Araştırma Makalesi
Adıyaman Üniversitesi
Mühendislik Bilimleri Dergisi
16 (2022) 248-262

DİNAMİK ORTAMLARDA DERİN TAKVİYELİ ÖĞRENME TABANLI OTONOM YOL PLANLAMA YAKLAŞIMLARI İÇİN KARŞILAŞTIRMALI ANALİZ

Ziya TAN[1*], Mehmet KARAKÖSE[2]

[1]Erzincan Üniversitesi, Kemaliye Hacı Ali Akın Meslek Yüksekokulu, Erzincan, 24600, Türkiye
[2]Fırat Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Elazığ, 23000, Türkiye
Geliş Tarihi/Received Date: 18.11.2021 Kabul Tarihi/Accepted Date: 04.03.2022 DOI: 10.54365/adyumbd.1025545

ÖZET

Takviyeli öğrenme, içinde bulunduğu ortamı algılayan ve kendi kendine kararlar verebilen bir sistemin, mevcut problemin çözümünde doğru kararlar almayı nasıl öğrenebileceği bir yöntemdir. Bu makalede, bir robotun haraketli engellerin(yayalar) olduğu bir ortamda engellere çarpmadan belirtilen alanda otonom bir şekilde hareket etmeyi öğrenmesi için derin takviyeli öğrenme tabanlı bir algoritma önerilmektedir. Oluşturulan simülatör ortamında derin öğrenme algoritmalarından Convolutional Neural Network(CNN), Long-short Term Memory(LSTM) ve Recurrent Neural Network(RNN) ayrı ayrı kullanılıp performansları test edilerek raporlanmıştır. Buna göre bu makale kapsamında literatüre üç önemli katkı sunulmaktadır. Birincisi etkili bir otonom robot algoritmasının geliştirilmesi, ikincisi probleme uygun olarak uyarlanabilen derin öğrenme algoritmasının belirlenmesi, üçüncü olarak otonom bir robotun hareketli engellerin olduğu kalabalık ortamlardaki hareket eylemini gerçekleştirmesi için genelleştirilmiş bir derin takviyeli öğrenme yaklaşımının ortaya konulmasıdır. Geliştirilen yaklaşımların doğrulanması için derin takviyeli öğrenme algoritmaları ayrı ayrı simüle edilerek eğitimi gerçekleştirilmiştir. Yapılan eğitim sonuçlarına göre, LSTM algoritmasının diğerlerinden daha başarılı olduğu tespit edilmiştir.

Anahtar Kelimeler: *Derin Takviyeli Öğrenme, Otonom Yol Planlama, Derin Öğrenme, LSTM, RNN*

COMPARATIVE ANALYSIS FOR AUTONOMOUS PATH PLANNING APPROACHES BASED ON DEEP REINFORCEMENT LEARNING IN DYNAMIC ENVIRONMENTS

ABSTRACT

Reinforcement learning is a method of how a system that perceives its environment and can make decisions on its own can learn to make the right decisions in solving the current problem. In this article, a deep reinforcement learning-based algorithm is proposed for a robot to learn to move autonomously in a specified area without hitting obstacles in an environment with moving obstacles (pedestrians). Convolutional Neural Network (CNN), Long-short Term Memory (LSTM) and Recurrent Neural Network (RNN), which are deep learning algorithms in the created simulator environment, are used separately and their performances are tested and reported. Accordingly, three important contributions to the literature are made within the scope of this article. The first is the development of an effective autonomous robot algorithm, the second is the determination of a deep learning algorithm that can be adapted to the problem, and the third is a generalized deep reinforcement learning approach for an autonomous robot to perform the movement action in crowded environments with moving obstacles. In order to verify the developed approaches, deep reinforcement learning algorithms were separately simulated and trained. According to the training results, it has been determined that the LSTM algorithm is more successful than the others.

Keywords: *Deep Reinforcement Learning, Autonomous Path Planning, Deep Learning, LSTM, RNN*

[*] e-posta[1]: ziyatan@erzincan.edu.tr ORCID ID: https://orcid.org/0000-0003-2813-5882 (Sorumlu Yazar)
e-posta[2] : mkarakose@firat.edu.tr ORCID ID: https://orcid.org/0000-0002-3276-3788

1. Giriş

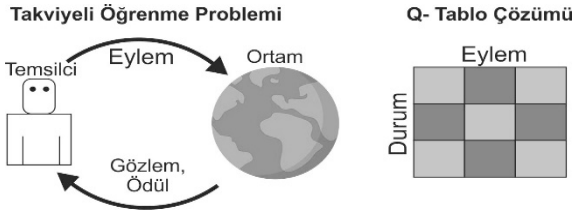
Yapay zekâ, makinelerin edindiği tecrübelerden öğrenmesini, yeni girdilere uyum sağlamasını ve insan benzeri görevleri gerçekleştirmesini mümkün kılar. Yapay zekâ, görevleri yerine getirmek için insan beynini modelleyerek taklit eden ve topladıkları bilgilere göre yinelemeli olarak kendini geliştirebilen sistemlerdir. Günümüzde bu sistemler satranç oynayan bilgisayarlardan kendi kendini geliştirebilen robotlara kadar kullanılmaktadır. Özellikle derin öğrenme ve takviyeli öğrenme yöntemleriyle geliştirilen yaklaşımlar teknolojik gelişmelerde önemli rol oynamaktadır.

Bu yaklaşımlar, bir temsilciye gereksinim duyulan, belirsiz özelliklere sahip birden fazla koşul altında bir görevi yerine getirme problemini çözmek için kullanılır. Aynı zamanda bilinmeyen etkenlerin bulunduğu karmaşık bir ortamda eğitim seti olmaksızın sonuç almak oldukça zordur [1]. Başka bir şekilde ifade etmek gerekirse, temsilci, kendine verilen bir görevi tamamlamaya çalışan akıllı bir sistem tarafından gerçekleştirilen doğru bir eylemdir. Bu eylemi Markov Karar süreçleri (MDP) [2] olarak adlandırılan modelin karar verme süreci olarak tanımlanmaktadır.

Takviyeli öğrenmedeki en önemli gelişmelerden biri, Q-öğrenme olarak bilinen bir politika dışı Zamansal-fark (Temporal Different) kontrol algoritmasının geliştirilmesiydi [3].Temel olarak tek aşamalı Q-öğrenme, denklem 1 de gösterildiği gibi tanımlanır:

$$(S_T, A_T) \leftarrow Q(S_T, A_T) + \alpha[\ R_{t+1} + \gamma max_a Q(S_{t+1}, a) - \ Q(S_T, A_T)\ ] \quad (1)$$

Q-öğrenme gibi takviyeli öğrenme yöntemleri stokastik algoritmalara göre daha güvenilirdir, çünkü durum(state) geçişlerinde bir ödül alarak mevcut eylemi ödüllendirir. MDP teorisinin temeline dayanan Q-öğrenme, farklı karmaşık koşullar altında temsilci eylemlerinin sorunlarını çözmek için ideal bir yöntemdir. Q-öğrenmede kullanılan tablo matrisi(Q-Table), durum ve hareket değerlerinin artmasıyla çok büyük bir boyuta gelir. Bu durum en büyük dezavantajlarından biridir ve temsilcinin yönlendirilmesinde yanlış kararlar vermesine yol açar. Bu tür durumlarda yaygın olarak kullanılan yapay sinir ağları ile sorun en aza indirilmektedir. Dolayısıyla araştırmacılar tarafından daha çok tercih edilmektedir. Q-Öğrenmenin yapısı şekil 1 de gösterilmiştir.



