

## AdaBelief Optimizasyon Tekniğinin Derin Öğrenmeye Dayalı Yaya Rotası Tahmin Uygulamalarına Etkisinin “Yakınsama” açısından İncelenmesi

Sevcan TURAN<sup>1\*</sup>, Feyzullah TEMURTAŞ<sup>2</sup>

<sup>1</sup> Bandırma Onyedli Eylül Üniversitesi, Fen Bilimleri Enstitüsü, Akıllı Ulaşım Sistemleri ve Teknolojileri Doktora Programı, 10200, Bandırma, Balıkesir, Türkiye

<sup>2</sup> Bandırma Onyedli Eylül Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü, 10200, Bandırma, Balıkesir, Türkiye

Alındı/Received: 11/01/2024; Kabul/Accepted: 12/02/2024; Yayın/Published: 12/03/2024

\* Sorumlu yazar e-posta: [sevcanturan@comu.edu.tr](mailto:sevcanturan@comu.edu.tr)

### Öz

Son yıllarda, görüntü işleme teknikleri kullanılarak yaya rotasının tahmini, hızla dikkat çeken bir araştırma konusu haline gelmiştir. Rota tahmin uygulamalarında Derin Öğrenmenin kullanımı, mühendislik çalışmalarıyla yapılan geleneksel parametre belirleme işlemlerine ihtiyaç duymayan ve daha doğru tahminler yapabilen yeni uygulamaların geliştirilmesini sağlamıştır. Rota tahmini için sıklıkla veriye dayalı olan gözetimli derin öğrenme modelleri kullanılmaktadır. Ancak, bu modellerin eğitimi yüksek hesaplama maliyeti getirmektedir. Bu maliyetleri azaltmak ve doğrulukları arttırmak için iyi yakınsama ve genelleştirme özelliklerine sahip optimizasyon yöntemlerini seçmek önemlidir. Bu çalışma, ETH/UCY veri kümeleri kullanılarak derin öğrenme mimarisi temelli geliştirilmiş rota tahmini algoritmalarının optimizasyon yöntemi açısından performansını incelemektedir. Özellikle, modelin eğitimi aşamasında yakınsama açısından AdaBelief optimizasyon tekniğinin avantajları ve dezavantajlarına odaklanılmaktadır. Çalışmanın sonuçları, AdaBelief'in eğitim sürecine pozitif bir katkıda bulunduğunu ve rota tahmini algoritmalarının genel performansını arttırabileceğini göstermektedir.

**Anahtar Kelimeler:** AdaBelief, derin öğrenme, ETH/UCY, yaya rota tahmini, optimizasyon

## Investigation of the Effects of AdaBelief Optimization Technique on Deep Learning-Based Pedestrian Path Prediction Applications in terms of “Convergence”

### Abstract

In recent years, the prediction of pedestrian paths using computer vision techniques has become an increasingly attractive topic of research. The use of deep learning techniques has led to the development of new path prediction applications that do not rely on the traditional parameter determination processes with engineering studies. This has resulted in more accurate predictions. Supervised deep learning models, which are data-driven, have been widely used for path prediction. However, the training of these models is associated with high computational costs. To address this issue, it is important to choose optimization methods that have good convergence and generalization properties in order to reduce costs and improve accuracy. This study examines the performance of path prediction algorithms based on deep learning architectures using the ETH/UCY datasets. In particular, the study focuses on the advantages and disadvantages of the AdaBelief optimization technique in terms of convergence during the training phase. The results of the study show that the AdaBelief makes a positive contribution to the training process and can improve the overall performance of the path prediction algorithm.

**Key Words:** AdaBelief, deep learning, ETH/UCY, pedestrian path prediction, optimization

**Atıf / To cite:** Turan S, Temurtaş F (2024). AdaBelief optimizasyon tekniğinin derin öğrenmeye dayalı yaya rotası tahmin uygulamalarına etkisinin “Yakınsama” açısından incelenmesi. Karamanoğlu Mehmetbey Üniversitesi Mühendislik ve Doğa Bilimleri Dergisi, 6(1): 1-10.

## 1. GİRİŞ

Derin öğrenme alanında popüler konulardan biri, yayaların gelecekteki rotasını tahmin etmektir (Gulzar ve ark. 2021; Rudenko ve ark. 2019). Yaya rota tahmini, geçmişteki rota bilgilerinin kullanılarak yayanın gelecekte takip edebileceği yolu tahmin etme sürecidir. Rota tahmini yaparken, renkli (Jain ve ark. 2019), mono (Kolcu ve Bolat 2018) veya kızılötesi kameralar (Pei ve ark. 2020; Zhou ve ark. 2020) vasıtasıyla elde edilen görüntüler ve lidar (Hecht 2018; Jain ve ark. 2019; Ma ve ark. 2019), radar veya ultrasonik sensörler (Wang ve ark. 2020) ile elde edilen sensör verileri kullanılabilir. Rota tahmini, sensör bilgileri ayrı ayrı kullanarak yapılabileceği gibi sensör füzyon teknikleri ile birleştirilerek de yapılabilir. Yaya rota tahmini ile, sosyal güvenlik açısından yayaların izlenmesi, insanlarla uyumlu bir şekilde hareket eden otonom araçların veya sosyal hayat ortamlarında insanlara yardım eden robotların geliştirilmesi gibi işlemler hedeflenmektedir (Rudenko ve ark. 2019). Bu alanda, ilk zamanlarda yayaların fiziksel hareket özellikleri ve çevresel etkenlerin etkileri mühendislik çalışmaları ile modellenerek kısa vadeli tahminler yapılmıştır (Bera ve ark. 2016; Hariyono ve ark. 2015; Keller ve Gavrila 2014). 2016 yılında, Uzun-Kısa Süreli Bellek (LSTM) kullanılarak ilk kez uzun vadeli rota tahmini yapılmış ve "Social-LSTM" adı verilen bu çalışma diğer çalışmalar tarafından karşılaştırma için bir referans olarak kabul edilmiştir (Alahi ve ark. 2016). Bu çalışmada, her bir yayaya ait bir LSTM tanımlanmış ve bu LSTM'ler arasında gizli katman bilgileri paylaşılmıştır. LSTM havuzlama işlemi ile yakın olan ve aynı sosyal etkiyi yaratacak yayaların gizli katman bilgisi birleştirilmiştir. Böylece yayalar arasındaki etkileşim öğrenilmeye başlanmıştır. Social-LSTM tarafından sunulan perspektifle, bu çalışma kapsamında da incelenmek üzere seçilmiş olan veri odaklı ve derin öğrenme tabanlı uzun vadeli yol tahmini çalışmaları geliştirilmeye başlanmıştır (Chen ve ark. 2021; Gupta ve ark. 2018; Huang ve ark. 2019; Mendieta ve Tabkhi 2020; Mohamed ve ark. 2020; L. Shi ve ark. 2021). Bu çalışmalar genellikle geçmişteki 8 karelik (3.2 saniyelik) rota bilgilerini kullanarak gelecekteki 12 karelik (4.8 saniyelik) takip edilebilecek rota bilgilerini tahmin etmeyi amaçlamaktadır.

