:** KNN Sınıflandırıcısı, bir veri seti içerisindeki en yakın komşular yöntemiyle, bir K sayısı, yani sınıf sayısı belirtilerek veya belirlenerek bunların sınıflandırılması sağlar.

from sklearn.neighbors import KNeighborsClassifier

- 2. Naive Bayes Sınıflandırması:** Bu sınıflandırıcı, bir olayın meydana gelme ihtimalini hesaplarken başka bir olayın ortaya çıkışını göz önüne alarak hesaplama yapar. Bir değişkenin sonuçlarını, yani bir hipotezin (h) gerçek olma olasılığını hesaplarken önceden edinilen bilgiler (d) de kullanılarak sınıflandırma işlemi gerçekleşir.

from sklearn.naive_bayes import GaussianNB

- 3. Decision Tree (Karar Ağacı) Sınıflandırması:** Karar ağacı yöntemiyle belirli sorular sorar ve bunların sonucunda sınıflandırma yapar.

from sklearn.tree import DecisionTreeClassifier

- 4. Random Forest Sınıflandırması:** Birden çok karar ağacı kullanılarak sınıflandırmalar yapar.

from sklearn.ensemble import RandomForestClassifier

- 5. Gradient Boosting Sınıflandırması:** Regression modelleri kısmında Gradient Boosting (Gradyan Arttırma)‘den bahsetmiştik. Kısacası, kayıp

fonksiyonlar yardımıyla eğimleri kullanarak eksiklikleri belirlemeye çalışan sınıflandırma yöntemidir.

```
from sklearn.ensemble import GradientBoostingClassifier
```

- 6. AdaBoost Sınıflandırması:** Bu sınıflandırma yöntemi başlangıç olarak rastgele bir eğitim kümesi seçer daha sonra her adımda son eğitimin doğru tahminine dayanarak yeni eğitim setini seçerek AdaBoost makine öğrenme modelini tekrar eder. Verileri hatasız bir şekilde uyuncaya veya belirtilen maksimum tahmin ediciye ulaşılan kadar çalışmasını sürdürür.

```
from sklearn.ensemble import AdaBoostClassifier
```

4.1.3 Değerlendirme Kriterleri

Sınıflandırma sonuçlarını analiz etmek için dört farklı değerlendirme yöntemi kullanılmıştır.

- 1. Doğruluk (Accuracy):** Bir ölçümün gerçek değere yakınlığını tanımlamak için kullanılır. Denklem 4.1'deki gibi hesaplanabilir.

$$Accuracy = \frac{TruePositives + TrueNegatives}{TruePositive + TrueNegatives + FalsePositives + FalseNegatives} \quad (3.1)$$

- 2. Kesinlik/Hassasiyet (Precision):** Sınıflandırıcının, sezgisel olarak negatif olan bir örneği pozitif olarak etiketlememesi yeteneğidir. Denklem 4.2'deki gibi hesaplanabilir.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \quad (3.2)$$

- 3. Hatırlama (Recall):** Sezgisel olarak sınıflandırıcının tüm pozitif örnekleri bulma kabiliyetidir. Denklem 4.3 ile hesaplanabilir.

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \quad (3.3)$$

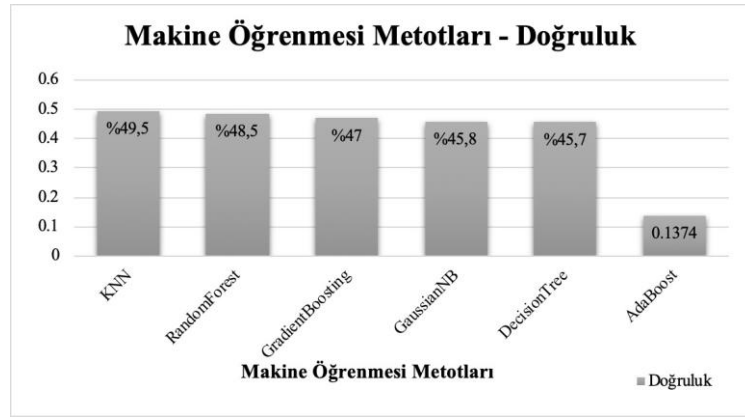
4. **F-Measure/F-score (F1 score):** Hassasiyet ve hatırlamanın harmonik ortalamasıdır. Denklem 4.4 ile hesaplanabilir.