Şekil 1. Q-öğrenme modeli

Derin takviyeli öğrenme(DRL), özellikle son yıllardaki gelişmelerle birlikte araştırmacılar tarafından tercih edilmektedir. DeepMind ekibinin Atari oynayabilen temsilciyi eğitmeleri 2010 yılının en dikkat çeken gelişmelerinden olmuştur. Aynı yıllarda Go oyununda dünya şampiyonunu yenebilen bir temsilci eğitildi. Bu gelişmelerin ardından takviyeli öğrenme çalışmaları hızla ilerledi. 2019 yılında Dota [4] oyunu derin takviye öğrenmesi kullanılarak geliştirildi. Starcraft II [5] , Quake III [6], gibi karmaşık ve çok oyunculu oyunlarda derin takviye öğrenimi yaklaşımı kullanılarak geliştirildi. Ayrıca uzun süreli bellek gerektiren görevlerde başarılı sonuçlar alan derin takviyeli öğrenmeyi destekleyen mimariler geliştirildi [7] [8].Sağlık alanında DRL nin kullanıldığı çalışma [9], Rubik küpün çözümünde [10], DRL'nin ses alanındaki ilerleyişi hakkında ayrıntılı bir çalışma hazırlamışlardır [11]. Benzer başka bir çalışmada ise konuşma duygu tanımada daha önceden eğitilmiş derin öğrenme algoritmaları kullanarak DRL yaklaşımı sunulmaktadır [12].Otonom robot navigasyonu ile ilgili geliştirilmiş bir öğrenme yöntemi sunan çalışma [13] sürekli hareket halindeki karmaşık robot problemlerinde çalışmalar yapılmıştır. Düşman radar tespiti ve füze saldırısı altında İHA'nın hayatta kalma olasılığı dikkate alınarak bir durum değerlendirme modelinin geliştirildiği makalede [14] DRL tabanlı yol

planlama problemi ele alınmaktadır. Robotlar için bilinmeyen ortamlarda otonom yol planlama probleminin çözümü için derin takviyeli öğrenme algoritmaları kullanılmıştır [15]. Bilinmeyen bir ortamda insansız gemilerin akıllı yol planlamasını gerçekleştirmek için DRL'ye dayalı otonom bir yol planlama modeli önerilen makalede [16], model, çevre ile sürekli etkileşim ve geçmiş deneyim verilerinin kullanımı yoluyla derin deterministik politika gradyanı (DDPG) algoritmasını kullanmaktadır. Bilinmeyen bir ortamda quadrotorun otonom olarak uçabilmesi için yeni, açıklanabilir derin sinir ağı tabanlı bir yol planlayıcı önerilen çalışmada [17] navigasyon problemi bir Markov Karar Süreci (MDP) olarak modellenmiş ve yol planlayıcı, simülasyon ortamında Derin Takviyeli Öğrenme yöntemi kullanılarak eğitilmiştir.

Çizelge 1. Yol planlaması için yapılan bazı çalışmaların karşılaştırılması

Makale	Problem	Simülator	Çözüm Yöntemi
[13]	Robot için Otonom yol planlama	Gazebo	Derin Takviyeli Öğrenme (PPO)
[14]	Tehdit altındaki İHA için kendini koruma olasılığına karşın yol planlama	The STAGE Scenario	Derin Takviyeli Öğrenme (D3QN, DDQN, DQN)
[15]	Otonom robot navigasyonu	Gazebo	Derin Takviyeli Öğrenme (D3QN)
[16]	İnsansız gemiler için otonom yol planlama	Visual Studio 2013	Derin Takviyeli Öğrenme (DDPG)
[17]	İHA için otonom yol planlama	AirSim	Derin Takviyeli Öğrenme (DDPG)

Çizelge 1'de gösterilen çalışmalar incelendiğinde farklı simülasyon ortamları ve araçlar kullanılsa da derin takviyeli öğrenmenin otonom yol planlamasındaki katkısı görülmektedir. Bu katkının daha karmaşık problemlerde de görülmesi mümkündür.

Son yıllarda, özellikle otomotiv ve robotik alanlarda kendi kendine hareket edebilen otonom sistemlerin gelişimi, akıllı sistemlerin başarısının geldiği noktayı ortaya koymaktadır. Otonom sistemlerde yol ve trafik durumu çeşitli sensörlerle desteklenmekte ve modellenmektedir. Sensörlerden gelen ham verileri anlamlı hale getirip sistemin yol planlamasına karar vermektedir.

Bu makalede, hareketli engellerin olduğu bir ortam oluşturulmuş ve temsilcinin bu engellere çarpmadan hareket etmesi hedeflenmiştir. Günümüzde, fabrikalarda, insan trafiğinin yoğun olduğu iş merkezleri veya alışveriş merkezlerinde otonom robotlara rastlamak mümkündür. Bunlar görevlerini yerine getirirken birçok hareketli ya da sabit engellerle karşılaşmaktadır. Bu tür durumlarda bu engellere çarpmadan hareket etmeleri gerekmektedir. Ancak başarılı bir eğitim bu kazaları en aza indirecektir. Temsilcinin eğitim aşamasında derin takviyeli öğrenme yaklaşımı kullanılmıştır. Bu aşamada derin öğrenme algoritmalarından CNN, LSTM ve RNN algoritmaları ayrı ayrı uygulanarak sonuçlar raporlanmıştır. Bu makalenin başlıca katkılarını aşağıdaki gibi özetleyebiliriz.

- Bir robotun hareketli engellerin olduğu ortamlarda engellere çarpmadan hareketini gerçekleştirmesi,
- Otonom yol planlamasında kullanılacak olan derin takviyeli öğrenme algoritmasının geliştirilmesi,
- Derin öğrenme algoritmalarının otonom yol planlamasındaki başarılarının karşılaştırılması

Bu makalenin geri kalanı belirtilen şekilde sunulmuştur. Bölüm 2, makalede kullanılan algoritmalar ve derin takviyeli öğrenme yaklaşımları tanımlanmıştır. Bölüm 3'te yapılan çalışmayla alakalı önerilen yaklaşım anlatılmaktadır. Bölüm 4'te simülasyon sonuçları değerlendirilmektedir. Son bölümde ise makalenin sonuç bölümü bulunmaktadır.

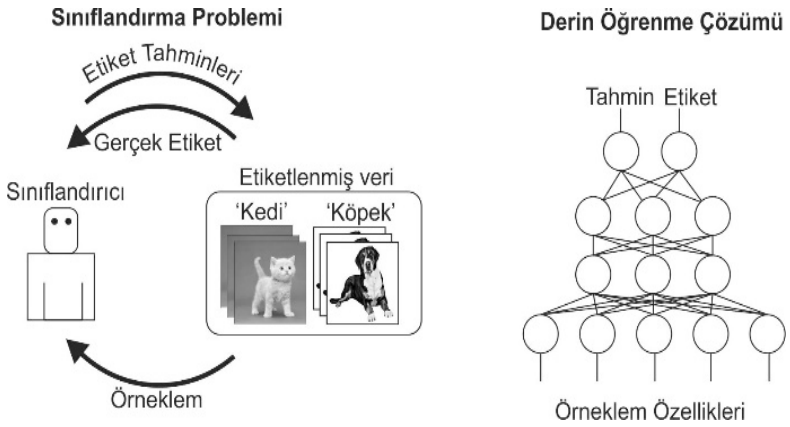
2. Derin Takviyeli Öğrenme

Bu bölümde çalışmada kullanılan derin takviye öğrenimi yaklaşımı ve kullanılan derin öğrenme algoritmaları açıklanmıştır.