Derin öğrenme tabanlı sistemlerde önemli bir aşama olan eğitim (training) uzun süre alan, çok sayıda parametreyi hafızada saklamayı ve işlemeyi gerektiren bir süreçtir. Optimizasyon teknikleri ile bu parametrelerin en iyi değerlerinin bulunup modelin en hassas ve doğru sonuçları üretebilecek hale getirilmesi hedeflenmekte ve optimizasyon tekniğinin hızlıca yakınsaması ve genelleştirmesine ihtiyaç duyulmaktadır. Modelin ihtiyaç duyduğu bu bilgileri hızlıca işlemek için yüksek performanslı bilgisayarlara, GPU işlemciler ve yüksek bellek kapasitelerine ihtiyaç duyulmaktadır (Xue-Wen Chen ve Xiaotong 2014) ve bu tür bilgisayarlar yüksek enerji tüketen cihazlardır (Mittal ve Vetter 2015). Ayrıca eğitim işlemi için bulut tabanlı hizmetler sunulmakta fakat bu hizmetlerin ücreti kullanılan GPU sayısı ve kullanım sürelerine göre belirlenmektedir. Bu gibi nedenlerle, daha az işlem adedi ve maliyetiyle hassas sonuçlar veren eğitim süreci gerçekleştirmek önem arz etmektedir.

Geliştirilen modele eğitim sırasında kullanılan veri kümesine ve derin öğrenme algoritmalarına göre bazı üst değişkenler (hyper-parameter) belirlenmektedir. Bu parametrelerden biri, geliştirilen modelde ağ yapısının gerektirdiği "ağırlık değerlerini" bulmak için yapılması gereken tüm işlemlerin eğitim veri kümesindeki tüm veriler üzerinde kaç kez tekrarlanması gerektiğini ifade eden "Dönem (epoch)" sayısıdır. Geliştirilen modelin az sayıda dönem ile yakınsamasını sağlamak için hangi optimizasyon tekniğinin kullanılacağına seçimi önem kazanmaktadır.

Literatürde optimizasyon yöntemlerini karşılaştıran çalışmalar mevcuttur (Liu ve ark. 2021; Ozyildirim ve Kiran 2020; Zhu ve ark. 2021). Bu çalışmanın katkısı ve amacı, "yol tahmini" çalışmalarında elde edilen en iyi değerlendirme metrik değerlerine ulaşmak için "yakınsama" açısından eğitim aşamasında optimizasyon yönteminin dönem sayısına etkisini incelemektir. Bu amaçla, açık kaynak kodlu ve farklı derin öğrenme modelleri kullanılan çalışmalar seçilmiş ve yayınların orijinal optimizasyon tekniği ile yeni bir optimizasyon yöntemi olan AdaBelief'in (Zhuang ve ark. 2020) sonuçları karşılaştırılmıştır.

AdaBelief'in tercih edilmesinin nedeni, hızlı bir şekilde yakınsaması, genelleştirmesi ve doğru sonuçlar elde etmesi nedeniyle son zamanlarda birçok farklı alanda yapılan çalışmalarda öne çıkmış olmasıdır: Güç santrallerinde elektrik yükünün tahmini (Shi ve ark. 2021), makine arızası öncesi gerekli bakımın tespiti (Lv ve ark. 2021), derin öğrenme algoritmaları için düşman örneklerin genelleştirilmesi (Wang ve ark. 2021), hava tahmini (Leinonen 2021), biyomedikal alandaki genom bağlantısı tahmini (Guo ve ark. 2021) vb.

### 1.1 İncelenen Açık Kaynak Kodlu Yaya Tahmin Çalışmaları

Çalışmalar; son yıllarda yayınlanmış, ETH/UCY veri kümelerini kullanan, açık kaynak koduna sahip ve farklı derin öğrenme yöntemlerini kullanan çalışmalardan seçilmiştir. Seçilen bu çalışmalar Tablo 1'de özetlenmiş olup detayları aşağıdaki gibidir:

*Social-GAN* (Gupta ve ark. 2018). Bu çalışmanın motivasyon noktası önceki çalışmaların sadece tahmin edilen rotaların Öklid mesafesi açısından hata oranını azaltmayı amaçladığı ve sosyal etkileşimin yalnızca yakın komşular arasında dikkate alındığı olarak belirtilmektedir. Yazarlar, bütün görüntüdeki yaya etkileşimini dikkate alarak, olası yolların oluşumunu ve iyi yolların öğrenimini temel alan bir model geliştirmişlerdir. Çalışmada, Çekişmeli Üretici Ağ (GAN) yapısı kullanılmıştır. GAN'ın Üretici (Generator) aşamasında, her yaya için geçmiş yol bilgilerine göre gelecekte takip edilecekleri farklı yollar oluşturulmuş ve Ayrıcı (Discriminator) aşamasında hangi yolların gerçek olarak takip edilebileceği ve hangilerinin sahte olabileceğinin ayırt edilmesi öğrenilmiştir. Üretici aşamasında ilk olarak, veri kümesindeki koordinat bilgisi, yapay sinir ağı (ANN) kullanarak sabit uzunluklu bir vektöre dönüştürülmüş ve her yaya için tanımlanan Kodlayıcı (Encoder) bölümündeki LSTM için girdi olarak

kullanılmıştır. Tüm yayalar tarafından elde edilen LSTM verileri birbirleriyle örnekleme (pooling) yoluyla paylaşılmıştır. Sonuçlara gürültü eklenmiş ve bu gürültü değerleri her yaya için tanımlanan Çözümleyici (decoder) bölümündeki LSTM için girdi olarak kullanılmış ve yayanın takip edebileceği olası yollar oluşturulmuştur. Ayırıcı aşamasında, üretici aşamasından gelen bilgiler ve gerçek rota bilgileri beraber ele alınıp verinin gerçek/sahte skoru hesaplanır böylece iyi yol bilgileri de öğrenilmiş olur. Üreticide farklı yolların oluşturulabilmesi için, gürültü ekleme aşamasında Rastgele Örnek Üretim algoritmaları kullanılmıştır.

*Social-STGCNN* (Mohamed ve ark. 2020). Çalışmada, zaman ve mekânı ele alan Zamansal Evrişimli Ağ (CNN) ve Evrişimli Çizge Ağ (GCN) kullanılmıştır. Yazarlar, geliştirdikleri model sayesinde, karşılaştırdıkları çalışmalara göre 48 kat daha hızlı ve %20 daha yüksek doğruluk oranında sonuçlar ürettiklerini ve 8,5 kat daha az parametre kullandıklarını belirtmektedirler. Rota tahmininde görüntüler çizge olarak temsil edilmiş, görüntüdeki yayalar düğüm ve yayalar arası etkileşimler ise ağırlıklı kenar olarak tanımlanmıştır. Geçmiş rota bilgileri kullanılarak elde edilen komşuluk matrisleri yoğun haline getirilmiş ve zaman içerisindeki değişim takip edilmiştir. Öncelikle, geçmiş rotalara ait özellik vektörleri, mekân ve zaman bilgisini içeren çizgeler üzerinde GCN kullanılarak hesaplanmıştır. Daha sonra bu özellikler kullanılarak, gelecek rota tahminleri Zamansal CNN modeli ile gerçekleştirilmiştir.

*STGAT* (Huang ve ark. 2019). Çalışmada, yayaların birbirlerine ne derecede etkide bulunacakları ile yayanın kendi geçmiş hareket yapısının birlikte dikkate alınarak gelecekteki rota tahminin yapılması gerektiği belirtilmektedir. Bu nedenle, yer-zaman etkileşimini ve yayaların birbirleriyle olan etkileşim oranını Dikkat Tabanlı Çizge Ağı (GAT) ile hesaplayan bir model geliştirmişlerdir. Rota tahmini uygulamalarında, yayaların koordinat verilerini ve yayalar arası etkileşimi çizge olarak ifade etmenin, tahminin daha doğru yapılabilmesini sağladığını belirtmektedirler. Bu amaçla, görüntüdeki her yaya bir düğüm olarak düşünülmüş, yayalar arası etkileşim oranı Dikkat Ağı ile hesaplanmış ağırlıklı kenar olarak kabul edilmiştir. Modelin kodlayıcı aşamasında, ilk olarak, her yayanın hareket özellikleri geçmiş yol bilgilerinden LSTM ile öğrenilmiştir. Daha sonra, yayaların mekânsal alandaki birbirlerine etkileri GAT ile hesaplanmıştır. Son olarak, bu sonuçlar kullanılarak, her yaya için yeni bir LSTM modülü kullanılarak etkileşimin zaman içi bağlamı bulunmuştur. Bu zaman ve mekân bilgileri sonra ANN aracılığıyla birleştirilmiştir. Birleştirilmiş bilgilere gürültü eklenmiş, çözümleyici modülüne girdi olarak gönderilmiş ve LSTM ile olası gelecek rotalar hesaplanmıştır.