$$F1 = 2x \frac{Precision \times Recall}{Precision + Recall} \quad (3.4)$$

4.1.4 Makine Öğrenmesi Test Sonuçları

Çalışmamızda denetimli makine öğrenme metotlarından DecisionTree, RandomForest, GradientBoosting, AdaBoost, KNN, GaussianNaiveBayes metotları üzerinde vücut kinematiği verileri test edilerek modellerin başarıları karşılaştırılmıştır.

Bu bölümdeki metotlarda kullanılacak eğitim ve test veri setleri %70 - %30 oranında hazırlanarak test edilmiştir. Şekil 4.1’de görüldüğü gibi makine öğrenmesi metotlarından doğruluk yüzdeleri arasında genelde çok büyük farklılıklar gözlemlenmesine karşın en yüksek doğruluk oranı KNN metoduyla elde edilmiştir. Bununla birlikte AdaBoost metodunun oldukça düşük bir doğruluk yüzdesine sahip olduğu tespit edilip, vücut kinematiği izleme verisi için uygun metot olmadığı görülmüştür.



Şekil 4.1: Makine Öğrenmesi Metotları - Doğruluk grafiği

Tablo 4.1’de farklı makine öğrenmesi metotlarının hassasiyet (precision), hatırlama (recall) ve F1-score değerleri yer almaktadır. Değerler, 6 farklı denetimli makine öğrenmesi için hesaplanmış olup bu metotlar sırasıyla; DecisionTree, RandomForest, GradientBoosting, AdaBoost, Gaussian Naive Bayes ve K-Nearest Neighbour metotlarıdır.

Tablo 4.1: Farklı Makine Öğrenmesi Metotlarına Göre Sınıflandırma Değerleri

Metot	F1-Score	Precision	Recall
DecisionTree	0.2817	0.3256	0.2960
RandomForest	0.3332	0.3683	0.3444
GradientBoosting	0.3105	0.3430	0.3230
AdaBoost	0.0782	0.1002	0.0967
Gaussian Naive Bayes	0.2897	0.2840	0.3149
K-Nearest Neighbour	0.3360	0.3882	0.3500



BEŞİNCİ BÖLÜM

5.1 SONUÇLAR VE ÖNERİLER

Bu çalışmada insan aktivite tanıma problemi için kullanılabilecek derin öğrenme modeli ile makine öğrenmesi metotlarının sonuçları karşılaştırılmıştır. Veri analitiği, insan vücuduna direkt olarak yerleştirilen kablosuz iki vücut sensörleri aracılığıyla yapılmıştır. İki farklı sensörden ardışık sırayla gelmesi beklenen verilerden; bazı olumsuz faktörlerden dolayı istenilen sırayla gelmeyen sensör verileri, belli bir kurala göre eşleştirilmiştir. Kuaterniyon olarak kaydedilen vücut kinematiği verilerindeki hareket açılarını hesaplamak için, değerler öncelikle rotasyon matrisine sonra da Euler açılarına dönüştürülmüştür.

Euler açı dönüşümünü tamamladığımız veri seti görselleştirilmiş, görselleştirilen veri setinde gürültüler ve bazı hatalı uç değerler olduğu farkedilmiştir. Bu sebeple verideki gürültüyü gidermek için medyan filtre kullanılmıştır. Medyan filtre sayesinde, verideki uç değerler ve gürültüler filtrelenmiştir. Gürültülü ham veri ile filtrelenmiş veri sinyallerinin görselleri karşılaştırılarak, medyan filtrenin etkisi gösterilmiştir.

Pozisyonların dağılımını farklı bir şekilde görmek için 3 boyutlu düzlemde çizilmiş ve bu çizimler pozisyonlara ayrılarak kategorize edilmiştir.

Veri seti üzerinde denediğimiz bir diğer algoritma ise k-means kümeleme algoritmasıdır. K-means algoritması için küme sayısını önceden belirleyebildiğimiz gibi; küme sayısı belli olmayan veri setleri için en uygun küme sayısını bulabileceğimiz algoritmalar test edilerek, en iyi sonuç veren algoritma belirlenmiştir.