2.1. Derin Öğrenme

Derin öğrenme [18] algoritmaları, özellik ve görevleri doğrudan giriş verisinden öğrenen bir makine öğrenme tekniğidir. Veriler ses, görüntü veya metinden oluşabilir. Derin öğrenme sistematik olarak yapay zekâ ve makine öğrenmesinin en altında yer almaktadır ve yapay zekâ uygulamalarında en popüler olan yaklaşımlarından biridir [19].

Derin öğrenme algoritmaları, öğrenme işlemi aşamasında etiketlenmiş bir örüntü üzerindeki tüm pikselleri giriş verisi olarak kullanır. Giriş verisi olarak verilen görüntü renkli ise giriş parametreleri 3 katı artar, eğer gri tonlu bir görüntü ise pixel değerleri kadar giriş verisi hesaplanır. Birincil olarak konvolusyonel katmanları, ReLu(Rektifiye Doğrusal Birim) katmanları ve havuzlama katmanları uygulanarak görüntü üzerindeki özellik haritası çıkarılmaktadır. Özellik haritası çıkarıldıktan sonra tam bağlantılı katmanlar ve Softmax katmanından meydana gelen bir dizi sınıflandırma katmanları kullanılmaktadır [20]. Bu aşamada her pikselin her bir sınıfa ait ihtimal değerleri hesaplanmaktadır. Oluşan bu değerler Softmax katmanı tarafından ilgili pikselin hangi sınıfa ait olduğunu belirlemektedir [21]. Şekil 2'de derin öğrenmenin çalışma prensibi gösterilmiştir.

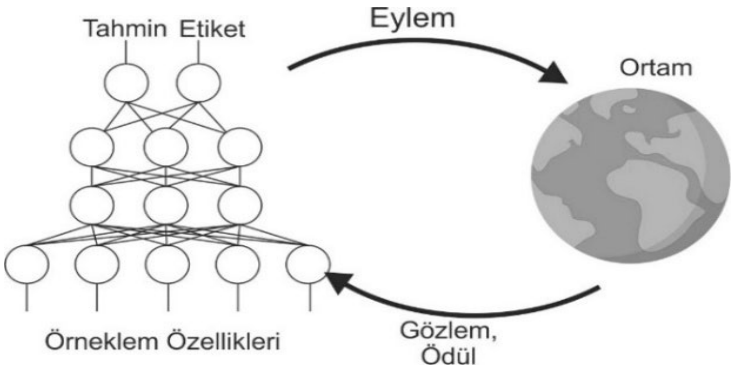


Şekil 2. Derin öğrenme modeli

2.2. Derin Takviyeli Öğrenme

Derin takviyeli öğrenme [22], takviyeli öğrenmede meydana gelen problemleri çözmek için derin öğrenmeden faydalanmaktadır. Alışagelmiş derin takviyeli öğrenme sistemleri algısal girdilerden eylem değerlerine veya eylem olasılıklarına kadar doğrusal olmayan bir eşleşmeyi hesaplamak için takviye öğrenme sinyalleri kullanılır. Bunun yanı sıra daha iyi ödül tahminleri üretmek veya yüksek oranda ödüllendirilen eylemlerin sıklığını artırmak için bu ağdaki ağırlıkları genellikle geri yayılım

yoluyla güncelleyen derin sinir ağı kullanır [23]. Şekil 3'te derin takviyeli öğrenme modeli gösterilmiştir.



Şekil 3. Derin takviyeli öğrenme modeli

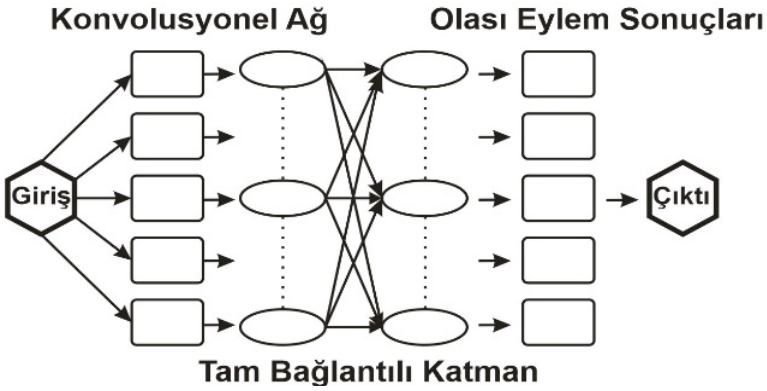
Q-öğrenme, temsilcimiz için oluşturulan Q-tablo sayesinde basit ama oldukça güçlü bir algoritmadır. Bu tablo, temsilcinin tam olarak hangi eylemi gerçekleştireceğini anlamasına yardımcı olur.

Takviyeli öğrenme, öğrenmeyi bir MDP ile temsil edilebilen kendi kendine öğretilen bir süreç olarak tanımlanır. Bir MDP'yi 4 farklı(S,A,P_a,R_a) parametreyle özetlenmektedir [24]:

S, mevcut durumlar dizisi, A, mevcut eylemler dizisi, $P_a(s_t, s_{t+1})$, s_t durumundaki bir a eyleminin s_{t+1} durumuna yol açma ihtimalidir. Temsilci tarafından gerçekleştirilen a eylemine göre s_t ve s_{t+1} durumları ortam tarafından belirlenir. $R_a(s_t, s_{t+1})$, a eylemini gerçekleştiren bir s_t durumundan s_{t+1} durumuna geçişte alınan anlık beklenen ödüldür.

Q-Tablosunun çok büyük olması durumunda temsilci için karar vermek çok zorlaşacaktır. Mevcut keşfedilmiş durumlardan yeni durumların Q değerini çıkarılamayacağından bazı sorunlar ortaya çıkacaktır. Ortaya çıkan en önemli sorunlardan biri, boyutu normalden çok büyük olan Q-Tablosunu kaydetmek ve güncellemek için gereken bellek miktarıdır. Bir diğer sorun ise gerekli Q-Tablosunu oluşturmak için her durumu keşfetme aşamasında geçen süre gerçekçi olmayacaktır. Bu durumda, Q değerlerini hesaplarken yapay sinir ağı kullanmak sorunun çözümünde büyük başarı sağlayacaktı.

Derin Q-öğrenme yaklaşımında, Q-değer fonksiyonuna yaklaşmak için bir sinir ağı kullanılmaktadır. Sinir ağı, durumu girdi olarak alır ve tüm olası eylemlerden Q-değeri çıktı olarak üretilir. Temel amaç temsilcinin bulunduğu ortamdan ödülleri toplamak ve puanı en üst düzeye çıkarmaktır [25].Şekil 4'te Derin Q- Ağının mimarisi gösterilmektedir.



Şekil 4. Derin Q-ağı mimarisi

Temsilci her döngüde modelin ağırlıklarını günceller. Çizelge 2'de Derin Q-Öğrenme algoritmasının öğrenme aşamasındaki sözde kodu verilmektedir.