*Causal-HTP* (Chen ve ark. 2021). Çalışmada, eğitim verilerinden kaynaklanan yanlışlık durumunun test sonuçlarına etkisi üzerine odaklanılmıştır. Hotel veri kümesi örnek olarak seçilmiş ve eğitim kümesi ile test kümesi arasındaki büyük farklar ortaya konulmuştur. Gerçek hayatta karşılaşılabilecek durumların çeşitliliğinden dolayı farkların ortaya çıkmasının kaçınılmaz olduğu düşünüldüğü rota tahminine etki eden çevresel faktörlerden

öğrenilen bilginin etkisinin azaltılıp daha fazla rota izini ön plana çıkartan Karşıt Örnek Analizi (Guo ve Fraser 2014) tabanlı bir yöntem geliştirilmiştir. Örneğin, eğitim veri kümesinde yürüyen insanlar kavşaklarda sola dönmeyi tercih etmişlerse, model yayaların her zaman kavşaklarda sola döndüğünü öğrenir. Bu sorunu aşmak için, model içinde önce gerçek geçmiş rota bilgisi ve çevresel etkiler temel alınarak tahminler yapılmış, sonra geçmiş rota ve çevresel etkiler değiştirilerek (rotaları kombine etme, ortalama vb.) tahminler tekrarlanmıştır. Bu iki sonuç ve referans bilgisi arasındaki farkı karşılaştırılarak rota tahmini yapılmıştır. Çalışmada, Karşıt Örnek Analizi yöntemi Social-STGCNN ve STGAT projelerine uygulanıp sonuçlar üzerindeki etkisi incelenmiştir.

*Carpe Posterum* (Mendieta ve Tabkhi 2020). Çalışmada gömülü sistemlerde çalışabilecek, gerçek zamanlı tahminler yapabilecek ve hızlı bir şekilde doğru sonuçlar verebilecek bir model geliştirme amaçlanmıştır. Modelde, İzomorfik Çizge Ağı çizgenin yapısal özelliklerini öğrenmek için kullanılmış ve CNN gelecekteki rotayı tahmin etmek için kullanılmıştır. Yayaların rota koordinat bilgileri ile başlangıç noktasına göre yer değiştirme bilgileri birlikte İzomorfik Çizge Ağına gönderilip Mesaj Gönderme yöntemiyle komşu düğümlerden elde edilen bilgilerle her yaya için hareket özellikleri elde edilmektedir. CNN'de, bu verilerden hız ve pozisyon değişimleri öğrenilip rota tahminleri yapılmaktadır.

**Tablo 1.** İncelenen açık kaynak kodlu çalışmaların özellikleri

| Çalışma                           | Derin Öğrenme Modeli              | Optimizasyon Tekniği  | Dönem sayısı  | Yığın ölçüsü |
|-----------------------------------|-----------------------------------|-----------------------|---|--------------|
| <b>Social-GAN</b>                 | GAN, LSTM                         | Adam                  | ETH: 769<br>Hotel: 765<br>Univ: 754<br>Zara1-2: 752 | 64           |
| <b>Social-STGCNN</b>              | GCN, Zamansal CNN                 | Rasgele Gradyan İnişi | 250   | 128          |
| <b>STGAT</b>                      | GAT, LSTM                         | Adam                  | 400   | 64           |
| <b>Casual-HTP (STGAT tabanlı)</b> | GAT, LSTM                         | Adam                  | 400   | 64           |
| <b>Carpe Posterum</b>             | Graf İzomorfizm Ağ, CNN           | Adam                  | 80  | 64           |
| <b>SGCN</b>                       | Dönüştürücü Ağ, GCN, Zamansal CNN | Adam                  | 150   | 128          |

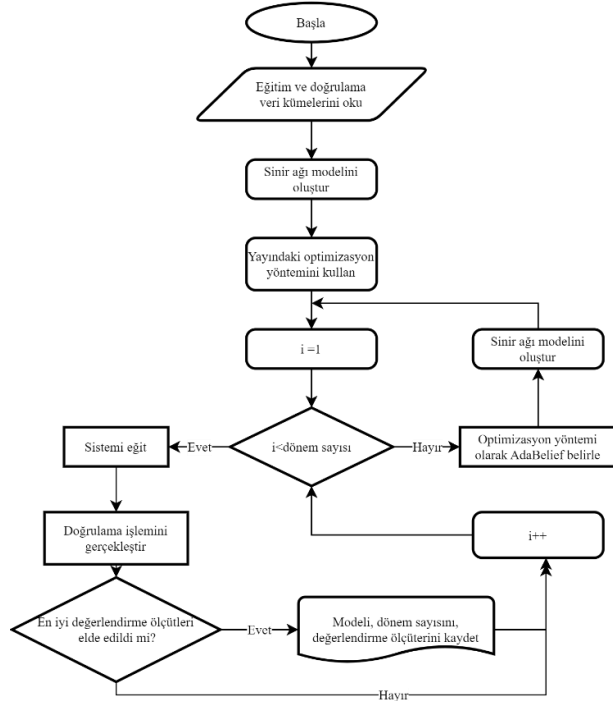
*SGCN* (Shi ve ark. 2021). Diğer çalışmalarda yayaların arasındaki etkileşimin genellikle yoğun ve yönü belirsiz çizge ile ifade edildiği ve çarpışmaları önlemek için gidilecek rotanın bulunmasına odaklanıldığını belirtilmiştir. Bu yaklaşımların yayaların birbirlerinin rotasını nasıl etkilediğini ifade etmekte yetersiz olduğu vurgulanmıştır. Bu eksiklikleri gidermek amaçlı, modelde seyreltilmiş yönlü komşuluk matrislerine dayalı iki ayrı çizge ağı kullanılmıştır: Birincisi yayanın bireysel hareket özelliğini zaman bağlamında bulmak, diğeri yayaların birbirlerini nasıl etkilediğini mekânsal bağlamında bulmak içindir. İlk olarak, zaman ve mekân boyutları açısından iki ayrı Dönüştürücü Ağ ve CNN kullanılarak

seyreltilmiş ve yönlü çizge komşuluk matrisi elde edilmiştir. Bu komşuluk matrisleri kullanılarak GCN ile zaman ve mekân dilimindeki rotaların özellikleri öğrenilmiştir. Ardından, her yaya için bu veriler birleştirilip Zamansal CNN'ye girdi olarak gönderilmiş ve rota tahminleri yapılmıştır.

Çalışmada ikinci bölümde uygulanan yöntem, çalışmalarda kullanılan veri kümeleri, değerlendirme ölçütleri, derin öğrenme yöntemleri ve optimizasyon yöntemleri ile ilgili bilgi verilecektir. Üçüncü bölümde elde edilen deneysel sonuçlar tartışılıp dördüncü bölümde çalışma hakkında yorumlar yapılacaktır.