Veri üzerinde temizleme, açı değerleri hesaplama, eşleştirme, ve filtreleme işlemlerini tamamladıktan sonra, medyan filtresi kullanarak filtrelediğimiz vücut kinematiği verileri üzerinde yeni makine öğrenmesi yöntemlerinden olan derin öğrenme metodu ile klasik denetimli makine öğrenmesi metotlarını uygulanarak, en iyi metot veya metotların doğrulukları (accuracy), hassasiyetleri (precision), geri çağırma değerleri (recall) ve f1-score'ları analiz edilmiştir.. Makine öğrenmesi metotları ve derin öğrenme metodu kıyaslamasının dışında, derin öğrenme metodunu eğitmek için kullanılan hiper-parametrelerin (eğitim turu sayısı, yığın boyutu, LSTM katman sayısı gibi) metodun doğruluklarını ve diğer sınıflandırma

kriterlerini nasıl etkilediği ile ilgili testler yapılarak sonuçlar analiz edilmiştir.

Sonuç olarak çalışmamızda kullandığımız denetimli makine öğrenmesi metotları (Decision Tree, Random Forest, Gaussian Naive Bayes, AdaBoost, GradientBoosting, KNN) ve LSTM sinir ağı eğitilerek oluşturulmuş derin öğrenme metodu başarı sonuçları karşılaştırıldığında en yüksek başarımın, derin öğrenme metoduyla elde edildiği görülmüştür. LSTM sinir ağının da kendi içinde en başarılı olduğu hiper parametrelerde katman sayısının 2, yığın boyutunun 32 ve eğitim turu sayısının 250 olduğu görülmüştür.

İnsan aktiviteleri sınıflandırmada başarısının oldukça yüksek olduğu derin öğrenme metotlarının, gelecekte vücut kinematiği verilerinin sınıflandırılmasında sıklıkla kullanılacağını düşünüyoruz. Aktivite sınıflandırmanın yanı sıra aktivitedeki ani ve beklenmedik değişim ve anormalliklerin de analiz edilebileceğini düşünüyoruz. Uzman kişilerle birlikte yapılacak bu analizler sayesinde, hasta ve yaşlı kişilerin sağlıklarını korumada ve tedavilerinde kolaylık sağlanacağını planlamaktayız.

REFERANSLAR

- Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., & Savarese, S. (2016). Social Istm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 961-971).
- Alsheikh, M. A., Selim, A., Niyato, D., Doyle, L., Lin, S., & Tan, H. P. (2016, March). Deep activity recognition models with triaxial accelerometers. In *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*.
- Chen, Y., & Xue, Y. (2015, October). A deep learning approach to human activity recognition based on single accelerometer. In *2015 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 1488-1492). IEEE.
- Gaidhani, A., Moon, K., Ozturk, Y., Lee, S., & Youm, W. (2017). Extraction and Analysis of Respiratory Motion Using Wearable Inertial Sensor System during Trunk Motion. *Sensors*, *17*(12), 2932.
- Hammerla, N. Y., Halloran, S., & Plötz, T. (2016). Deep, convolutional, and recurrent models for human activity recognition using wearables. *arXiv preprint arXiv:1604.08880*.
- Hassan, M. M., Uddin, M. Z., Mohamed, A., & Almogren, A. (2018). A robust human activity recognition system using smartphone sensors and deep learning. *Future Generation Computer Systems*, *81*, 307-313.
- Jiang, W., & Yin, Z. (2015, October). Human activity recognition using wearable sensors by deep convolutional neural networks. In *Proceedings of the 23rd ACM international conference on Multimedia* (pp. 1307-1310). Acm.
- Aminian, K., & Najafi, B. (2004). Capturing human motion using body-fixed sensors: outdoor measurement and clinical applications. *Computer animation and virtual worlds*, *15*(2), 79-94.
- Kidder, S. M., Abuzzahab, F. S., Harris, G. F., & Johnson, J. E. (1996). A system for the analysis of foot and ankle kinematics during gait. *IEEE Transactions on Rehabilitation Engineering*, *4*(1), 25-32.
- Kuipers, J. B. (1999). *Quaternions and rotation sequences* (Vol. 66, pp. 127-143). Princeton: Princeton university press.