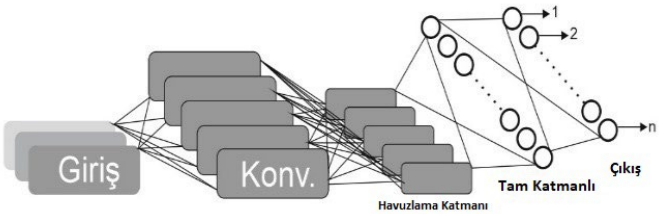
Çizelge 2. Derin q-öğrenme algoritmasının sözde kodu

1	Hafıza yenilenir
2	Rastgele ağırlıklarla network yenilenir
3	Her bir adım için:

-Başlangıç durumu ayarlanır
-Her zaman adımı için:
-Eylem Seçilir
-Keşif veya sömürü karar verilir
-Eylemi gerçekleştir
-Ödülü gözlemle ve bir sonraki duruma git
-Deneyimi ve hafızayı kaydet
-Network ağırlıklarını güncelle

2.3. Evrişimli Sinir Ağı (CNN)

Konvolusyonel sinir ağları (CNN) çoklu dizi biçiminde gelen verileri işlemek için tasarlanmıştır. Doğal sinyallerin özelliklerinden yararlanan bu ağların arkasında dört farklı temel yaklaşım vardır. Bu temel yaklaşımlar; yerel bağlantılar, paylaşılan ağırlıklar, havuzlama ve birçok katmanın kullanımından oluşmaktadır. Şekil 3. de gösterildiği gibi klasik bir CNN mimarisi ardışık birkaç katmandan oluşmaktadır [18]. Evrişimli bir katmandaki değerler özellik haritalarında düzenlenir. Burada her birim bir takım filtre yardımıyla bir diğer katmanın özellik haritalarındaki yerel yamalara bağlanır. Bu katmanlarda oluşan ağırlık değerleri daha sonra doğrusal olmayan bir fonksiyondan geçirilir. Evrişimli katmanın bir önceki katmandaki özelliklerin niteliklerini saptamak olsa da, havuzlama katmanın görevi anlamsal olarak benzer özellikleri tek bir nesnede birleştirmektir [26]. Havuzlama katmanları belirlenen filtre tarafından bir önceki katman üzerinde belli adımlarla kaydırılarak verileri alır ve böylece özellik matrisinin boyutu azaltır. CNN'ler temelde, konvolusyonel katmanı, havuzlama katmanı ve tam bağlı katman oluşur. Bu katmanların birden fazla kullanılması mümkündür. Bir klasik CNN mimari yapısı Şekil 5' te gösterilmiştir.



Şekil 5. Klasik CNN mimarisi

2.4. Tekrarlayan Sinir Ağları (RNN)

RNN, düğümler arasındaki bağlantıların zamansal bir dizi boyunca yönlendirilmiş bir döngü oluşturduğu bir yapay sinir ağları sınıfıdır [27]. İleri beslemeli sinir ağlarından üretilen RNN ler değişken boyuttaki girdileri için hafıza kullanırlar [28].

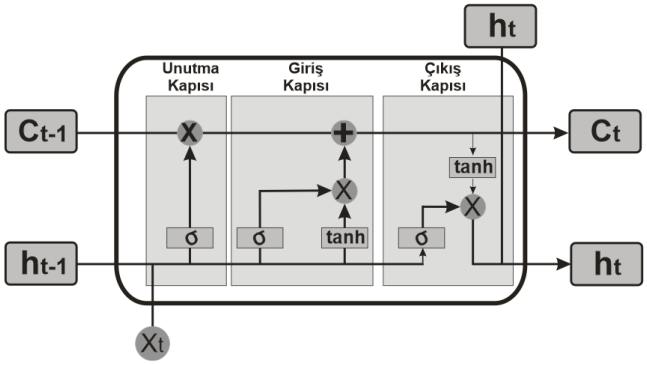
Geri yayılım(backpropagation) ilk kullanılmaya başlandığında RNN için heyecan vericiydi. RNN genelde dil işleme ve konuşma gibi sıralı girdileri içeren problemlerin çözümünde daha başarılıdır [29]. RNN'lerin en büyük dezavantajı eğitim aşamasında geri yayılım yapan gradyanların her adımda boyutunun fazla oranda değişmesi nedeniyle kaybolmaların veya patlamalarına sebep olabilir. Mimarideki ilerlemeler ve onları eğitme yolları sayesinde metinlerdeki bir sonraki karakteri veya devam eden kelimeyi tahmin etmede çok başarılı olmuştur [30]. RNN için kullanılan sinir ağları diğer sinir ağlarına göre farklılık gösterir. En önemli farklardan biri geçici bir hafızaya sahip olmasıdır [31]. Şekil 6'da RNN algoritmasının yapısı gösterilmektedir.

Şekil 6. Temel RNN mimari yapısı

2.5. Uzun Kısa Vadeli Hafıza Ağları (LSTM)

LSTM [32], kaybolan gradyan problemini önleyen derin öğrenme alanında kullanılan bir RNN mimarisidir. Alışagelmiş ileri beslemeli sinir ağlarının aksine, LSTM' in yalnızca tek veri noktalarını değil, tüm veri dizilerini işleyebilen geri besleme bağlantıları bulunmaktadır [33].

LSTM Sepp Hochreiter ve Juergen Schmidhuber tarafından 1997 yılında kaybolan gradient problemine çözüm bulmak için geliştirilmiştir [29].RNN mimarilerinde önceki bilgi kullanımına dayalı bir yaklaşım vardır. Mimarisi ayrıntılı bir şekilde incelendiğinde tüm katmanların aynı ağırlıkları paylaştığı ileri beslemeli ağlar olarak değerlendirilebilir. Temel amaç bu ağırlıkları uzun süre hafızada tutmak olsa da bunun ne kadar zor olduğunu yapılan deneylerle kanıtlamışlardır. [34]. Bu problemin çözümündeki en büyük fikirlerden biri RNN ağını açık bir hafıza ile genişletmektir. Bu yaklaşıma yönelik ilk öneri gizili birimler kullanan LSTM ağları olmuştur [35].Basit bir LSTM yapısında bir hücre, giriş kapısı, çıkış kapısı ve unut kapısı bulunmaktadır. Bu hücreler sisteme giren bilgi giriş çıkışını düzenlemektedir [36].



Şekil 7. LSTM hücresinin mimarisi

Şekil 7' de LSTM hücresinin mimari yapısı görülmektedir. Buna göre LSTM yapısının giriş verisi olarak bilginin herhangi bir değişikliğe uğramadan bir sonraki adıma geçmesini sağlayan hücre durumunu C_t olarak tanımlanır. C_t, isteğe bağlı olarak üç farklı kapı tarafından kontrol edilir. İlk kapı unutma kapısıdır. Unutma kapısı, hücre durum vektörü C_{t-1} tarafından hangi değerlerin unutulacağını kontrol eder. İkinci kapı olan giriş kapısı, hangi değerlerin güncelleneceğine karar verir. Üçüncü kapı ise sigmoid fonksiyonu tarafından çıkış değerinin hangisi olacağına karar verir [37].