## 2. MATERYAL VE METOT

Bu çalışmanın kapsamında, derin öğrenme algoritmalarının eğitim aşamasında AdaBelief optimizasyon tekniğinin yakınsama açısından etkisi incelenmiştir. Bu amaçla, PyTorch kütüphanesi kullanılarak geliştirilmiş, ETH-UCY (genellikle kabul edilen veri kümeleri) veri kümelerini kullanan, son yıllarda yayınlanmış ve açık erişimli yazılım kaynak koduna sahip makaleler seçilmiştir. Şekil 1'deki çalışmanın akış şemasında da görülebileceği gibi, açık erişimli kaynak kodlar kullanılarak, model ilk olarak yayımlandığı şekilde eğitilmiştir. En iyi değerlendirme ölçütlerinin elde edildiği dönem numarası ve bu dönemde elde edilen değerlendirme ölçütü değerleri kaydedilmiştir. Daha sonra AdaBelief optimizasyon yöntemi olarak ayarlanmış, sistem yeniden eğitilmiş ve ilgili sonuçlar kaydedilmiştir. Casual-HTP ve STGAT çalışmalarında, her iki eğitim sırasında da bellek hataları alındığından, yığın boyutu 32 olarak belirlenmiştir. Casual-HTP çalışmasında, STGAT tabanlı model kullanılmıştır.



Şekil 1. Çalışmanın akış diyagramı

## 2.1. Veri Kümeleri

Bu çalışma kapsamında ETH (Pellegrini ve ark. 2009) ve UCY (Lerner ve ark. 2007) veri setleri tercih edilmiştir. Şekil 2'de örnek görüntüleri bulunan ve kuş bakışı görüntülere sahip ETH-UCY, rota tahmini çalışmalarında temel kalite test veri kümesi olarak kabul edilmektedir. ETH içerisinde ETH ve Hotel; UCY içerisinde ise Univ, Zara1 ve Zara2 adında farklı özelliklere sahip video akışları bulunmaktadır. Bu videolardan 0,4 saniyede bir kesitler alınmış ve elde edilen çerçevelerdeki yayaların rota bilgilerinden veri kümesi oluşturulmuştur. Rota bilgilerinin içerisinde video çerçeve numarası (FrameID), yaya numarası (PedID), dünya koordinatları cinsinden y ve x koordinat bilgileri bulunmaktadır. ETH 750 ve UCY 786 farklı yayanın rota bilgilerini içermektedir.



Şekil 2. ETH/UCY Veri kümesinden örnek görüntüler. a) ETH, b) Hotel, c) Univ, d) Zara1, e) Zara2

## 2.2. Rota Tahmininde Kullanılan Değerlendirme Ölçütleri

Rota tahmininde değerlendirme ölçütleri olarak Ortalama Yer Değiştirme Hatası (ADE) ve Nihai Yer Değiştirme Hatası (FDE) kullanılır. ADE hesaplanırken, tüm yayalar için gelecekteki on iki çerçevedeki tahmin edilen koordinatların gerçek referans koordinatlarına olan mesafesinin ortalaması alınır. FDE hesaplanırken, son çerçeve için tahmin edilen koordinat ile gerçek referans koordinatı arasındaki mesafe bulunur (Rudenko ve ark. 2019). N yaya sayısı,  $T_p$  koordinat tahmini yapılacak çerçeve sayısı,  $(\hat{x}_{tn}, \hat{y}_{tn})$  tüm çerçeveler için tahmin edilen koordinatlar,  $(x_{tn}, y_{tn})$  gerçek referans bilgisindeki koordinatlar,  $(\hat{x}_{Tpn}, \hat{y}_{Tpn})$  ulaşılabilecek son koordinat bilgisi için tahmin edilen değer ve  $(x_{Tpn}, y_{Tpn})$  gerçek referans bilgisindeki son ulaşılabilecek koordinat bilgisi olmak üzere ADE ve FDE sırasıyla Eş. 1 ve Eş.2'deki gibi hesaplanır:

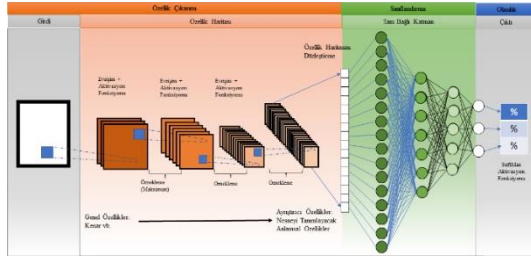
$$ADE = \frac{\sum_{n \in N} \sum_{t \in T_p} \sqrt{(\hat{x}_{tn} - x_{tn})^2 + (\hat{y}_{tn} - y_{tn})^2}}{N \times T_p} \quad (1)$$

$$FDE = \frac{\sum_{n \in N} \sqrt{(\hat{x}_{Tpn} - x_{Tpn})^2 + (\hat{y}_{Tpn} - y_{Tpn})^2}}{N} \quad (2)$$

## 2.3. Seçilen Çalışmalarda Kullanılan Derin Öğrenme Yöntemleri

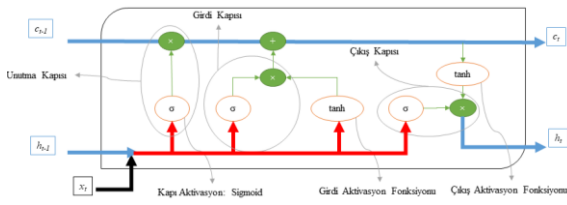
**Evrişimli Sinir Ağı (Convolutional Neural Network-CNN)** (Le Cun ve ark. 1989): Genel yapısı Şekil 3'te gösterilen ve bilgisayarla görme süreçlerinde çok sık kullanılan CNN, filtreler aracılığıyla evrişim işlemleri uygulayarak girdi olarak gönderilen veriler hakkında

özellikler elde edilmesini sağlayan derin öğrenme yöntemidir (Turan ve ark. 2021). CNN yapılarında ilk katmanlarda verilerden kenarlar vb. temel özellikler elde edilirken, derin katmanlarda verilerin ayrıştırılmasını sağlayan ayırt edici özellikler elde edilmektedir. Bu özellikler daha sonra sınıflandırma katmanı tarafından yorumlanmaktadır.



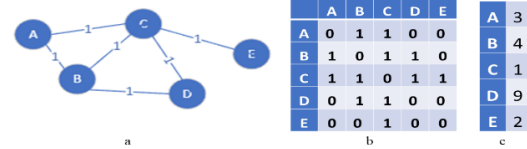
Şekil 3. CNN Yapısı (Turan ve ark. 2021)

**Uzun-Kısa Süreli Bellek (Long-Short Term Memory-LSTM)** (Hochreiter ve Schmidhuber 1997): LSTM, gelecek durum tahmin işlemlerinde geçmişteki verilerin kullanılmasına ihtiyaç duyulduğunda tercih edilmektedir. Şekil 4'te görüldüğü üzere LSTM'de hücre durumu, girdi kapısı, unutmaya kapısı ve çıktı kapısı yapıları bulunmaktadır. Hücre durumu (c) geçmiş bilgileri tutmaktadır. Unutmaya kapısı hücre durumundaki hangi bilgilerin saklanması gerektiğine, giriş kapısı o zaman dilimindeki girdilerden (x) hangisinin hücre durumuna dahil edilmesi gerektiğine karar vermekte ve çıkış kapısı da anlık girdi verisi ile hücre durumunu kullanarak sonuç (h) üretmektedir.