- Lara, O. D., & Labrador, M. A. (2012). A survey on human activity recognition using wearable sensors. *IEEE communications surveys & tutorials*, *15*(3), 1192-1209.
- Liu, J., Shahroudy, A., Xu, D., & Wang, G. (2016, October). Spatio-temporal lstm with trust gates for 3d human action recognition. In *European Conference on Computer Vision* (pp. 816-833). Springer, Cham.
- Mannini, A., & Sabatini, A. M. (2010). Machine learning methods for classifying human physical activity from on-body accelerometers. *Sensors*, *10*(2), 1154-1175.
- Mayagoitia, R. E., Nene, A. V., & Veltink, P. H. (2002). Accelerometer and rate gyroscope measurement of kinematics: an inexpensive alternative to optical motion analysis systems. *Journal of biomechanics*, *35*(4), 537-542.
- Murad, A., & Pyun, J. Y. (2017). Deep recurrent neural networks for human activity recognition. *Sensors*, *17*(11), 2556.
- Nweke, H. F., Teh, Y. W., Al-Garadi, M. A., & Alo, U. R. (2018). Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges. *Expert Systems with Applications*, *105*, 233-261.
- Ordóñez, F., & Roggen, D. (2016). Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, *16*(1), 115.
- Paladugu, P., Hernandez, A., Gross, K., Su, Y., Neseli, A., Gombatto, S., & Ozturk, Y. (2016, June). A sensor cluster to monitor body kinematics. In *2016 IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN)* (pp. 212-217). IEEE.
- Ravi, D., Wong, C., Lo, B., & Yang, G. Z. (2016). A deep learning approach to on-node sensor data analytics for mobile or wearable devices. *IEEE journal of biomedical and health informatics*, *21*(1), 56-64.
- Ravi, D., Wong, C., Lo, B., & Yang, G. Z. (2016, June). Deep learning for human activity recognition: A resource efficient implementation on low-power devices. In *2016 IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN)* (pp. 71-76). IEEE.
- Ronao, C. A., & Cho, S. B. (2015, November). Deep convolutional neural networks for human activity recognition with smartphone sensors. In *International*

Conference on Neural Information Processing (pp. 46-53). Springer, Cham.

Ronao, C. A., & Cho, S. B. (2016). Human activity recognition with smartphone sensors using deep learning neural networks. *Expert systems with applications*, *59*, 235-244.

Davis III, R. B., Ounpuu, S., Tyburski, D., & Gage, J. R. (1991). A gait analysis data collection and reduction technique. *Human movement science*, *10*(5), 575-587.

Sazonov, E. S., Fulk, G., Hill, J., Schutz, Y., & Browning, R. (2010). Monitoring of posture allocations and activities by a shoe-based wearable sensor. *IEEE Transactions on Biomedical Engineering*, *58*(4), 983-990.

Bamberg, S. J. M., Benbasat, A. Y., Scarborough, D. M., Krebs, D. E., & Paradiso, J. A. (2008). Gait analysis using a shoe-integrated wireless sensor system. *IEEE transactions on information technology in biomedicine*, *12*(4), 413-423.

Su, Y., Routhu, S., Aydinalp, C., Moon, K., & Ozturk, Y. (2015, December). Low power spinal motion and muscle activity monitor. In *2015 IEEE Global Communications Conference (GLOBECOM)* (pp. 1-5). IEEE.

Uslu, G., Altun, Ö., & Baydere, S. (2011, December). A Bayesian approach for indoor human activity monitoring. In *2011 11th International Conference on Hybrid Intelligent Systems (HIS)*(pp. 324-327). IEEE.

Uslu, G., & Baydere, S. (2015). RAM: Real Time Activity Monitoring with feature extractive training. *Expert Systems with Applications*, *42*(21), 8052-8063.

Uslu, G., Dursunoglu, H. I., Altun, O., & Baydere, S. (2013). Human activity monitoring with wearable sensors and hybrid classifiers. *International Journal of Computer Information Systems and Industrial Management Applications*, *5*, 345-353.