Yapay Sinir Ağları çok güçlü bir tekniktir ve görüntü tanıma ve diğer birçok uygulama için kullanılır. Dezavantajlarından biri, modelle ilişkili bir belleğin olmamasıdır. Bu, metin veya zaman serileri gibi sıralı veriler için bir sorundur. RNN, bir tür bellek işlevi gören bir geri bildirim görünümü ekleyerek bu sorunu giderir. Böylece modele geçmiş girdiler bir ayak izi bırakır. LSTM, hem kısa vadeli hem de uzun vadeli bir bellek bileşeni oluşturarak bu fikri genişletir. Bu nedenle, LSTM, dizisi olan her şey için iyi sonuç veren bir ağdır. Çünkü bir kelimenin anlamı kendisinden önce gelenlere bağlıdır. Bu sayede, doğal dil işleme ve anlatı analizinin Sinir Ağlarından yararlanma yolunu açmıştır. LSTM, metin üretimi için kullanılabilir. Modeli bir yazarın metni üzerinde eğitebilir ve yazarın stilini ve ilgi alanlarını taklit eden yeni cümleler oluşturulabilir.

3. Önerilen Yaklaşım

Bu bölümde, ikinci bölümde ayrıntılı bir şekilde anlatılan üç farklı derin öğrenme algoritmasının hareketli engellerin olduğu bir ortamda ayrı ayrı eğitilerek başarılarının karşılaştırılması işlenmektedir. Takviyeli öğrenme ile derin öğrenmenin bir arada kullanıldığı derin takviyeli öğrenme yaklaşımı robotların eylemlerini seçmede temsilcilere sınırsız bir hareket ortamı sunmaktadır. CNN, LSTM ve RNN algoritmaları derin öğrenme problemlerinde başarılı sonuçlar vermiştir. Bu başarının, takviyeli öğrenme ile birleştirilerek hareketli engellerin olduğu bir simülasyon ortamında test edilmesi sağlanmıştır. Robotun engellere çarpmadan hareket etmeyi öğrenmesi amaçlanmıştır. Özellikle LSTM mimarisi gibi geri besleme yapısından ve kısa süreli hafızaya sahip olmasından dolayı bir önceki aşamadaki deneyimleri de unutmadığı için otonom şekilde gezinen robotlar için daha başarılı olacağı düşünülmektedir. Bu simülatörde kullanılan robota bu açıdan kattığı kazanımlar yadırganamaz.

Bu çalışmanın literatüre en büyük katkılarından biri LSTM ve RNN gibi dil işlemede başarısı kanıtlanmış derin öğrenme algoritmalarının bu tür problemlerde de uygulanabileceği ve başarılı sonuçlar elde edilebileceğidir.

3.1. Keşif Ve Sömürü Tercihi

Aslında günlük hayatımızda da sık sık karşılaştığımız bir durumdur. Bir şeyi yaparken bildiğim şekilde mi devam edeyim mi yoksa başka bir şey denemeliyim diye kararsız kalabiliriz. Takviyeli öğrenmede, eğer yaptığımız bir şeyi devam ettiriyorsak 'sömürü', yeni bir şey yapıyorsak 'keşif' olarak tanımlanır. Bu durumun hangi yönde ne kadar devam edileceği belirlenmektedir. Açgözlü (Ɛ –Greedy) politikasını bu deneyimlerden en yüksek kazancı elde etmek için kullanılmaktadır.

3.2. Açgözlü (Ɛ –Greedy)

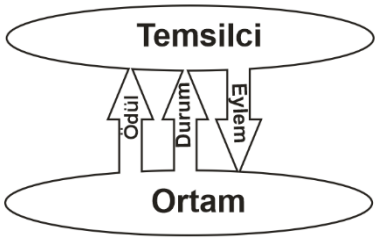
Bu çalışmada, Ɛ –Açgözlü davranış politikası tercih edilmiştir. Başlangıç değeri 1 olarak belirlenmiştir. Temsilcinin eylemini belirlemek için bu değer belli oranda azaltılmıştır. Bu değerler Çizelge 4'te gösterilmektedir.

$$a = \begin{cases} a, & 1-\varepsilon \\ rastgele\ eylem, & \varepsilon \end{cases} \quad (2)$$

Denklem (2) de Ɛ, temsilcinin öğrenme aşamasındaki keşif olasılığını gösterir.

3.3. Temsilci-Ortam Etkileşimi

Son yıllarda yapay sinir ağlarının bilgisayarlı görme, konuşma tanıma gibi problemlerde geleneksel makine öğrenimi yaklaşımlarında başarılı olduğu görülmüştür [38]. Ayrıca aynı sinir ağlarının başarılı sonuçlar almak ve daha fazla ödül toplamak için takviyeli öğrenme uygulamalarında da umut verici sonuçlar verdiği gözlemlenmiştir. Bu durumda yapay sinir ağları temsilci görevini üstlenir ve ortamdaki her adımı bir ödülle ödüllendirir [39]. Bir başka deyişle buna derin takviyeli öğrenme denmektedir. Şekil 8'de temsilci-ortam etkileşimi gösterilmektedir.



Şekil 8. Temel takviyeli öğrenme yapısı

Toplanan her ödül, eylemleri gerçekleştiren temsilcinin bir sonraki hareketi için öğrenme kriteridir. Ödüllerin toplanmasındaki esas eylem, temsilcinin politikaya (π) göre hareket etmesidir. Bu nedenle, RL sorununu formüle ederken, çevreden maksimum ödülü getiren bir ilke tanımlanmalıdır. MDP, mevcut durumların yalnızca son duruma (s') bağlı olduğunu ve geçmiş tüm durumlara bağlı olmadığını belirtir [40].

Takviye öğrenme algoritmalarında temel felsefe, istenilen eylemin gerçekleşmesi için bir eylemin ödül veya cezasını bir yöntemle düzenlemektir [41].

Eylem (a) = Temsilcinin her adımda aldığı ödüle göre geçerli politikanın belirlediği hareketi temsil etmektedir. Bu çalışmada 4 farklı eylem bulunmaktadır: ileri git, geri git, sağa git ve sola git.

Durum (s): Temsilcinin aktif adımda ortamdaki konumunu belirtmektedir.

Ödül (R): Temsilcinin her adımda ortamla olan etkileşiminden aldığı ödüldür. Bu değer temsilcinin daha sonraki adımlarında hangi eylemi seçmesinde rol oynar.

Ortam: Temsilcinin kendini eğitmek için gerekli duyduğu fiziksel ya da sanal ortamlardır. Ortamları başlıca stokastik ve deterministik olarak ikiye ayırabiliriz. Deterministik ortamda, temsilcinin durumu ve seçilen eylem ortamın bir sonraki durumu bilinmektedir. Temsilcinin herhangi bir belirsizlik için endişelenmesine gerek yoktur. Stokastik ortamda ise rastgelelik vardır ve bir sonraki eylem belirlenemez [42].

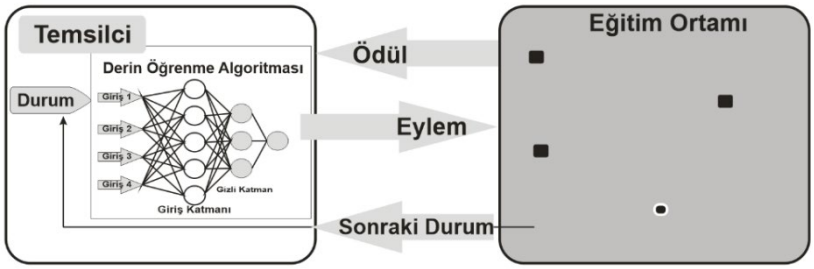
Politika (π): Politikanın temel amacı, ortamda bulunan temsilcinin davranışlarını belirlemektir. Temsilcinin hedefi, maksimum ödülü alacak politikayı seçmektir [43].