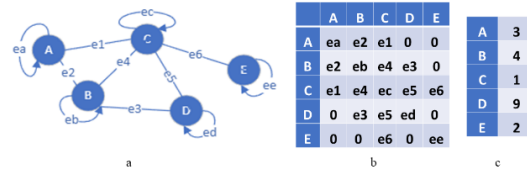
Şekil 4. LSTM Yapısı (Turan ve ark. 2021)

**Evrışimli Çizge Ağı (Graph Convolution Network- GCN)** (Kipf ve Welling 2016): Çizgeler, Şekil 5.a'da görülebileceği üzere düğüm ve düğümler arasındaki komşuluk ilişkisini ifade eden kenarlardan oluşan veri yapılarıdır. Düğümler V, kenarlar E ile gösterildiği durumda çizge  $G = \{V, E\}$  şeklinde ifade edilmektedir. Çizge ağlarda düğümler arasındaki ilişki Komşuluk Matrisi (Adjacency matrix) ile düğümlere ait bilgiler ise özellik vektörü ile tutulmaktadır. Hesaplamalar aşamasında komşuluk matrisi ile özellik vektörleri çarpıldıktan sonra bir araya getirme yöntemleri uygulanarak komşu düğümlerdeki bilgiler düğüm bilgilerine dahil edilir. Sonrasında normalizasyon, ağırlık matrisiyle çarpma ve aktivasyon fonksiyonu işlemleri gerçekleştirilir. Böylece çizgeye ait özellikler elde edilmiş olur ve bu özellikler ile düğüm, kenar veya çizge seviyesinde bilgi çıkarımı işlemleri gerçekleştirilebilir (Wu ve ark. 2019; Zhang ve ark. 2019; Zhou ve ark. 2020).



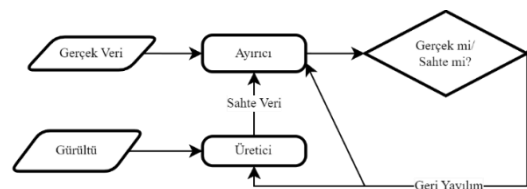
Şekil 5. a) Basit çizge yapısı. b) Basit komşuluk matrisi. c) Örnek düğüm özellik vektörü

**Dikkat Tabanlı Çizge Ağı (Graph Attention Network-GAT)** (Veličković ve ark. 2017): Çizge ağlarında genellikle komşuluk matrisindeki değerler, iki komşu düğümün birbirini aynı şekilde etkileyeceği varsayılarak '1' olarak atanmaktadır. Ancak yayaların düğüm olarak ifade edildiği yaya rotası tahmini uygulamalarında düğümler birbirini farklı oranlarda etkilemektedir. Düğümlerin birbirini ne derecede etkilediği üzerine yapılan çalışmalar GAT yapısını ortaya çıkarmıştır. GAT yapısında düğümlerin birbirleri üzerindeki etkisi ele alınmış ve dikkat değeri ile komşuluk matrisi tasarlanmıştır. Şekil 6, GAT' a dayalı örnek grafiği ve komşuluk matrisini göstermektedir.



Şekil 6. a) Dikkat tabanlı çizge yapısı. b) Komşuluk matrisi. c) Örnek düğüm özellik vektörü

**Çekişmeli Üretici Ağ (Generative Adversarial Network-GAN)** (Goodfellow ve ark. 2014): Şekil 7'de görülebileceği gibi GAN, üretici ve ayırtıcı olarak adlandırılan iki bölümden oluşur. Üretici aşamasında veri kümesindeki verinin türüne, yapısına ve özelliklerine uygun yeni sahte veri üretilmeye çalışılır. Ayırtıcı aşamasında, gerçek veri kümesi ile üretici aşamasında oluşturulan sahte veriler birlikte ele alınarak hangisinin sahte hangisinin gerçek olduğu belirlenmeye çalışılır. Eğitim turları ilerleyip geri yayılım yöntemiyle ağın ağırlık değerleri güncellendikçe, ağ veri kümesine yakın veri üretmeye ve verinin sahte mi yoksa gerçek mi olduğunu tespit etmeye başlar. Böylece üretici sahte veri üretiminde, ayırtıcı sahte/gerçek veriyi ayırt etmede uzmanlaşır.



Şekil 7. GAN yapısı

## 2.4. Optimizasyon Yöntemleri

Derin öğrenmenin eğitim aşamasında, modelin ürettiği hatayı azaltmak için optimizasyon yöntemleri kullanılmaktadır. Optimizasyon yöntemleriyle, modelde tercih edilen hata fonksiyonunun türevi kullanılarak modeldeki ağırlık değerlerinin ne kadar değişmesi gerektiği hesaplanmaktadır. Hızlı, hassas yakınsama ve



genelleme elde etmek için hangi optimizasyon tekniğinin tercih edileceğinin belirlenmesi önemli bir konudur. Aşağıda, çalışma sırasında kullanılan optimizasyon teknikleri hakkında bilgi verilmektedir:

**Rasgele Radyan İnişi (Stochastic Gradient Descent (SGD))** (Bottou ve ark. 1991): Algoritmada hata fonksiyonunun türevinin değeri hesaplanırken veri kümesindeki tüm veriler yerine rastgele seçilmiş veriler kullanılmaktadır. Böylece türevleri birbirine çok yakın olan tüm verilerle hesaplama maliyeti ortadan kaldırılmış olur. Rastgele verilerin seçilmesi ayrıca yerel minimum sorununu da ortadan kaldırır. Derin öğrenme modelinin yeni ağırlık değerleri:  $\alpha$  öğrenme katsayısı,  $L$  hata fonksiyonu,  $Z$  rastgele seçilen veriler olmak üzere Eş. 3'teki gibi hesaplanmaktadır.

$$w_{t+1} = w_t - \alpha \frac{\partial L(Z, w_t)}{\partial w_t} \quad (3)$$

**Adam** (Kingma ve Ba 2014): Bu yöntemde hata fonksiyonunun birinci türevi kullanılarak birinci ve ikinci momentum değerleri hesaplanmaktadır. Momentum değerleri, ilk eğitim döngülerinde sıfır olma yanlılığı büyük olduğu için düzeltilmektedir. Daha sonra düzeltilen momentum değerleri ile ağırlıkların ne kadar değiştirilmesi gerektiği hesaplanmaktadır. Yeni ağırlık değerleri:  $g_t$  hata fonksiyonunun türevi,  $m_t$  düzeltilmiş birinci momentum,  $v_t$  düzeltilmiş ikinci momentum,  $\alpha$  öğrenme katsayısı,  $\beta_1$  ve  $\beta_2$  sırasıyla birinci ve ikinci momentum azalma oranları,  $\epsilon$  sıfıra bölünmeyi önlemek için kullanılan hiper parametre olmak üzere Eş. 4-6 ile hesaplanmaktadır.

$$m_t = \frac{\beta_1 m_{t-1} + (1 - \beta_1) g_t}{1 - \beta_1} \quad (4)$$

$$v_t = \frac{\beta_2 v_{t-1} + (1 - \beta_2) g_t^2}{1 - \beta_2} \quad (5)$$

$$w_{t+1} = w_t - \alpha \frac{m_t}{\sqrt{v_t + \epsilon}} \quad (6)$$

**AdaBelief** (Zhuang ve ark. 2020): Algoritma temel olarak Adam optimizasyon yöntemine dayanmaktadır. Farklı olarak, ikinci momentum hesaplanırken parametre olarak birinci momentum da kullanılmaktadır. Yeni ağırlık değerleri:  $g_t$  hata fonksiyonunun türevi,  $m_t$  düzeltilmiş birinci momentum,  $s_t$  düzeltilmiş ikinci momentum,  $\alpha$  öğrenme katsayısı,  $\beta_1$  ve  $\beta_2$  sırasıyla birinci ve ikinci momentum azalma oranları,  $\epsilon$  sıfıra bölünmeyi önlemek için kullanılan hiper parametre olmak üzere Eş. 7-9 ile hesaplanmaktadır.

$$m_t = \frac{\beta_1 m_{t-1} + (1 - \beta_1) g_t}{1 - \beta_1} \quad (7)$$

$$s_t = \frac{\beta_2 s_{t-1} + (1 - \beta_2) (g_t - m_t)^2 + \epsilon}{1 - \beta_2} \quad (8)$$

$$w_{t+1} = w_t - \alpha \frac{m_t}{\sqrt{s_t + \epsilon}} \quad (9)$$

Türev ve birinci momentum değerleri birbirine yakınsa AdaBelief büyük adımlarla, aksi takdirde küçük adımlarla ilerlemektedir. Bu sayede daha hızlı ve istikrarlı yakınsama sağlamaktadır.