Wang, J., Chen, Y., Hao, S., Peng, X., & Hu, L. (2019). Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, *119*, 3-11.

Yang, J., Nguyen, M. N., San, P. P., Li, X. L., & Krishnaswamy, S. (2015, June). Deep convolutional neural networks on multichannel time series for human activity recognition. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Yao, S., Hu, S., Zhao, Y., Zhang, A., & Abdelzaher, T. (2017, April). Deepsense: A unified deep learning framework for time-series mobile sensing data processing.

In *Proceedings of the 26th International Conference on World Wide Web* (pp. 351-360). International World Wide Web Conferences Steering Committee.

Zebin, T., Scully, P. J., & Ozanyan, K. B. (2016, October). Human activity recognition with inertial sensors using a deep learning approach. In *2016 IEEE SENSORS* (pp. 1-3). IEEE.

Zeng, M., Nguyen, L. T., Yu, B., Mengshoel, O. J., Zhu, J., Wu, P., & Zhang, J. (2014, November). Convolutional neural networks for human activity recognition using mobile sensors. In *6th International Conference on Mobile Computing, Applications and Services* (pp. 197-205). IEEE.

Zhao, Y., Yang, R., Chevalier, G., Xu, X., & Zhang, Z. (2018). Deep residual bidirectional LSTM for human activity recognition using wearable sensors. *Mathematical Problems in Engineering, 2018*.

Zhu, C., & Sheng, W. (2009, May). Human daily activity recognition in robot-assisted living using multi-sensor fusion. In *2009 IEEE International Conference on Robotics and Automation* (pp. 2154-2159). IEEE.

Zhu, W., Lan, C., Xing, J., Zeng, W., Li, Y., Shen, L., & Xie, X. (2016, March). Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks. In *Thirtieth AAAI Conference on Artificial Intelligence*.

ÖZGEÇMİŞ

1. **Adı Soyadı** : Ayşenur Gençdoğmuş
2. **Doğum Tarihi** : 08.10.1994
3. **Unvanı** : Araştırma Görevlisi
4. **Öğrenim Durumu** : Lisansüstü Öğrencisi
5. **Çalıştığı Kurum** : İstanbul Sabahattin Zaim Üniversitesi

Derece	Alan	Üniversite	Yıl
Lisans	Bilgisayar Mühendisliği	İstanbul Sabahattin Zaim Üniversitesi	2012-2016
Y. Lisans	Bilgisayar Bilimi ve Mühendisliği(Tezli)	İstanbul Sabahattin Zaim Üniversitesi	2016-2019
Doktora			

6. Akademik Unvanlar

Araştırma Görevlisi: 2017-

7. Projeler

1. TÜBİTAK 4005 Bilim ve Toplum Yenilikçi Eğitim Uygulamaları Destekleme Programı:

Akademisyen ve Lisansüstü Öğrencilerine Yönelik Disiplinlerarası Bütünleştirilmiş Eğitim: R Programlama Dili ile Veri Analizi (Eğitmen)

2. TÜBİTAK Normal Araştırma Projesi:

Hareket Kinematığı Veri Analizi (Araştırmacı)

3. TÜBİTAK 2209-A Üniversite Öğrencileri Araştırma Projeleri Destekleme Programı:

Kronik veya Potansiyel Bel Ağrısına Sahip Hastaların Sensörlerden Okunan Hareket Kinematığı Verilerinin Analizi ve Derin Öğrenme ile Hareket Tahmini Yapılması (Danışman)

8. İdari Görevler

1.İstanbul Sabahattin Zaim Üniversitesi Mühendislik ve Doğa Bilimleri Fakültesi Bilgisayar Mühendisliği Bölümü Ders, Sınav Programı ve Mezuniyet Komisyonu Üyesi (2018)

2.İstanbul Sabahattin Zaim Üniversitesi Mühendislik ve Doğa Bilimleri Fakültesi Bilgisayar Mühendisliği Bölümü Staj Komisyonu Üyesi (2018)

3.BOLOGNA ve Kalite Komisyonu Üyesi (2018)