İndirim Faktörü (γ): Bir temsilcinin ödül fonksiyonunu belirlemede kullanılır. İndirim faktörü anlık ödüllerden etkilenir.

3.4. Pygame Simülatörü İçin Derin Takviyeli Öğrenme Yaklaşımı

PyGame, Python tarafından interaktif oyunlar tasarlamak için geliştirilen bir simülasyon kütüphanesidir. Bu makalede, temsilcimizi eğitmek için PyGame aracılığıyla 400*400 piksel boyutlarında bir ortam oluşturuldu. Şekil 9'da gösterildiği gibi 4 farklı hareketli engel ve bir temsilci bulunmaktadır. Temsilcinin engellere çarpmadan hareket etmesi hedeflenmiştir. Temsilci, ortamda

bulunduğu süre hareketli engellere çarptığı her eylem için -150 puan ceza ve her doğru eylem için +2 ödül puanı almaktadır.



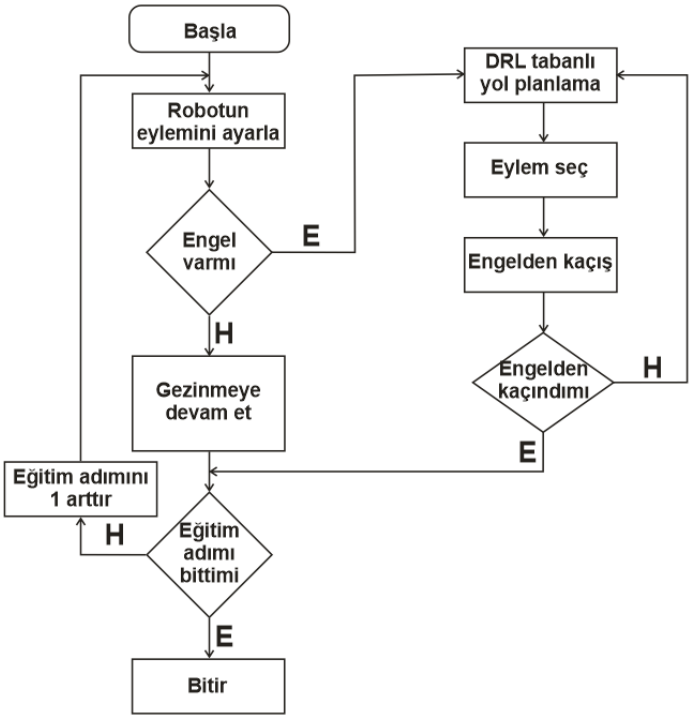
Şekil 9. Eğitim ortamı ile temsilcinin etkileşimi.

Şekil 9'da gösterilen şekilde eğitim ortamında bulunan küçük siyah kareler hareketli engelleri temsil etmektedir. Çizelge 3'te ortamda kullanılan 4 farklı eylem gösterilmektedir.

Çizelge 3. Ortamda gerçekleştirilen eylem tipleri

No	Eylem
0	Sola hareket
1	Sağa hareket
2	Yukarı hareket
3	Aşağı hareket

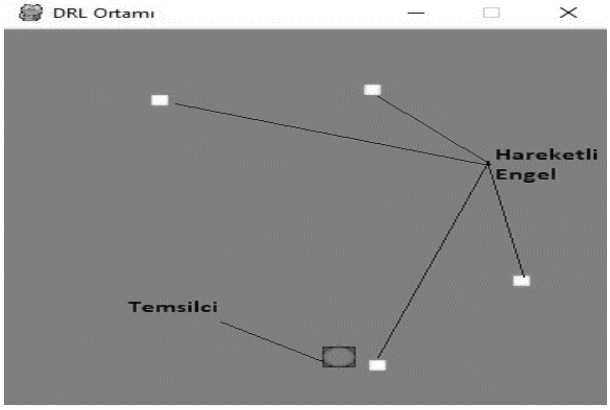
Şekil 10'da bu çalışmada kullanılan deneylerin akış şeması gösterilmektedir.



Şekil 10. Yapılan deneylerin akış şeması

4. Simülasyon Sonuçları

Bu çalışmada Python yazılım dili Keras Kütüphanesi ile birlikte kullanılarak yapay sinir ağları oluşturulmuştur. Derin öğrenme algoritmalarından LSTM, RNN, ve CNN algoritmaları benzer parametrelerle ayarlanarak ayrı ayrı eğitim gerçekleştirmiştir. Şekil 11'de PyGame tarafından oluşturulan eğitim ortamı gösterilmektedir.



Şekil 11. Eğitim ortamı

Kullanılan Derin Q- Öğrenme algoritmalarında 48 nöronlu bir katman içermektedir. Kayıp fonksiyonu olarak Mean Squared Error, aktivasyon fonksiyonu Linear ve çıkış katmanında Sigmoid aktivasyon fonksiyonu kullanılmıştır [44]. Eğitimi sonlandırmak için temsilcinin toplamda 5000 puan alması veya engellerden birine çarpması gerekmektedir. Bu kriterler göz önüne alındığında gerçekleşen eğitim, 500 adım için CNN 7 saat, RNN 45 saat ve LSTM 25 saat sürmüştür. Eğitim, i5 işlemcili 8gb ram olan standart kapasitede bir bilgisayarda gerçekleştirilmiştir. Derin Q-Öğrenme algoritmasında kullanılan parametreler Çizelge 4'te ki gibidir.

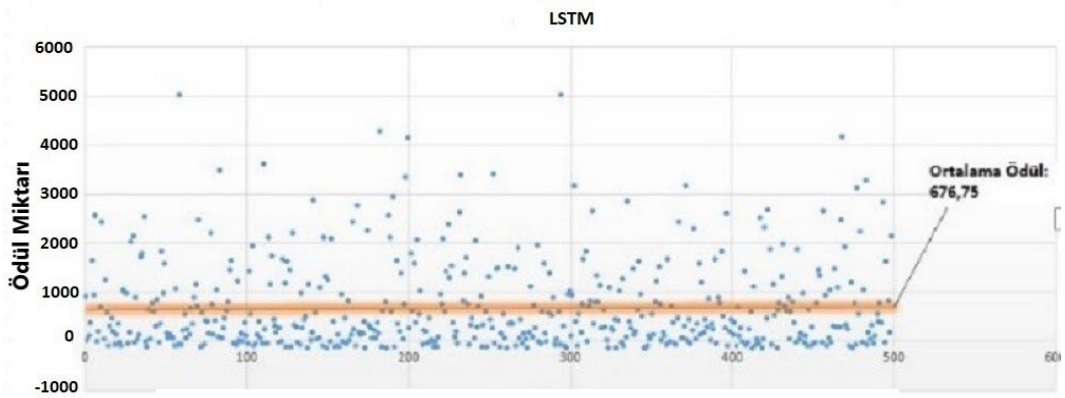
Çizelge 4. Kullanılan parametre değerleri

Parametre	**Değer**	
Tur Sayısı	500	
Öğrenme Oranı	0.01	
Gamma	0.95	
Keşif Oranı	En büyük	1
	En az	0.1
	Düşüş oranı	0.95