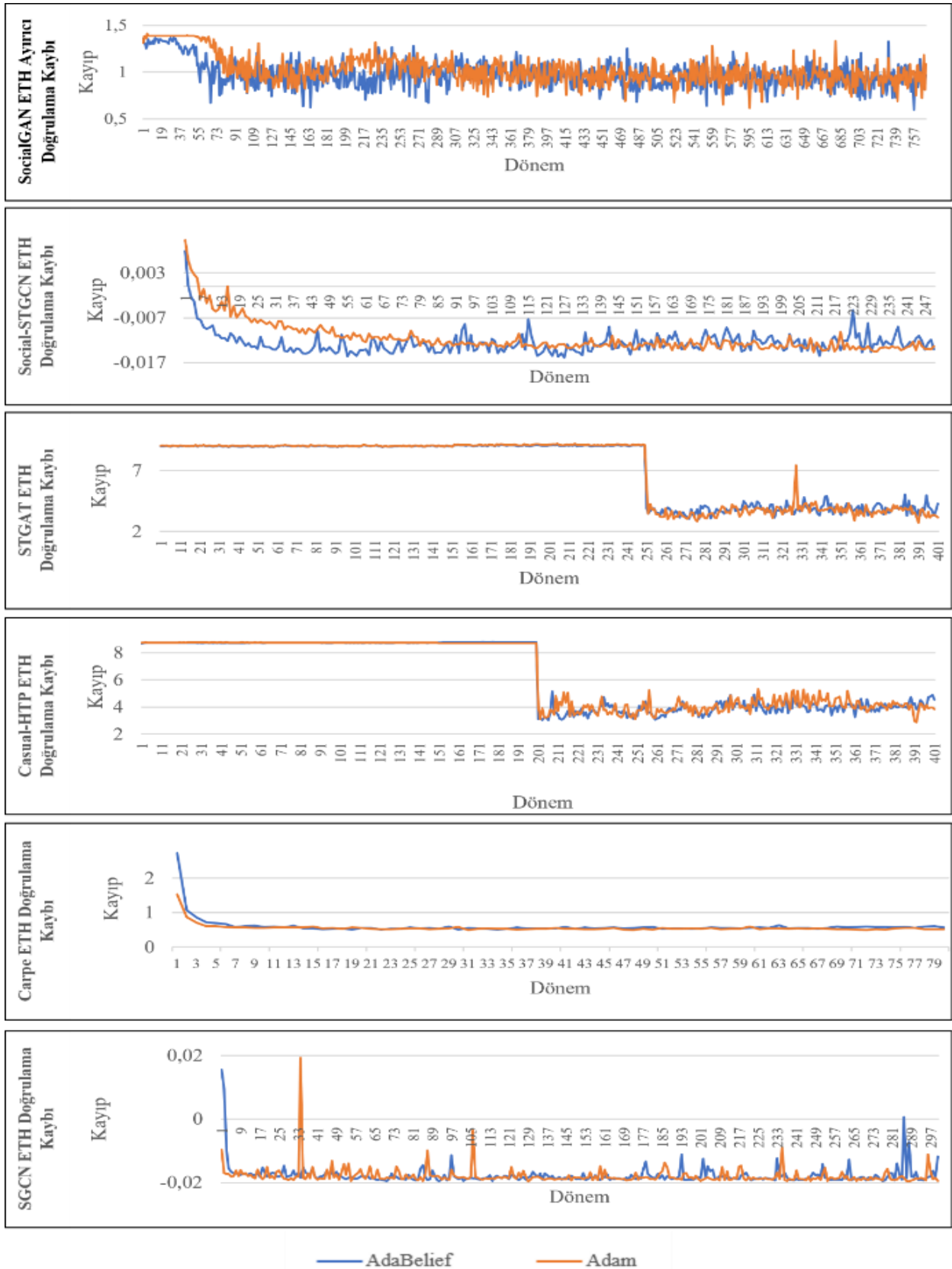
### 3. BULGULAR

Yayanın izleyebileceği rotayı yüksek hassasiyetli ve hızlı bir şekilde tahmin etmek güvenlik uygulamaları, reklamcılık, otonom araçlar vb. uygulama alanları için

önemli bir işlemdir. Bu nedenle derin öğrenme tabanlı yaya rotası tahmin çalışmaları araştırmacıların dikkatini çekmektedir. Bu çalışma kapsamında yaya rotası çalışmalarına optimizasyon yönteminin derin öğrenme eğitim aşamasına etkileri açısından inceleme yapılmıştır. İncelenmek üzere seçilen çalışmaların açık kaynak kodları kullanılarak, araştırmacılar tarafından yayınlanan parametreler ve optimizasyon yöntemi ile modeller eğitilmiş ve en iyi doğrulama kaybının elde edildiği dönem ve bu dönemdeki değerlendirme ölçütleri kaydedilmiştir. Daha sonra optimizasyon yöntemi AdaBelief olarak değiştirilerek eğitim süreci yeniden gerçekleştirilmiştir. Modeller doğrulama kaybı açısından incelendiğinde AdaBelief tekniğinin genel olarak daha hızlı düşüş sağladığı ve daha istikrarlı bir şekilde ilerlediği görülmektedir. Şekil 8, AdaBelief ve orijinal optimizasyon tekniği ile çalışmaların ETH veri kümesindeki doğrulama kaybı grafiklerini göstermektedir. STGAT, Casual-HTP ve SGCN uygulamalarında Adam kullanırken kaybın aşırı değişken olduğu dönemler olmasına rağmen AdaBelief'te istikrarlı durum korunmaktadır. Öte yandan, Social GAN Ayırıcı aşaması, Social-STGCN, Carpe Posterum ve SGCN uygulamalarında doğrulama kaybında daha hızlı azalma sağlanmaktadır. Bu sonuçlar AdaBelief'in, hızlı doğrulama kaybı elde etme yeteneğine sahip olduğunu göstermektedir.

Hızlı yakınsama açısından bakıldığında ihtiyaç duyulan dönem sayısını azaltması nedeniyle AdaBelief önemli katkı sağlamaktadır. İkinci momentum hesaplanırken hata fonksiyonunun eğimi ile birinci momentum değerinin birbirine çok yakın olduğu durumda büyük, diğer durumlarda daha küçük değişim değeri hesaplanmakta ve bu sayede hızlı yakınsama sağlayarak dönem sayısını azaltmaktadır. AdaBelief en başarılı sonucu Mesaj Geçişi ve İzomorfik Evrişimli Çizge Ağı tabanlı Carpe Posterum uygulamasında vermiştir. Carpe Posterum'un eğitim aşamasında, ihtiyaç duyulan dönem sayısı bu beş veri kümesinde ortalama %32 oranında azalmıştır. Tablo 2'de her iki optimizasyon tekniği için doğrulama veri kümesinde elde edilen en düşük ADE/FDE değerleri ile bu değerlerin elde edilip modelin kaydedildiği dönem numarası gösterilmektedir. Tabloda iyileştirme elde edilen değerler italik olarak gösterilmiştir.

Tablo 2'de görüldüğü gibi AdaBelief ile ya da ADE/FDE açısından ya da dönem sayısı açısından ya da her ikisi açısından önemli kazanımlar elde edilmiştir. Eğer çalışmalara erken kesme yöntemi eklenirse daha az sayıda dönem sayısı ile sonuca ulaşılabilecektir. Modellerde kullanılan hiper parametrelerde iyileştirmeler ile kazanımların artırılabilir düşünlümlenmektedir (Yang ve Shami 2020). Bununla birlikte GAT ve LSTM kullanan ve temelde aynı mimariye sahip olan STGAT ve Casual-HTP uygulamalarında önemli bir gelişme sağlanamamıştır. Yayaların sosyal etkileşim içerisinde hareket ettiği göz önünde bulundurularak rota tahmin çalışmalarında yayaların birbirini etkileme oranlarının hesaba katıldığı "dikkat tabanlı" çalışmaların ön plana çıkarılması ve eğitim aşamalarının iyileştirilmesi gerekmektedir.



Şekil 8. İncelenen çalışmaların ETH veri kümesinde AdaBelief ve orijinal optimizasyon yöntemiyle eğitilirken elde edilen doğrulama kaybı grafikleri

**Tablo 2.** Doğrulama veri seti için en düşük ADE ve FDE değerleri ve bu değerlerin elde edilmiş olduğu dönem sayısı.

| Yayın  | Veri Kümesi | ADE/FDE Yayın Optimizasyon | Dönem Sayısı | ADE/FDE AdaBelief | Dönem Sayısı |
|--|-------------|----------------------------|--------------|-------------------|--------------|
| <b>Social GAN, Kaynak kod: (SGAN 2023)</b>                 | Eth         | 0,31/0,63                  | 541          | 0,31/0,62         | 453          |
|  | Hotel       | 0,32/0,67                  | 656          | 0,32/0,66         | 687          |
|  | Univ        | 0,32/0,65                  | 607          | 0,31/0,62         | 728          |
|  | Zara1       | 0,32/0,67                  | 615          | 0,32/0,66         | 512          |
|  | Zara2       | 0,35/0,72                  | 562          | 0,34/0,70         | 311          |
| <b>STGAT, Kaynak kod: (STGAT 2023)</b>                     | Eth         | 1,01/2,00                  | 390          | 1,07/2,23         | 364          |
|  | Hotel       | 0,69/1,45                  | 367          | 0,67/1,40         | 398          |
|  | Univ        | 0,76/1,64                  | 368          | 0,78/1,69         | 273          |
|  | Zara1       | 0,54/0,94                  | 329          | 0,64/1,39         | 329          |
|  | Zara2       | 0,49/1,09                  | 374          | 0,50/1,10         | 393          |
| <b>Social-STGCNN, Kaynak kod:(SocialSTGCNN 2023)</b>       | Eth         | 0,37/0,63                  | 230          | 0,34/0,58         | 126          |
|  | Hotel       | 0,37/0,64                  | 246          | 0,35/0,60         | 152          |
|  | Univ        | 0,36/0,59                  | 195          | 0,32/0,55         | 135          |
|  | Zara1       | 0,38/0,61                  | 194          | 0,36/0,62         | 218          |
|  | Zara2       | 0,40/0,68                  | 220          | 0,41/0,69         | 235          |
| <b>SGCN, Kaynak kod: (SGCN 2023)</b>                       | ETH         | 0,29/0,51                  | 253          | 0,28/0,54         | 263          |
|  | Hotel       | 0,30/0,52                  | 232          | 0,30/0,53         | 148          |
|  | Univ        | 0,25/0,45                  | 228          | 0,24/0,43         | 291          |
|  | Zara1       | 0,30/0,55                  | 246          | 0,30/0,54         | 94           |
|  | Zara2       | 0,33/0,59                  | 78           | 0,33/0,58         | 93           |
| <b>CARP<sup>c</sup> Posterum, Kaynak kod: (CARPE 2023)</b> | Eth         | 0,44/0,89                  | 72           | 0,47/0,90         | 34           |
|  | Hotel       | 0,46/0,94                  | 59           | 0,48/0,96         | 35           |
|  | Univ        | 0,38/0,75                  | 58           | 0,40/0,78         | 66           |
|  | Zara1       | 0,45/0,92                  | 74           | 0,47/0,93         | 50           |
|  | Zara2       | 0,50/1,00                  | 53           | 0,51/0,98         | 27           |
| <b>Casual-HTP, Kaynak kod:(CausalHTP 2023)</b>             | Eth         | 1,00/1,98                  | 391          | 1,10/2,01         | 205          |
|  | Hotel       | 0,70/1,41                  | 207          | 0,69/1,43         | 316          |
|  | Univ        | 0,69/1,49                  | 214          | 0,74/1,60         | 295          |
|  | Zara1       | 0,60/1,36                  | 388          | 0,65/1,43         | 399          |
|  | Zara2       | 0,48/1,01                  | 210          | 0,51/1,08         | 235          |

#### 4. TARTIŞMA VE SONUÇ

Yaya tespiti, takibi ve rota tahmini işlemlerine 5. seviye otonom araçlar, şehir güvenliği, reklam vb. uygulamalarda, yaya kamera görüntüsüne girdiği anda hangi aksiyonun gerçekleştirilmesi gerektiğine karar verilmesi konusunda ihtiyaç duyulabilmektedir. Bu ihtiyacı karşılamak için günümüzde yaya rotası tahmin çalışmaları derin öğrenmeye dayalı olarak geliştirilmektedir. Derin öğrenmede, modeller için hızlı eğitim prosedürleri ve bu aşamada düşük işlem maliyetleri ön plana çıkmaktadır.

Bu çalışmada, yaya rota tahmin çalışmalarında AdaBelief optimizasyon tekniğinin kullanılmasının eğitim aşamasında “yakınsama” açısından kazanımları incelenmiştir. AdaBelief’in özellikle Çizge ve Evrişimli Ağ kullanılan modellerde tercih edilebilecek bir yöntem olduğu görülmüştür. Örneğin Mesaj Geçişi, İzomorfik Evrişimli Çizge Ağ ve Evrişimli Ağ tabanlı Carpe Posterum’da AdaBelief’in ortalama %32’lik bir kazançla

daha az dönem sayısı ile eğitim gerçekleştirebildiği gözlemlenmiştir. Ancak LSTM veya dikkat tabanlı yaya yolu tahmin uygulamalarında elde edilen görece az olan kazanımın modellere göre parametre ayarlamaları yapılarak arttırılabileceği düşünülmektedir.

Yaya rotasının tahmin edilmesinde çevredeki tüm nesnelerin yayanın takip edeceği yolu etkileyecektir. Bu nedenle dikkat tabanlı çalışmaların daha hassas ve hızlı eğitilmesine ihtiyaç duyulmaktadır. Bu nedenle, bundan sonraki aşamada etkilerin hesaba katılması için Dikkat Tabanlı Çizge Ağ, geçmiş rota bilgilerinin dahil edilebilmesi için Dönüştürücü Ağ kullanılan ve optimizasyon tekniği olarak AdaBelief’in tercih edildiği yaya rotası tahmin çalışması geliştirilecektir.

#### ÇIKAR ÇATIŞMASI

Yazarlar arasında çıkar çatışması yoktur.

#### KAYNAKLAR

Alahi A, Goel K, Ramanathan V, Robicquet A, Fei-Fei L, Savarese S (2016). Social LSTM: Human Trajectory Prediction in Crowded Spaces. IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, USA.