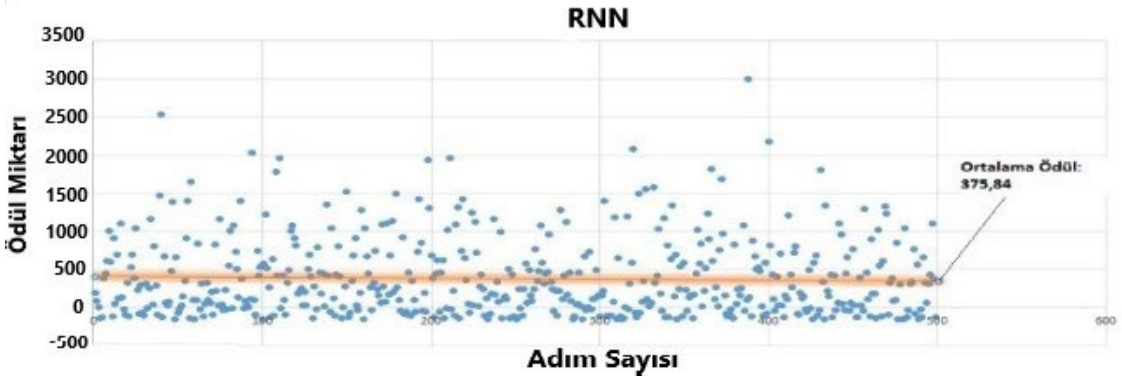
LSTM ve RNN her ne kadar dil işleme veya kelime tamamlama problemlerinde çok başarılı olsalar da, özellikle LSTM in bu tür problemlerde de başarı olduğu sonuçlar incelendiğinde görülmüştür [45]. Şekil 12'de gösterilen adım-ödül grafikleri incelendiğinde LSTM mimarisinin başarısı daha net görülmektedir. 500 adımlık eğitim süresince tavan ödül olan 5000 i iki defa almıştır. Ortalama ödül olarak da diğer mimari ağlara göre daha fazla puan almıştır. Şekil 13 ve 14 incelendiğinde ise CNN ve RNN algoritmalarının LSTM algoritmasına göre daha kötü sonuçlar verdiği gözlemlenmiştir. Çizelge 5 incelendiğinde ödül-adım grafiklerine paralel olarak ödül ortalamalarında da başarılı mimarı LSTM olmuştur.

Çizelge 5. Alınan ödüllerin ortalaması

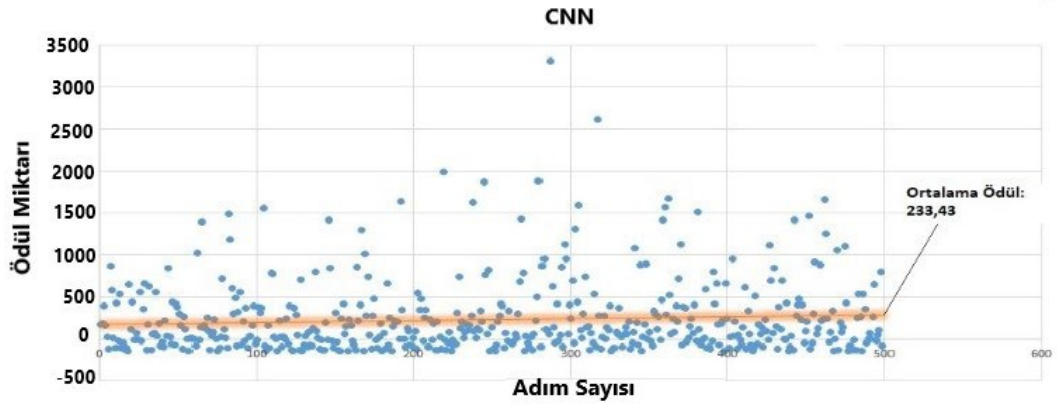
Algoritma	**Ortalama Ödül**
CNN	233.44
LSTM	**676.75**
RNN	375.84



Şekil 14. LSTM algoritmasının eğitim sonuç grafiği



Şekil 14. RNN algoritmasının eğitim sonuç grafiği



Şekil 14. CNN algoritmasının eğitim sonuç grafiği

5. Sonuç

Derin Takviyeli Öğrenme, son yıllarda otonom araçlar alanında yapılan çalışmalarda başarılı sonuçlar ortaya koymuştur. Ancak problemin karmaşıklığı ve ortamın büyüklüğü arttıkça eğitim süreci çok zaman alacaktır ve donanımsal olarak yüksek performanslı bilgisayarlara ihtiyaç duyulacaktır. Bu makalede, dinamik ortamlarda karşılaşılabilecek engellere karşı otonom robotların hareket problemlerine yönelik Derin Takviyeli Öğrenme yaklaşımı kullanarak bir öneri ortaya konmuştur. Genelde bu tür problemlerde CNN derin öğrenme algoritması kullanılırken bu çalışmada tekrarlayan sinir ağlarından RNN ve LSTM de test edilmiştir. Sonuç olarak üç farklı derin öğrenme algoritması temsilciyi bir simülatörde 500 adım eğitilerek sonuçlar gözlemlenmiştir. Simulator aracılığıyla yapılan

otonom yol planlamasında temsilci tarafından yönlendirilen robotun hareketleri raporlanmıştır. Eğitim sonucunda alınan ortalama ödüller incelendiğinde LSTM in başarısı görünmektedir. Bunun başlıca sebeplerinden biri tekrarlayan sinir ağ yapısı ve hafıza olmasıdır. Ayrıca ödül-adım grafikleri incelendiğinde CNN algoritmasında eğitim için 500 adımın yeterli olmadığı görülmüştür. İleriki çalışmalarda adım sayısını daha fazla arttırarak eğitim gerçekleştirilirse farklı sonuçlar alınabilir

Kaynaklar

[1] Z. Tong, H. Chen , X. Deng, K. Li ve K. Li, A. Scheduling scheme in the cloud computing environment using deep Q –learning. Information Sciences 2020: 1171-1191.

[2] L. A. Baxter. Markov decision processes: Discrete stochastic dynamic programming. Technometrics 1995; 37(3): 353-353.

[3] C. J. Watkins ve P. Dayan. Q-Learning. Machine Learning 1992;3(8): 279-292.

[4] C. Berner, G. Brockman, B. Chan, V. Cheung, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, R. Józefowicz, S. Gray, C. Olsson, J. Pachocki, M. Petrov, H. P. d. O. Pinto, J. Raiman, T. Salimans, J. Schlatter, J. Schneider, S. Sidor, . I. Sutskever, J. Tang, F. Wolski ve S. Zhang. Dota 2 with large scale deep reinforcement learning. arXiv:1912.06680v1 2019.

[5] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre ve T. Cai. Grandmaster level in StarCraft II using multi-agent reinforcement learning. Nature 2019;575: 350-354.

[6] M. Jaderberg, W. M. Czarnecki, I. Dunning, L. Marris, G. Lever, A. G. Castañeda, C. Beattie, N. C. Rabinowitz, A. S. Morcos, A. Ruderman ve N. Sonnerat. Human-level performance in 3D multiplayer games with population-based reinforcement learning. Science 2019;364:859-865.