- Bera A, Kim S, Randhavane T, Pratapa S, Manocha D (2016). GLMP- realtime pedestrian path prediction using global and local movement patterns. IEEE International Conference on Robotics and Automation. Stockholm, Sweden.
- Bottou L (1991). Stochastic gradient learning in neural networks. Proceedings of Neuro-Nimes, 91(8): 12.
- CARPE (2023). [https://github.com/TeCSAR-UNCC/CARPe\\_Posterum](https://github.com/TeCSAR-UNCC/CARPe_Posterum).
- CausalHTP. (2023). <https://github.com/CHENGY12/CausalHTP>.
- Chen G, Li J, Lu J, Zhou J (2021). Human Trajectory Prediction via Counterfactual Analysis. IEEE/CVF International Conference on Computer Vision. Montreal, Canada.
- Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014). Generative adversarial nets. Advances in Neural Information Processing Systems, 27.
- Gulzar M, Muhammad Y, Muhammad N (2021). A Survey on Motion Prediction of Pedestrians and Vehicles for Autonomous Driving. IEEE Access, 9:137957–137969.
- Guo J, Li J, Leng D, Pan L (2021). Heterogeneous Graph based Deep Learning for Biomedical Network Link Prediction. arXiv preprint arXiv:2102.01649.
- Guo S, Fraser MW (2014). Propensity score analysis: Statistical methods and applications (Vol. 11). SAGE Publications.
- Gupta A, Johnson J, Fei-Fei L, Savarese S, Alahi A (2018). Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks. Salt Lake City, USA.
- Hariyono J, Shahbaz A, Jo K-H (2015). Estimation of walking direction for pedestrian path prediction from moving vehicle. IEEE/SICE International Symposium on System Integration. Nagoya, Japan.
- Hecht J (2018). Lidar for Self-Driving Cars. Optics & Photonics News, 28–33.
- Hochreiter S, Schmidhuber J (1997). Long Short-Term Memory. Neural Computation, 9(8): 1735-1780.
- Huang Y, Bi H, Li Z, Mao T, Wang Z (2019). STGAT: Modeling Spatial-Temporal Interactions for Human Trajectory Prediction. IEEE/CVF International Conference on Computer Vision. Seoul, Korea.
- Jain AB, Casas S, Liao R, Xiong Y, Feng S, Segal S, Urtasun R (2019). Discrete Residual Flow for Probabilistic Pedestrian Behavior Prediction. Conference on Robot Learning. Osaka, Japan.
- Keller CG, Gavrila DM (2014). Will the Pedestrian Cross? A Study on Pedestrian Path Prediction. IEEE Transactions on Intelligent Transportation Systems, 15(2): 494–506.
- Kingma DP, Ba J (2014). Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980.
- Kipf TN, Welling M (2016). Semi-Supervised Classification with Graph Convolutional Networks. arXiv preprint arXiv:1609.02907.
- Kolcu C, Bolat B (2018). Yayaların yürüyüş rotalarının belirlenmesi. Electric Electronics, Computer Science, Biomedical Engineering's Meeting (EBBT). İstanbul, Türkiye.
- Le Cun Y, Jackel LD, Boser B, Denker JS, Graf HP, Guyon I, Henderson D, Howard RE, Hubbard W (1989). Handwritten digit recognition: applications of neural network chips and automatic learning. IEEE Communications Magazine, 27(11): 41–46.
- Leinonen J (2021). Improvements to short-term weather prediction with recurrent-convolutional networks. IEEE International Conference on Big Data (Big Data). Orlando, FL, USA.
- Lerner A, Chrysanthou Y, Lischinski D (2007). Crowds by Example. Computer Graphics Forum, 26(3): 655–664.
- Liu Y, Zhang M, Zhong Z, Zeng X, Long X (2021). A comparative study of recently deep learning optimizers. International Conference on Algorithms, High Performance Computing, and Artificial Intelligence (AHPICAI 2021). Sanya, China.
- Lv Y, Zhou Q, Li Y, Li W (2021). A predictive maintenance system for multi-granularity faults based on AdaBelief-BP neural network and fuzzy decision making. Advanced Engineering Informatics, 49: 101318.
- Ma Y, Zhu X, Zhang S, Yang R, Wang W, Manocha D (2019). TrafficPredict: Trajectory Prediction for Heterogeneous Traffic-Agents. AAAI Conference on Artificial Intelligence. New Orleans, Louisiana, USA.
- Mendieta, M., & Tabkhi, H. (2021, May). Carpe posterum: A convolutional approach for real-time pedestrian path prediction. AAAI Conference on Artificial Intelligence. Vancouver, Canada.

- Mittal S, Vetter JS (2015). A Survey of Methods for Analyzing and Improving GPU Energy Efficiency. *ACM Computing Surveys*, 47(2): 1–23.
- Mohamed A, Qian K, Elhoseiny M, Claudel C (2020). Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Seattle, WA, USA.
- Ozyildirim BM, Kiran M (2020). Do optimization methods in deep learning applications matter? *arXiv preprint arXiv:2002.12642*.
- Pei D, Jing M, Liu H, Sun F, Jiang L (2020). A fast RetinaNet fusion framework for multi-spectral pedestrian detection. *Infrared Physics & Technology*, 105: 103178.
- Pellegrini S, Ess A, Schindler K, Van Gool L (2009). You'll never walk alone: Modeling social behavior for multi-target tracking. *2009 IEEE 12th International Conference on Computer Vision*. Kyoto, Japan.
- Rudenko A, Palmieri L, Herman M, Kitani KM, Gavrila DM, Arras KO (2019). Human Motion Trajectory Prediction: A Survey. *The International Journal of Robotics Research*, 39(8): 895-935.
- SGAN (2023). <https://github.com/agrim Gupta92/sgan>.
- SGCN (2023). <https://github.com/shuaishiliu/SGCN>.
- Shi H, Wang L, Scherer R, Wozniak M, Zhang P, Wei W (2021). Short-Term Load Forecasting Based on Adabelief Optimized Temporal Convolutional Network and Gated Recurrent Unit Hybrid Neural Network. *IEEE Access*, 9: 66965–66981.
- Shi, L., Wang, L., Long, C., Zhou, S., Zhou, M., Niu, Z., & Hua, G. (2021). SGCN: Sparse graph convolution network for pedestrian trajectory prediction. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Nashville, TN, USA.
- SocialSTGCNN (2023). <https://github.com/abdullahmohamed/Social-STGCNN>.
- STGAT (2023). <https://github.com/huang-xx/STGAT>.
- Turan S, Milani B, Temurtaş F (2021). Different Application Areas Of Object Detection With Deep Learning. *Akıllı Ulaşım Sistemleri ve Uygulamaları Dergisi*. 4(2): 148-164.
- Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y (2017). Graph Attention Networks. *arXiv preprint arXiv:1710.10903*.
- Wang H, Li C, Zhang Y, Liu Z, Hui Y, Mao G (2020). A Scheme on Pedestrian Detection using Multi-Sensor Data Fusion for Smart Roads. *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. Antwerp, Belgium.
- Wang Y, Liu J, Chang X (2021). Generalizing Adversarial Examples by AdaBelief Optimizer. *arXiv preprint arXiv:2101.09930*.
- Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS (2019). A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1): 4-24.
- Xue-Wen Chen, Xiaotong Lin (2014). Big Data Deep Learning: Challenges and Perspectives. *IEEE Access*, 2: 514–525.
- Yang L, Shami A (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415: 295–316.
- Zhang S, Tong H, Xu J, Maciejewski R (2019). Graph convolutional networks: a comprehensive review. *Computational Social Networks*, 6(1): 11.
- Zhou D, Qiu S, Song Y, Xia K (2020). A pedestrian extraction algorithm based on single infrared image. *Infrared Physics & Technology*, 105: 103236.
- Zhou J, Cui G, Hu S, Zhang Z, Yang C, Liu Z, Wang L, Li C, Sun M (2020). Graph neural networks: A review of methods and applications. *AI Open*, 1: 57–81.
- Zhu Z, Sun H, Zhang C (2021). Effectiveness of Optimization Algorithms in Deep Image Classification. *arXiv preprint arXiv:2110.01598*.
- Zhuang J, Tang T, Ding Y, Tatikonda S, Dvornik N, Papademetris X, Duncan JS (2020). AdaBelief Optimizer: Adapting Stepsizes by the Belief in Observed Gradients. *Advances in Neural Information Processing Systems*, 33: 18795-18806.