[7] A. Graves, G. Wayne, . M. Reynolds, T. Harley, . I. Danihelka, S. G. Colmenarejo, E. Grefenstette, . T. Ramalho ve J. Agapiou. Hybrid computing using a neural network with dynamic external memory. Nature 2016; 538: 471-476.

[8] G. Wayne, C.-C. Hung, D. Amos, M. Mirza, A. Ahuja, A. Grabska-Barwinska, J. Rae, P. Mirowski, J. Z. Leibo, M. Gemici, M. Reynolds, T. Harley, J. Abramson, S. Mohamed, D. Rezende, D. Saxton ve A. Cain. Unsupervised predictive memory in a goal-directed agent. arXiv:1803.10760, 2018.

[9] S. W. Kaled ve Y. Sırma. Image visual sensor used in health-care navigation in indoor scenes using deep reinforcement learning (drl) and control sensor robot for patients data health ınformation. Journal of Medical Imaging and Health Informatics 2021;11(1).

[10] I. Akkaya, A. Marcin, C. Maciek, L. Mateusz, M. Bob, P. Arthur, P. Alex, M. Plappert ve P. Glenn. Solvıng rubık's cube with a robot hand. arXiv:1910.07113 2019.

[11] S. Latif, H. Cuayáhuitl, F. Pervez, F. Shamshad, H. S. Ali ve E. Cambria. A survey on deep reinforcement learning for audio-based applications. arXiv:2101.00240 2021.

[12] T. Rajapakshe, R. Rana ve S. Khalifa. A novel policy for pre-trained deep reinforcement learning for speech emotion recognition. arXiv:2101.00738 2021.

[13] M. Luong ve C. Pham. Incremental learning for autonomous navigation of mobile robots based on deep reinforcement learning. Journal of Intelligent & Robotic Systems 2020;101(1): 1-11.

[14] C. Yan, X. Xiang ve C. Wang. Towards real-time path planning through deep reinforcement learning for a uav in dynamic environments. Journal of Intelligent & Robotic Systems 2020; 98: 297-309.

[15] S. Wen, Y. Zhao, X. Yuan, Z. Wang, D. Zhang ve L. Manfredi. Path planning for active SLAM based on deep reinforcement learning under unknown environments. Intelligent Service Robotics 2020; 1-10.

[16] S. Guo, X. Zhang, Y. Zheng ve Y. Du. An autonomous path planning model for unmanned ships based on deep reinforcement learning. Sensors 2020; 20(2): 426-440.

[17] L. He, N. Aouf ve B. Song. Explainable deep reinforcement learning for uav autonomous path planning. Aerospace science and technology 2021;118.

[18] P. Li, M. A. Aty ve J. Yuan. Real-time crash risk prediction on arterials based on LSTM-CNN. Accident Analysis & Prevention, 2020.

[19] Z. Tan ve M. Karaköse. On-Policy deep reinforcement learning approach to multi agent problems. In Interdisciplinary Research in Technology and Management, Kolkata 2021.

[20] B. Bulut, V. Kalın, B. B. Güneş ve R. Khazhin. Deep learning approach for detection of retinal abnormalities based on color fundus ımages. 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), İstanbul,Türkiye 2020.

[21] S.Bozkurt. Derin öğrenme algoritmaları kullanılarak çay alanlarının otomatik segmentasyonu, Yüksek Lisans Tezi. İstanbul 2018.

[22] M. M. Ejaz, T. B. Tang ve C.-K. Lu. Autonomous visual navigation using deep reinforcement learning: An Overview. IEEE Student Conference on Research and Development. Bandar Seri Iskandar, Malezya 2019.

[23] D. Silver, A. Huang, C. Maddison, A. Guez, L. Sifre ve V. Den. Mastering the game of go with deep neural networks and tree search. Nature 2016; 529: 484-495.

[24] S. Carta, A. Ferreira, A. S. Podda, D. R. Recupero ve A. Sanna. Multi-DQN: An ensemble of deep q-learning agents for stock market forecasting. Expert Systems with Applications 2021;164.

[25] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg ve D. Hassabis. Human-level control through deep reinforcement learning. Nature 2015: 529-533.

[26] Y. LeCun, Y. Bengio ve G. Hinton. Deep Learning. Review 2015; 521:436-450.

[27] S. Dupond. A thorough review on the current advance of neural network structures. Annual Reviews in Control 2019;14: 200-230.

[28] A. Tealab. Time series forecasting using artificial neural networks methodologies: A systematic review. Future Computing and Informatics Journal 2018; 3(2): 334-340.

[29] F. Rundo. Deep LSTM with reinforcement learning layer for financial trend prediction in fx high frequency trading systems. Applied Sciences 2019; 20(9): 44-60.

[30] M. Hibat-Allah, M. Ganahl, L. E. Hayward, R. G. Melko ve J. Carrasquilla. Recurrent neural network wave functions. Physıcal Revıew Research 2020;2(2).

[31] X. Li, L. Li, J. Gao, X. He, J. Chen, L. Deng ve J. He. Recurrent reinforcement learning: A hybrid approach. arXiv:1509.0344, 2015.

[32] S. Hochreiter ve J. Schmidhuber. Long short-term memory. Neural Computation 1997; 9(8): 1735–1780.

[33] Z. Qun, L. Xu ve G. Zhang. LSTM neural network with emotional analysis for prediction of stock price. Engineering Letters 2017; 25(2).

[34] Y. Bengio, P. Simard ve P. Frasconi. Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks 1994;5(2):157-166.

[35] A. Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (LSTM) network. Physica D: Nonlinear Phenomena 2020; 404.

[36] F. Shahid, A. Zameer ve M. Muneeb. Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM. ScienceDirect 2020; 140.

[37] H. Fan, M. Jiang, L. Xu, H. Zhu, J. Cheng ve J. Jiang. Comparison of long short term memory networks and the hydrological model in runoff simulation. Water 2020; 12(1): 175-180.

[38] Z. Tan ve M. Karaköse. Proximal policy based deep reinforcement learning approach for swarm robots. In 2021 Zooming Innovation in Consumer Technologies Conference (ZINC). Novi Sad, 2021.

[39] S. Ha, J. Kim ve K. Yamane. Automated deep reinforcement learning environment for hardware of a modular legged robot. 15th International Conference on Ubiquitous Robots 2018:348-354.

[40] A. Ramaswamy. Theory of deep q-learning: a dynamical systems perspective. arXiv:2008.10870v1, 2020.

[41] R. S. Sutton ve A. G. Barto. Reinforcement Learning:An Introduction. London: MIT Press, 2015.

[42] T. T. Nguyen, N. D. Nguyen ve S. Nahavandi. Deep Reinforcement Learning for Multiagent Systems: A Review of Challenges,Solutions, and Applications. IEEE Transactıons on Cybernetıcs 2020; 50(9).

[43] S. Bhagat, H. Banerjee, Z. T. H. Tse ve H. Ren. Deep reinforcement learning for soft, flexible robots:brief review with impending challenges. Robotics, 2019.

[44] J. Qi, J. Du, S. M. Siniscalchi, X. Ma ve C.-H. Lee. On mean absolute error for deep neural network based vector-to-vector regression. IEEE Signal Processing Letters 2020;27: 1485 – 1489.

[45] Z. Tan ve M. Karaköse. Comparative evaluation for effectiveness analysis of policy based deep reinforcement learning approaches. International Journal of Computer and Information Technology 2021;10(3): 1-15.